1162 pages of:

BLENDER DUMBASS 2.0

ARTICLES COMPILATION

VOLUME 1

lbry://@blenderdumbass
or simply
J.Y.Amihud

This book in a mere collection of the meaty parts from my articles on LBRY. You can read them as articles one by one, or instead you can read this book. The articles will be slightly edited to fit nicely into a book format. For example I will remove the sponsored segments and the Moria's Race segments. But I will keep the actual articles them selves. The actual text will be kept. Promotional bullshit around the articles will be removed, since I don't see it as needed here.

Enjoy. @blenderdumbass
(Or J.Y.Amihud)

Should we cancel those who want to cancel Richard Stallman?

Freedom of speech is something people usually agree on. But what is Freedom Of Speech really. This article kickstarted my journey into answering this and other questions related to Freedom. It was not the first ever article I wrote. But it is a good starting point for this book.

`lbry://@blenderdumbass:f/should-we-cancel-those-who-want-to-cancel-richard:0`

Background

For nearly a week there is going a war about Richard Stallman. And whether he can stay on the board of directors of the FSF. You probably know the story already. But for the ones who are not familiar, a little recap.

Richard Stallman made some questionable statements about things. And in 2019 had resigned from the FSF board of directors for saying thing he said. A lot of people found it unfair. Since it's against the idea of Freedom of Speech.

Few days ago there was a little announcement about Richard Stallman coming back to the board of directors of the FSF. And the Free Software (Open Source) world gone viral about it.

Some people think he should not be allowed any position at the FSF and those who support him should resign too. Other (like me) think it's unfair to demand person to resign no matter how horrible were the things per said.

I'm biased

I'm a biased source. Because I personally never disagreed with the statements that he argued about. He was arguing about pedophilia and things related to it. And that's with no coercion and assault it should be allowed. Also he had a more famous statement about the Epstein case and Marvin Minsky. Claiming that the words "sexual assault" are too strong.

I'm a biased source since my Girl Friend Rita is currently Underage. And we are together for a year and a half. We never did anything illegal. But we had faced a large amount on pressure both on me and on her. I was under investigation by the police and spent 3 days in a real prison for willing to keep this relationship. But we stayed firm and we are still together and waiting for her maturity in order for the pressure to end.

Counter-action

A lot of people are doing the opposite and trying to support the FSF and Richard Stallman. For example there is an open letter in support of Richard

Stallman. You can find my name is on there already. And you can add your own too.

But some more extreme views to support Richard Stallman are also present. Mainly the idea of @DistoTube to cancel those who want to cancel Stallman. (He is talking about the Gnome project people).

The problem with this approach

The main problem with cancelling those who want to cancel us is hypocrisy. The idea of Freedom which is pioneered by FSF and Richards Stallman includes Freedom of Speech. And those who want to cancel Stallman do that in speech.

While they don't start any physical attacks on anybody, their opinions should also be respected. And their projects not cancelled.

Richard Stallman has a dislike toward the Coca Cola company that I share with him. For that in Mexico they led people to death with their business model there. This is different from speech.

There is a boycott on the SONY company because they put a man in prison for trying to change software on his PlayStation device. This is different from speech.

If all the entity does is states an argument. It would be wrong to cancel them because of it. No matter if it's Richard Stallman, Gnome, or any other entity.

Alternatives

This isn't good to see people trying to cancel people. But instead of cancelling them. How about using speech to protect the Freedom of Speech. How about carefully reading their arguments and stating counter-arguments. How about publishing works that state an opposite opinion.

If you have an opinion. You can write an article and either post link to it or write the whole thing in the comments of this post. Let's keep speech working.

Happy Hacking!

EFF against Richard Stallman. My answer to that.

When there is bad speech. Speech that you don't like. Speech that you don't agree with, the best way to counteract this speech is by arguing with it, adding more speech. Not by shutting them up. It's not going to help nobody.

`lbry://@blenderdumbass:f/answer-to-eff:c`

In my post *[previous article]* I urge people to fight for Freedom Of Speech using speech. And I would start by criticizing / arguing with the statement of the Electronic Frontier Foundation (EFF) on their statement about the current Richard Stallman situation.

Their post:

https://www.eff.org/deeplinks/2021/03/statement-re-election-richard-stallman-fsf-board

Credit to the EFF

While I'm against their post on the situation. The EFF is not a bad organization as a whole. And one thing I love about their web site in general that's all the JavaScript code on their site in under GNU GPL license. It's one of not many sites which LibreJS doesn't have any issue with.

Let's break their post apart.

> ...after a series of serious accusations of misconduct...

The sentence links to another article by Selam G. (Sep 16, 2019). I guess we can't ignore the post it links to in order to understand what "accusations" they are talking about.

There is a list of 2 things they accuse Richard Stallman of. Let's look at them one by one. Because over-generalization is a bad thing. (In the article you can see 3. But the third one is not a direct accusation against Richard Stallman. But a proposal of how to deal with uncomfortable things.)

1. Richard Stallman has problematic opinions.

The article goes on explaining his views on child pornography and underage sex. And states some numerical values for making it sound as "wrong".

> If there are a large number of people in the United States who think that child pornography and sexual intercourse with minors should be legalized, this is the first I’m hearing of it, and please show me the evidence.

Basically as I understand it. They accuse Richard Stallman of wrong thought because his opinion is not popular enough. First. Any opinion should be able to be had in a Free society that respects Freedom of

Speech. It doesn't matter how much you personally dislike it. Or how much popular it is. Any opinion matters. And any opinion should be possible. So the whole point 1 doesn't make any sense to begin with. No matter what would he say.

Second. Even if the opinion is not popular. Or even if the opinion makes you, or most people uncomfortable. It doesn't mean it's wrong. His views on that subject could be wrong. But also could be right. The only way to check, would be to make a test. Which isn't very possible, to put it lightly.

But even if he is mistaken or wrong. I'm a black woman that was born in France and lives in India. And my name is Sara MacBrine. It's all lies. And I'm perfectly aware that they are lies. And I still can write them because Freedom of Speech exists. I can be as wrong as I want to be. Or as vulgar as I want to be. Or as unpopular as I want to be. It should never be a crime.

1. Richard Stallman has been contributing to a negative environment for women at MIT for over thirty years.

The article goes on recalling events of cringe situations Richard Stallman had with some women in the MIT. And that he had mattress in the office.

I understand that some people are uncomfortable. And especially technical people who don't have a lot of experience with talking to humans could be very uncomfortable. I know plenty of very smart technical people who are very hard to sit with.

But nothing of what the article described looks like illegal activity. Maybe some of his behaviour is "childish" (I don't like simple mindedness to attribute to children. It's Ageism in my opinion). Like it looks like a young boy's thing to say "I gonna kill myself if you don't like me." But it looks rather harmless to me. You can disagree with me if you want.

So let's go back to the EFF article. After the link to the accusations we just looked at, the article follows with.

> We are also disappointed that this was done despite no discernible steps taken by him to be accountable for, much less make amends

> for, his past actions or those who have been harmed by them.

If a person's believes are that people should be allowed to do X. Apologizing for doing X would be a hypocrisy. Richard Stallman is a big supporter of any type of sex Freedom. He ended up being uncomfortable because of these views to the people who do not share these views with him. And it's understandable. Him being accountable for views. Or being accountable for being uncomfortable because of his views, feels like a thought police, censoring regime to me.

> Stallman’s re-election sends a wrong and hurtful message to free software movement, as well as those who have left that movement because of Stallman’s previous behavior.

I respectfully disagree and my counter-argument is: Free Software's 0's Freedom states that everybody should be Free to use the software for what ever purpose at what ever time. Free Software is about Freedom. Freedom of speech is one of the Freedoms. Tor project is known to a lot of people as this "Evil" program that let's people sell drugs online. But there is nothing wrong with Tor as there is nothing wrong with Richard Stallman. Some aspects of both could be uncomfortable.

People poop ones every some time. This is unconformable to think about. Your mother had sex with your father. This is uncomfortable to think about. Kids masturbate. This is uncomfortable to think about. People eat other people. This is uncomfortable to think about. *Let's now cancel life.*

> ...individuals cannot place misguided feelings of loyalty above their commitment to that cause.

This is exactly why people who disagree strongly with Richard Stallman also support him. The cause is Freedom of Speech now. And silencing a person because of his unpopular opinions is unjust.

And then they conclude with what should be done, saying that Richard Stallman should resign again. To which I would just add. Please don't. Richard Stallman is a guy that we need. And there is nothing wrong with him being a leader of FSF.

My conclusion

Richard Stallman is the man that is so good at Free Software ideology that anybody unable to say things that he said would just not be right for the job. Imagine promoting something but doing it only in the way that make everybody comfortable. This is impossible. If per is pro Freedom. It's better per be pro Freedom in all ways. Even those where it's uncomfortable to people that might agree with the other half of the statement.

Freedom of children for example cannot happen if some things are not allowed for them that are allowed for non-children. It's hypocritical to be for children to have a right to vote for example. But be against children to have a right to have sex. If what you argue is for total and complete freedom of children. And not just one aspect of it.

Richard Stallman is for total and complete Freedom of everybody. And you can't have it without sexual Freedom too. And it is uncomfortable aspect of otherwise very comfortable idea. Only few people are as firm in their believes as Richard Stallman is.

So instead of cancelling him. We shall solute him for it.

Happy Hacking!

Odysee Livestreams are Bloated and Problematic

Sometimes writing about problems may be the first step into solving them. This is why I would encourage people to submit bug reports and other issues to your favourite Free Software developers.

`lbry://@blenderdumbass:f/My-Thoughts-on-Odysee_livestreams:c`

Background

A few days ago following the Richard Stallman situation I discovered a channel, @Tuxfoo that had live-streams in Odysee. Many channels on Odysee are actually YouTube channels but with a Sync feature turned on. Which means that when they have "streams" or "stream highlights" it's usually a stream from YouTube or Twitch. Because the LBRY protocol on which Odysee was built wasn't supporting streaming live.

This is changing. There been talks about the streaming in the near future. And on some posts of @LBRY you can see tests for the streams. The @Tuxfoo channel had real Odysee "native" streams. Which means that there was some way to enable streaming already.

And a few hours of digging a bit I actually found a way to do streams, about which I made this video.

Streaming is not for everyone

It's turned out to be that streaming yet is not meant for everybody. According to this post quote:

> Many features, like live streaming and transcoding right now, go through selective release cycles. In many cases, rather than play favorites or release things randomly, we will use the leveling system to roll them out. In the future, some features may only be available to higher level accounts.

Meaning you have to have a certain number of stars in order to get the livestream. Because they are still testing it they don't just want to enable who ever, randomly start streams. Their servers might not even be capable if Odysee suddenly turns into a clone of Twitch.

In my video earlier. I didn't know this information. And though it was some kind of regional limitation. Because I saw an example where people had the livestream select-able from the upload menu as another option. It was not very hard to unlock tho.

The simple upload url looks like this https://odysee.com/$/upload. If you want to simply write a post. Like the one your are reading. You can either click the Post button in the menu on the top.

Or use the url https://odysee.com/$/upload?type=post.

For livestreaming I found 2 urls that will enable you to do it at what ever star level. You use https://odysee.com/$/livestream to enter the livestream menu. There will be you past streams. And the menu for the Stream Server and Stream Key.

The url https://odysee.com/$/upload?type=livestream will hack the UI giving you the configuration of the stream. It will give you things like Name, Description, Thumbnail, Tags, and the deposit info.

This is all just hacking. Or should I say cracking? I actually don't know. The software is Free that means I can use it for what ever purpose. And I can edit the software. So there should not be a problem if I find a way to enable livestreaming when it's not yet should be enabled for me.

Bitwave.tv (the problem with Odysee streaming)

Later that day I was playing around with a stream. I made this stream. You can see it's dead. Because it's ended and Odysee doesn't record the stream. I knew it. So I recorded it myself. And made a little highlights video.

During the stream I couldn't make it show the video in the GNU IceCat browser. My default browser. It worked reasonably well in Brave. And I started digging on why.

The IceCat browser is developed for the users that want to use only Free Software. And gives the user total control over the web page. For example. All the non-free (proprietary) JavaScripts are blocked by the LibreJS. It's not perfect. Odysee is also blocked by default with LibreJS. But I can make it work if I want it to. And knowing that Odysee is Free Software under the MIT license (the good one). I can choose to whitelist all the scripts coming from Odysee.

The other thing that IceCat does. Is it blocks all third party requests. For example. A lot of websites use services like Google Analytics to collect data about the use of the site. Of course it's contributing to the surveillance of Google. So please don't use it.

The third party blocker will block the Google thing because it's scripts are coming from a different domain. On Odysee there are a lot of third party domains. Most of them lbry.tv , lbry.com, spee.ch and some others are pretty harmless. So I enable them.

But when I entered the stream window. A new domain appeared in the list of check-boxes. It was bitwave.tv. If you go to their site. It claims to be a "open source livestreaming service". On their github page. They have a GNU GPL v3 license. Which makes them Free Software.

So I enabled the domain. And I enabled the various scripts it was trying to load. (All of them had missing License, but since the whole thing is Under GPL v3 I didn't care much). And then it started requesting more domains.

This time tho. It started requesting google.com and youtube.com which is weird to say the least. Also it seems that they've been infiltrated. Their Warrant Canary was removed. Or so it seems. The service doesn't seem clean at all. Apart from having a good license for it's code.

It feel like one of those "free software applications" that are meant to be clients for a proprietary platform. But actually are tiny browsers that simply load the website. And this is not cool. Since the JavaScript of the website is usually not free. This is why we use LibreJS.

What if bitwave.tv is a weird youtube API thing to stream video. Using YouTube streaming service. But implementing a different UI. It seems like it's the case. Since it was trying to connect to youtube and google.

Of course I'm not sure. It was yesterday and I didn't even look at their code yet..

Conclusion

I think what I may do is to stay away from streaming at all for a while. I might use Jitsi or something if I really need to show people something in real time. It can handle quite a class. If you want to do such thing. Please contact me. I would love to hold a class of Blender-Heads.

For Odysee streaming. I don't know. Maybe all those issues will be fixed. Maybe the implementation is not terrible but looks like that only on the surface. Maybe after people will dig into the code of all of it. Odysee code and bitwave.tv code. Maybe it is all harmless. But until then. I will probably stay away from these.

For those who want to move from YouTube and Twitch to Odysee for streaming. Yes. It's better. It's like having proprietary software on GNU / Linux. It's better then using Windows. It's a step in the right direction. But it should not be the end point. The end point should be 100% Freedom.

Happy Hacking!

Are LBC against the law?

LBRY Inc. the company being the protocol on which these articles are published was sued by the SEC. They claim that LBC are against the law and with it they might make all crypto-currency illegal in the US.

lbry://@blenderdumbass:f/are-lbc-against-the-law:2

Disclaimer

I'm not a lawyer. And I don't live in America. I'm a 23 year old Dumbass that by chance reads English for some reason. The following document is my thoughts on the LBRY vs SEC situation that's going on.

Allegations

In this post by @LBRY they state a legal situation with the US government. The SEC (Securities and Exchange Commission) based on details of implementation and use of the LBC crypto-currency by the LBRY Inc.

Their problem is that LBC looks extremely similar to Securities (a form of investment) and since the exchanges are not properly registered, they violate the Securities Act of 1933 laws.

In English. The Government of the US things that LBRY protocol is a Market Manipulation scheme to make few individuals (people at LBRY Inc.) get rich by cheating the system. Or something like that.

They think that LBC is too similar to stock market shares or something. As in people buy them and expect to sell them for more money if the company gets bigger over time. But since they are not registered in a formal procedure, they argue that it's all illegal.

Misunderstandings

LBRY published the official document of the complaint from SEC. And I read through all the 16 pages of the legal code there. It's quite migraine inducing to be clear. But I survived through it.

For what I understand all their trying to do is to prove how LBC is a "Security" (a share-like investment thing). And they, from what I got, either intentionally or by mistake confusing the LBRY Inc. with LBRY Protocol / LBRY Network with LBRY Software. Which are related things but not the same things.

Also they completely misunderstand the ideas behind things like Free Software and Decentralization. Because in their complaint they say quote: *(pages 12 - 13)*

> As of March 2021, LBRY’s efforts to deliver on its promises and develop the LBRY Network are on-going. LBRY maintains managerial and entrepreneurial control over the LBRY Network. LBRY continues to control its software code for its applications and the protocol. LBRY continues to unilaterally make strategic and managerial decisions about the future of the LBRY Network. LBRY continues to unilaterally decide how to allocate the capital and resources it has pooled from investors to grow the Network, which it represents on its website will increase the value of LBC.

LBRY protocol, Network, Software are all Free Software and Decentralized. Decentralized in implementation, decentralized in development. LBRY does not control the protocol or the software. They control only their branch. And anybody can fork LBRY and develop it into a different direction.

I can fork LBRY applications and add or remove features to and from them. And release my own versions of them. This is fundamental. Because the Government had completely overlooked that.

They claim that the growth of LBC is directly linked to how well the company LBRY Inc. is doing it's job. Producing software, promoting and so on. But LBC will grow with the protocol. With or without the LBRY Inc. (the company) involvement.

What if Uncle Bob suddenly forks the software for LBRY, makes it's own Bobysee web site using the protocol. And makes a huge campaign to promote Bobysee. And it becomes a huge success. The LBC will grow. Even if LBRY Inc. will not do anything at all meanwhile.

What if LBRY looses in Court?

The consequences of loosing this battle are huge. Because suddenly all crypto-currency platforms can be targeted with similar allegations. Making their existence either impossible. Or a bureaucratic nightmare.

In the allegation papers they state things that they think should be done with LBRY Inc. *(pages 15 - 16)* And as I understand it. They want everybody related to LBRY Inc. to be prohibited to ever using blockchain technology. **Basically a kind of Ban on**

crypto. But only for people who worked on LBRY.

For the LBRY protocol the changes will be minimal. Maybe LBRY.TV and ODYSEE.COM will close. But using the LBRY Desktop you will have access to the protocol meanwhile. Until somebody develops a more user friendly web-site solution. *Perhaps Uncle Bob*

Counter-action (How to help LBRY?)

- Go to HelpLbrySaveCrypto.com and see what are the official ways to help this cause.
- Share the message. Write your own articles. Make videos.
- Sign petitions to Free crypto-currencies. There is this one. Tho if there is a better way of doing similar thing, it would be cool. Since Change.org has problems.
- If you are a lawyer, crypto-currency expert, smart person. Please look at this situation. And publish your opinion.
- If you are a blogger, news organization. Contact press@lbry.com

Let's help Free Software!

Happy Hacking!

Libre Binge (Using GNU / Linux isn't enough)

Sometimes you need something to watch but don’t know quite, what exactly. So I made a little compilation of good clips for people to see. Here I list them (with links).

`lbry://@blenderdumbass:f/Libre-Binge-Watching:5`

The COVID-19 pandemic made people bored

Taking inspiration from this post from DefectiveByDesign.org I want to make a playlist. A list of video / audio links that are nice to spend time with. During the rest of the pandemic we have too much Free time that we can use. And there is an urge to watch a bunch of videos. Of course there is Odysee. But some videos I want to share with you are not on the platform. And re-uploading them is a tedious job.

I will not use any non-free services for the video links. Meaning I will not use YouTube or Netflix or what ever else that doesn't respect your Freedom.

I don't have a definite plan of what list it will be at the moment of typing this sentence. But I have an idea of the subject.

And the subject will be:

Using GNU / Linux is not enough

For the last 10 or so years I was using a GNU / Linux Operating System as a daily driver. Something tho struck me very recently. The importance of Free

Software. Importance of Freedom. Is way more important than just using an operating system X.

Me and my brother were constantly in an argument for the last couple of years. It's basically a never-ending "Linux vs Windows" battle. Which if you think deeply about it, makes no sense.

First of all, the "Linux" is not an operating system that can be compared to Windows. GNU/Linux variants might be. A whole system. Not a mere kernel.

But most of the time the problem he has with GNU/Linux are ones that miss the point if you really think about them. **Software (Games in his case) that work well on Windows but don't work well on GNU/Linux.**

This lead me into a rabbit hole of discovery. And for the past few years I started understanding what actually I want from GNU/Linux. The Freedom. The Freedom that could be stripped away should I install Steam and play a proprietary software game. The Freedom that will be gone should I upload or watch videos on YouTube.

I had an Android Smartphone. I used to regard it as enough. Because Android is a Linux Distro (Not a GNU/Linux distro). But due to chain of *unfortunate* events I don't have it. And I'm happy.

So here are some videos that will lead you to understand Freedom. And stuff we need to fight for and against in order to keep the freedom from being taken from us.

DRM

One of the strongest enemies of technological Freedom is a Digital Restrictions Management and laws that comes with it. In order to understand it and understand technological Freedom I recommend a talk by **Cory Doctorow** he gave at the Libre Planet in 2017:

Beyond unfree: The software you can go to jail for talking about

https://media.libreplanet.org/u/libreplanet/m/beyond-unfree-the-software-you-can-go-to-jail-for-talking-about/

Free Art

On a good note. Let's not get ourselves too invested in the negative things. And let's focus a bit on the positive. A lot of complaints about Free Software and lack of Digital Shackles come from people who "there for the artists". Let's hear from an actual artist about all of it. David Revoy. And his decision to publish his work under the Creative Commons Attribution license. The Libre-Planet Q&A from 2020:

Free software design and Q&A with David Revoy

https://media.libreplanet.org/u/libreplanet/m/free-software-design-and-q-a-with-david-revoy/

Patents

Not everything is always good. And while some problems are easy to understand. Some other problems require a focused thought to even get the problem. But when you see it ones you can never un-see it. Software Patents are one of those problems. There is an old video I would like to recommend you on that subject that made me understand it quite a bit. It's a presentation by Richard Stallman dated May 18, 2005 at the University of Calgary, Canada:

The Danger of Software Patents

https://audio-video.gnu.org/video/rms-speech-patents-calgary-20050518-320x240.ogg

Make worlds with Free Software: The lesson on Hacking.

Blender is one of the coolest Free Software projects out there today. Just to know what it can do will boost your expectations for Free Software. Ian Hubert is one of the most entertaining people using Blender. And I would recommend you to watch his charisma in action at his talk in Blender Conference at 2019:

World Building in Blender

lbry://@Blender:2/world-building-in-blender-ian-hubert:2

Music

And for the end. Just to keep you in hope I would recommend to learn and listen to various recordings of the Free Software song. There is no one official recording of it. Each and every one could be official. There are recordings in many styles of music. And you can probably find one for your taste. The song was written by Richard Stallman. And the list of versions can be found at:

https://www.gnu.org/music/free-software-song.html

Should "Content" be Free?

People want Free Speech while simultaneously wanting to ban people they don’t like or cancel ideas they disagree with. But should anything be free to publish? And if yes, how to make it relatively safe?

`lbry://@blenderdumbass:f/Should-Content-be-Free:b`

We are using Odysee and the LBRY protocol because we want "Content Freedom". But should Content be Free? Should people be able to post what ever they want?

What is Content?

In the Wikipedia page about Content there is a quote:

> While the marketing and media world have broadly accepted the term "content", some writers complain about its inherent vulnerability towards misinterpretation. Others assert that the term devalues the work of authors or sets up a false analogy of information as material objects which biases any discussion using the word.

In the Wiki-dictionary about the word Content is says quote:

> From Middle French *content* (“satisfied”), from Latin contentus (“contained; satisfied”), past participle of*continēre* (“to contain”).

Content. Stuff inside a container. Stuff inside a box, inside a package. The thing you got the package for. But is "content" in the digital world means "stuff in a box"?

I buy a pack of rice. The content of this pack is the rice. I will take the content out and consume it. There will be less and less rice. Until there is no rice left in the pack. No more content in the container.

When you watch a video on Odysee. Does it make the video disappear? When you listen to a song. Is the song gets consumed, and there is no more song? Or you can watch the video again, listen to the song again. Potentially countless times?

I'm not the first person to think about the language and see these kinds of "mistakes". Whether they are intentional or not. People should give language a bit of care. You can use what ever words you want. But then it's you to blame if you confused me.

You can look at the list of words to avoid here. From now on I gonna refer to "content" in this post as "publication" or "work" (depending on the type of "content"). LBRY is a publication protocol. The videos or other publications are works by people who done them.

Should Publications be Free?

From one side yes. All publications should be allowed. And no work, video, audio, text, image or anything else should be censored, removed, blocked, banned, illegal and so on. Any publication whether factual or not embodies knowledge, information, opinion. And those should be allowed to be expressed and published.

From another side no. Some publications make people uncomfortable. Some information being available could lead to dangerous situations. What if the enemy knows about our military plans? What if the Facebook knows about you too much? And for even more extreme cases. What if the work's nature, the stuff in the video, the stuff in the image are so disturbing that it might cause psychological harm?

Publication Freedom vs Privacy

Facebook started out as a Freedom place. Where people would publish their photos and stories. And everybody could see and read them and react to them. A "*content* freedom". But it turned into a surveillance engine making data about people available to other people it wasn't supposed to be available to.

Should I be free to stalk you?

Publication Freedom vs Psychology

One of the views Richard Stallman was persecuted over the last week or so was his old statements about Child Pornography. And that it should be legal. Imagine that suddenly any type of video or image is legal. No matter how gross or how terrible or how unpleasant it is. You go to Odysee and among the usual memes and videos about tech you see a video titled "Orgy with dead babies" with a clear thumbnail which is a screenshot from it.

Should I be free to gross you?

Freedom vs Power

In the Facebook's case. It's either you get the Freedom to control what you decide to publish and where. And what you want to remain hidden. Or Facebook has the power to decide it for you.

In the second case. It's either you have the Freedom to choose what you want to see. Or somebody will force something on you.

Freedom is when you can control yourself. And things related to you personally. **Power** is when you can control other people.

The YouTube algorithm

On YouTube *(don't use YouTube)* there are countless hours of video being uploaded every minute. It hosts so much video that finding what you need just by looking at the list of all videos would be virtually impossible. So ways to find stuff was needed. Search and Categories, Tags and other things were implemented early to help people navigate through the insane amount of stuff, the platform is hosting.

Subscriptions were added to help people find people who's works they like. And who's future works they are interested to see.

And everything was fun and amazing. People had the freedom to search for what ever they wanted and to curate them selves a page of recommendations they personally chosen. *Freedom*

But then they chose to introduce an "algorithm" (which is totally SaaSS by the way) that auto-curates the recommendation page for you using some kind of **hidden** "magic" algorithm that's maybe AI *but who knows really* (It's not Free Software). Anyway they control what you will discover. *Power*

The 2 questions from above

Should I be free to stalk you?

As long as you intentionally make a publication for random strangers to see, I should be able to see this publication. Examine it. Share it. Criticize it. If I choose to do so. But if some work. Or image or a video is not intended to be published. Unfinished work. Personal information. Anything you choose to hide from random strangers. I should be able to ask you for it. But you still should have the Freedom to refuse to give me it.

Should I be free to gross you?

No. At least not in the way you didn't allow me. I mean if you need gross publications you can still find them anyway. People publish them. Sometimes illegally. LBRY has an option to flag a publication as mature. And only people who will enable viewing mature will see the publication.

In my opinion it's both wrong and not enough. It's wrong for an unrelated problem I'm trying to fight against called ageism. But it's a minor issue of naming. Let's propose a different name. Let's say gross. So there will be either normal publication and gross one. This doesn't seem to be enough either.

In my opinion it should be a way deeper system. For example. Let's give people ability to flag publications by the type of the grossness. For example. porn, death, swearing etc... The flags should work similarly to tags. That means people could make additional flags that are not yet present in the system. Users could choose what flags to allow and which to block. So a person that likes porn but doesn't like eating-poo could enjoy one but avoid the other. *Freedom*

The House That Jack Built

Child Pornography is illegal in most countries. And it's completely understandable. It make people sick inside to even think about it. But I think there is an argument to make that it's still not the worst type of video that could be produced.

One very interesting individual is Lars Von Trier. A film director from Denmark with a hard past. He is suffering from chronic melancholy and make very uncomfortable movies. I think he is trying to criticize the censorship laws of many countries. Because his films are usually going on the edge of what's allowed to film. Antichrist. The 2 Nymphomaniac films. And the worst of them The House That Jack Built.

I have little brothers. And one of things that's unavoidable when you live with little kids is you see them naked ones in a while. Sharing this kind of images would be against the law. Tho most of the time it's harmless. Unless they start a fight in the tub. And I need to break them apart.

The House That Jack Build didn't try to depict something that's not allowed by law. It had very gruesome scenes of Child Murder. Those kinds of images are completely legal. Still watching the movie was so much worse than any naked butt of a kid.

After seeing the film I had 6 months depression. I had nightmares about those scenes. I needed to contact the actors who played the children just to calm down. Just to stop worrying. In a different movie I saw a naked kid's butt. I laughed a bit, that's it.

Do I want The House That Jack Built to be banned? No. Some people didn't even notice the murder scenes. It was my personal problem. What I would like to have had instead is to know how severe will be what I was going to see.

Today before watching a new movie. I usually go to some movie database site. And see the list of severe scenes. And if there is something to do with Child Murder I avoid these movies.

Conclusion

Gross stuff. Stuff like Child Pornography. Or even **worse** Child Death Images exist. They are being published regardless of whether it's allowed to do so or not. They are watched. Shared. Used in different ways by different people. This thought is probably makes you very uncomfortable. But life is uncomfortable. If a person X wants to see publication Y. Why should I care?

The Freedom of publication should be there. But not without a consideration for the person that's going to view your publication. Some way of warning them about things they might find uncomfortable should exist. Whether it's a list of things on some database site, flag / tag system or just a warning in the beginning of the publication could be nice.

About laws. I think I kind of agree with Richard Stallman here. If the warning system will be sufficient enough. Why not allow any type of publication? Why not have publication Freedom?

Happy Hacking!

Making Cars With Blender (Free Software)

Not every article should be philosophical rambling. Sometimes I just want to share with you some of my passions. This time a passion of making cars using Blender.

`lbry://@blenderdumbass:f/makingcarsinblender:6`

When I still had my YouTube channel *(you can find an archived version of it on lbry://@blenderdumbass:c)* One of the last things that I was doing there was the assets for the movie Moria's Race. And by the name of the film you can guess it was a lot of cars. I had to design, model and rig a lot of cars.

When I did them. A part of it was recorded and streamed on YouTube back then. *Some of it survived the LBRY synchronization. But some didn't.*

I want to try both compressing the information I leaned while making those car designs. And also I want to try to publish it in a form of a text article with some images here and there.

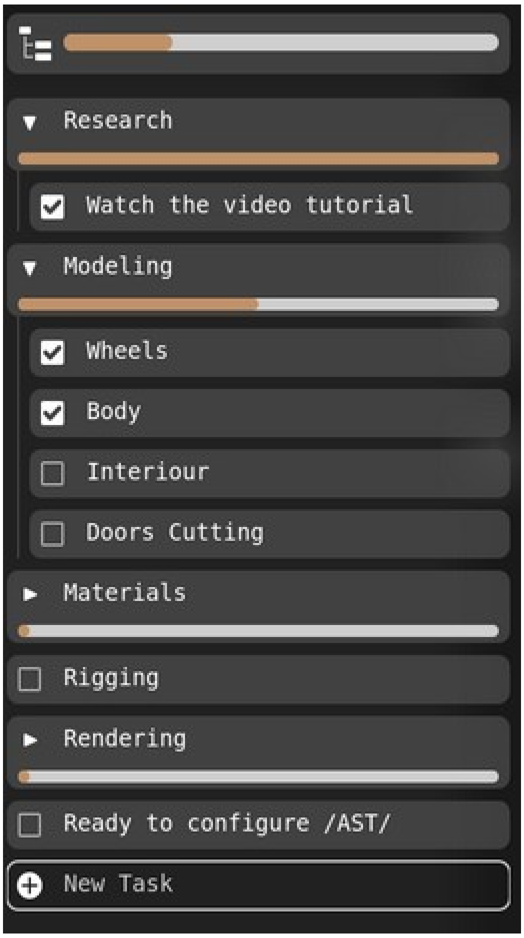
The images in the article are uploaded to LBRY protocol too. Using a separate channel. If you are interested [this is the channel](#).

Plan

A lot of models and designs require an initial plan. Concept art. Plan of action. A checklist of tasks. And other things. For the movie Moria's Race I've made myself such checklist. And for a purposes of this articles. I will expend on the checklist to make it more general.

[ ].Research
  [ ] Photos
  [ ] Concept Drawings
  [ ] Blueprints
[ ].Modeling
  [ ] Wheels
  [ ] Body
  [ ] Interior
  [ ] Doors Cutting
[ ].Materials
  [ ] Car material
  [ ] Wheels Materials
  [ ] Seats material
[ ] Rigging
[ ].Rendering
  [ ] Make Preview
  [ ] Edit Preview
[ ] Ready to configure /AST/

This checklist could be printed out or filled in a file. You could even make a spread sheet with this. To make it feel up the % of a given vehicle while checking in tasks one by one.



▼ Research

☑ Watch the video tutorial

▼ Modeling

☑ Wheels

☑ Body

□ Interiour

□ Doors Cutting

► Materials

□ Rigging

► Rendering

□ Ready to configure /AST/

⊕ New Task

Or on other hand you can do what I did. Use a special purpose software to display checklists. Making movie assets is hard and I made a peace of software (Free Software *GNU GPL v3*) to help me make those assets. It's called Victorious Children Studio Organizer. Or simply VCStudio.

You can visit and get a copy of VCStudio from out git repository:

https://notabug.org/jyamihud/VCStudio

Image above is how the same checklist will look inside the VCStudio. It's also a checklist editor in the same time as a functional checklist. And you can nest tasks with sub-tasks. And build large hierarchies of tasks. So have fun with it.

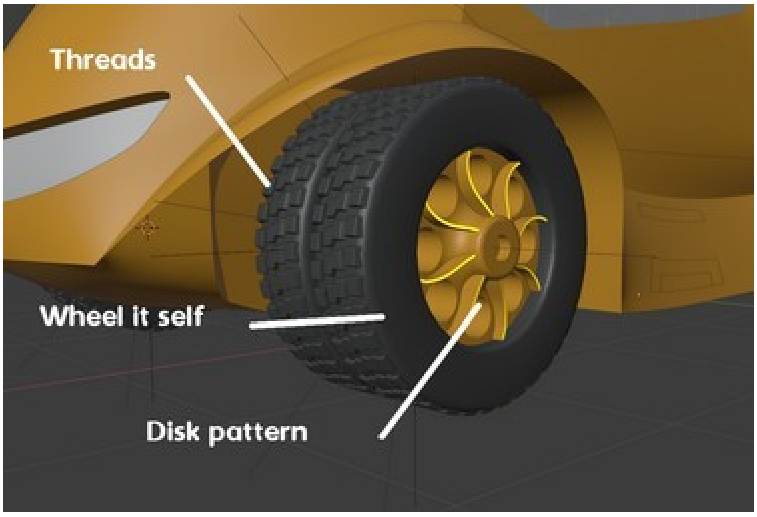
Modelling

In the middle of the checklist you can see the main 4 tasks for Modelling. The Wheels, Body, Interior and the Door Cutting. You can work in what ever order you would like. But I found this order works best when needed a reasonable good design of a car, in a reasonable time.

Wheels

I will start with wheels for the same reason you might want to add spheres for the eyes when sculpting a head. So I could feel my shape a bit more clearly.

For each wheel I use 2 to 3 objects. Let's break down one complex wheel.



Wheel it self. *The cylinder.* Is the easy thing to put. But also the easy thing to get wrong. It's your wheel sketch. It's what you will base the proportion of the other parts to. So it's important to make the proportions, the sizes as close to perfect as possible.

You are now an engineer and trying to design a car that works. And it will not work if the wheels do not make sense. (At least in the context of the story world you are trying to populate).

Threads are optional. Some very quick race cars do not have them. Because they need maximum traction. And it requires all points of the wheel to touch the ground. Threads are used for many various things. Like pushing away the water. Or reaching and grabbing through soft material like snow or ground. The reason for the threads will influence the threads design.

On the race cars I used a barely noticeable bump map for the threads. Not made in modeling. But in the shader. For the truck it would not work. They are too aggressive to just use a normal map.

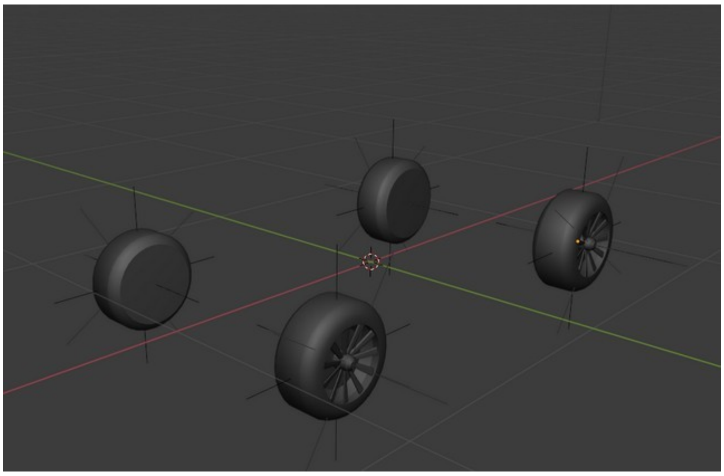
I see a lot of people making a mistake and combining the wheel it self with the threads into one object. *Perhaps because it's one object in the real world*. But it's making the modeling of the wheel so much more complicated. The main part could stay this shaved off peace. While the stuff of top is separate. It will make it easier to make.

For the threads I use the Array Modifier with an offset object. You have to decide how many duplications there should be. And in the offset rotation, type in the expression of 360/<amount> the <amount> is the amount of duplications. Blender can do easy math calculations in any input entry where you input a number. So typing 360/3 will give you 120 after you press enter.

Disk Pattern is technically similar to the Threads. In that it's using the same technique to make. But this time you can experiment with the shape. Get your creative juices flowing and make something unique for this wheel.

Note that in the picture above The Pattern is done using 3 parts too. The main part from the middle. The 8 blades thing with yellow corners. And the 8 holes in the main part made using a hidden object that applies a Boolean Modifier to the main part. That makes the cutting of the holes.

Body

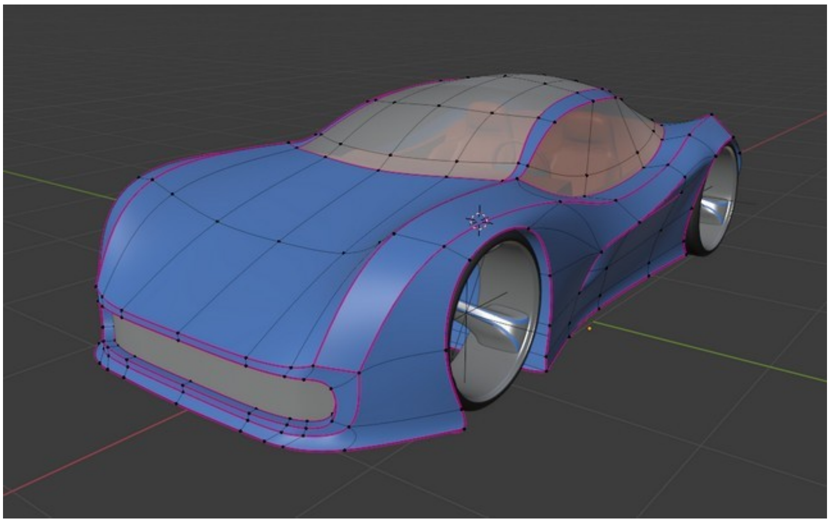


The main body of the car is very important. Before you can do that I would duplicate the wheels into all their positions ahead of time. Effectively baking down the length and width of the wheel base.

Note: These aren't mere copies of the wheel. Because realistically you would need to make a copy of the whole wheel setup with modifiers and offset objects. And it would be a maintenance nightmare.

So instead I pack the wheel into a separate collection. Un-check this collection to make it disappear from the scene. And add it as an object (that is an instance of that collection). A link. Meaning that I can always go back into the wheel.

But when I'm not editing it. It's as if it was baked and linked 4 times.



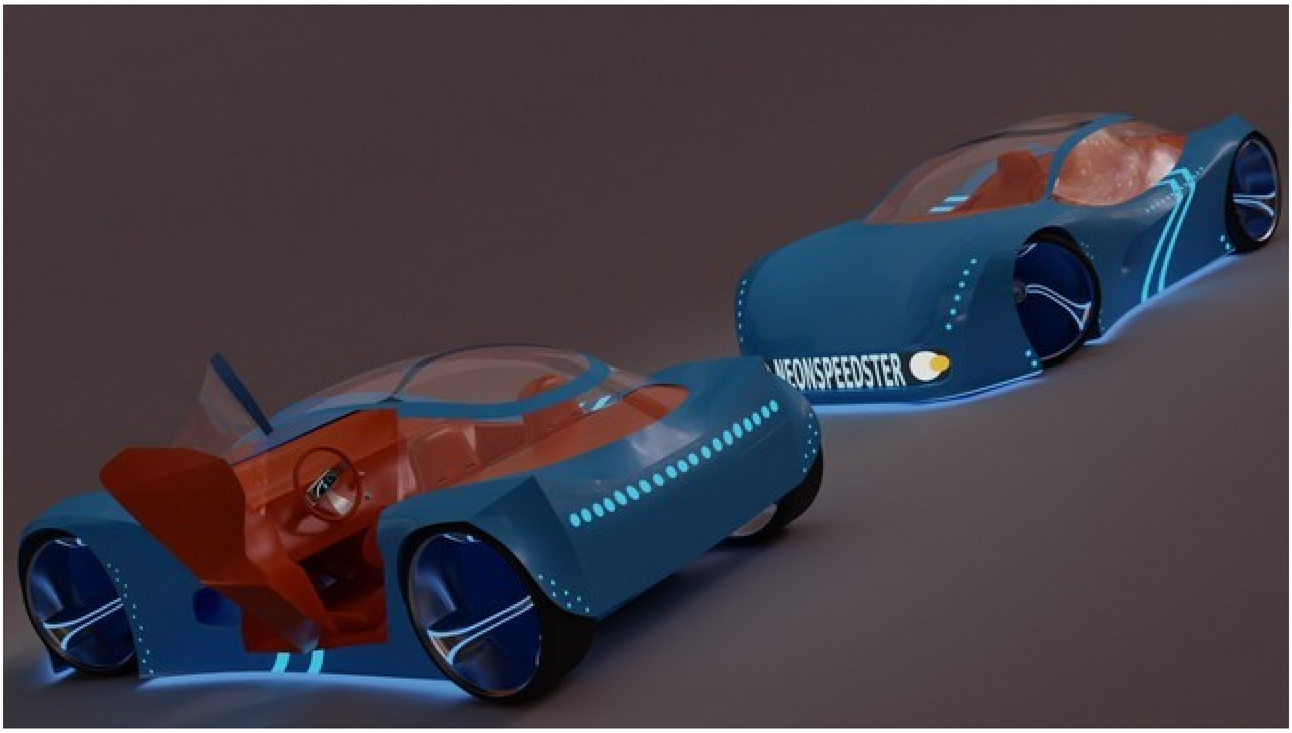
When it comes to the body it self. It's something that's your wild imagination will produce. In my work on this film I was trying to minimize the vertices and the polygons in the model. To make the Subdivision Surface Modifier do it's magic and make the car look slick and cool.

When dealing with car body shapes. I try not to use tricks like breaking the shape into separate components. Like I did with the wheel and the threads. Because I want to present the smooth flow of one thing going into another.

The less polygons you can get away with, the better. You can also make a very heavy use of edge crease and mark sharps (those in pink on the screen shot).

Putting the wheels there ahead of the time makes it very easy to see the shape around the wheel. And preserve it's roundness. Using less polygons make it easier to fix mistakes. Easier to tweak things. Until the model looks as your heart desires.

Interior and Doors



When you have the main shape ready, making the interior is not hard. Especially for those cars, who's doors will not open during the film. Because then you don't need to cut them.

The interior main part. The floor and the walls are easy since you can cut their corners from the main shape using a Boolean Operation.

Same thing I do when "Cutting Doors". I use both the interior and the exterior objects. And create a cutter object that's just a deformed cube that's going to slice the door out of the shape of the car.

The shape of that cutting object, even tho simple. Is very important to get right. So it doesn't cut too much. And in the same time cut's enough.

Then I just duplicate the car and do 2 Boolean operations. One to remove the door shape from the rest of the car. And one to remove the rest of the car from the door shape.

Conclusion

I'm not going to write about rigging too much in this article. I guess you can find plenty of me actually doing it in my archive channel. With this article I wanted to show both the ability of Free Software to do what you need. And the way your mind should work when approaching life in general.

Other people would build every part of the car separately. Have plans and blueprints for everything. Make the threads of the wheel flow into the rest of it. Just because it's the way it is. And they never question it.

Hacking on the other hand is a way of life that way cooler then that. It's when you can figure out stuff. And find new ways to do stuff. And find creative ways to use stuff in applications it wasn't designed for.

This is why we have the Freedom 0.

> To use the software for what ever purpose at any time.

So people could experiment with what they can do.

During writing of this article I had Odysee comment section conversations about LiveStreams and Jitsi vs Zoom. People expect things to just work. Because they don't have the Hacker mentality. People expect everything to be official and "by the book". When you can have fun and explore things. Copy things. Change things. Use things in weird ways.

Assignment

I want you to get a copy of Blender and play with it. Maybe try to make a car. And post the images you created in the comments section.

It's quite simple. You upload you image to Odysee. As I did with all the screen shots in this article. Then you right click on it and choose Copy Image URL.

Then use this *(markdown)* syntax

```
![](https://example.com/image.png)
```

when writing the comment to hack the image to appear in the comment section.

Happy Hacking!

.. But Those Few Channels Are Not On Odysee Yet... (How to watch YouTube in Freedom?)

YouTube – used to be a place where people publish their video-files and other people watch those video-files.

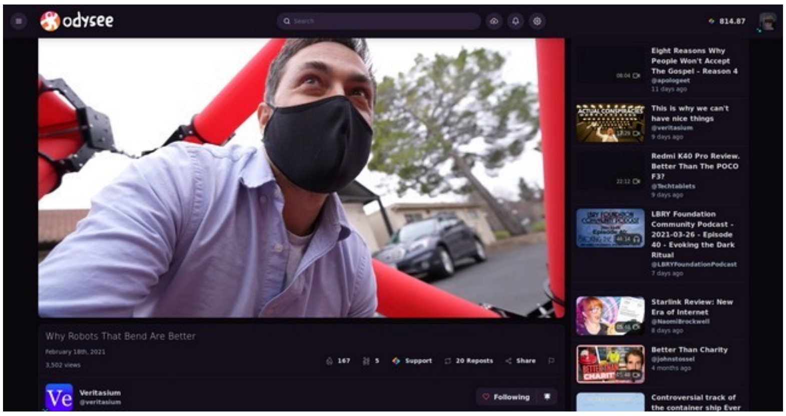
YouTube – A highly censored advertising platform that collects your data and does various other nasty things to the users.

`lbry://@blenderdumbass:f/but-those-few-channels-dont-have-odysee-yet:9`

We all love LBRY protocol, Odysee and Free Software. And we all love that something is here finally to get away from YouTube. But there is a problem. Some channels are still only there. And you want, very hard to see their videos. But you can't, since you will run YouTube's proprietary JavaScript if you would. But you don't want to loose your Freedom. Not again. No... But you want to see their videos so much.

Is there a way to view their videos without loosing freedom?

Alternative 0 (Odysee)

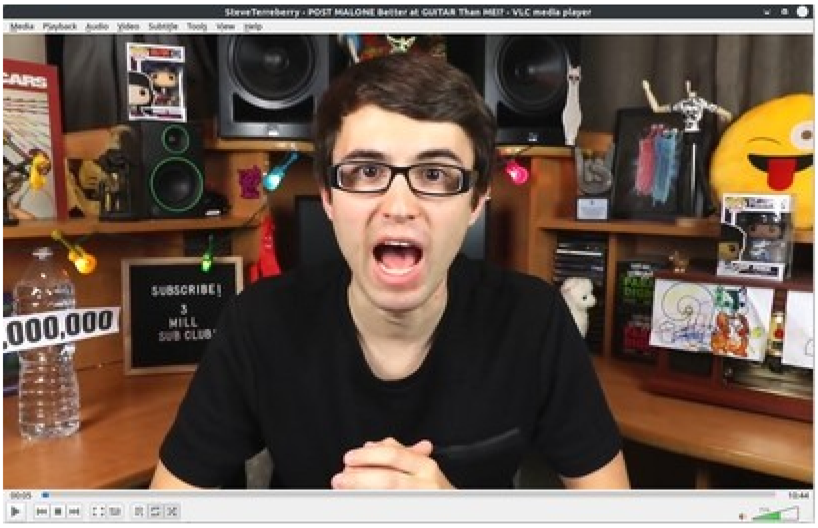

odysee
Search
814.87
Why Robots That Bend Are Better
February 18th, 2021
3,502 views
167
5
Support
20 Reposts
Share
Veritasium
@veritasium
Following
Eight Reasons Why People Won't Accept The Gospel - Reason 4
@apologeet
11 days ago
This is why we can't have nice things
@veritasium
9 days ago
Redmi K40 Pro Review. Better Than The POCO F3?
@Techtablets
9 days ago
LBRY Foundation Community Podcast - 2021-03-26 - Episode 40 - Evoking the Dark Ritual
@LBRYFoundationPodcast
7 days ago
Starlink Review: New Era of Internet
@NaomiBrockwell
8 days ago
Better Than Charity
@johnstossel
4 months ago
Controversial track of the container ship Ever

Many YouTube channels have synced their accounts with the LBRY protocol. Some for backup. Some for crypto-currencies. And some just because they care about people's Freedom. So before going to YouTube and looking for stuff there. Take a second.

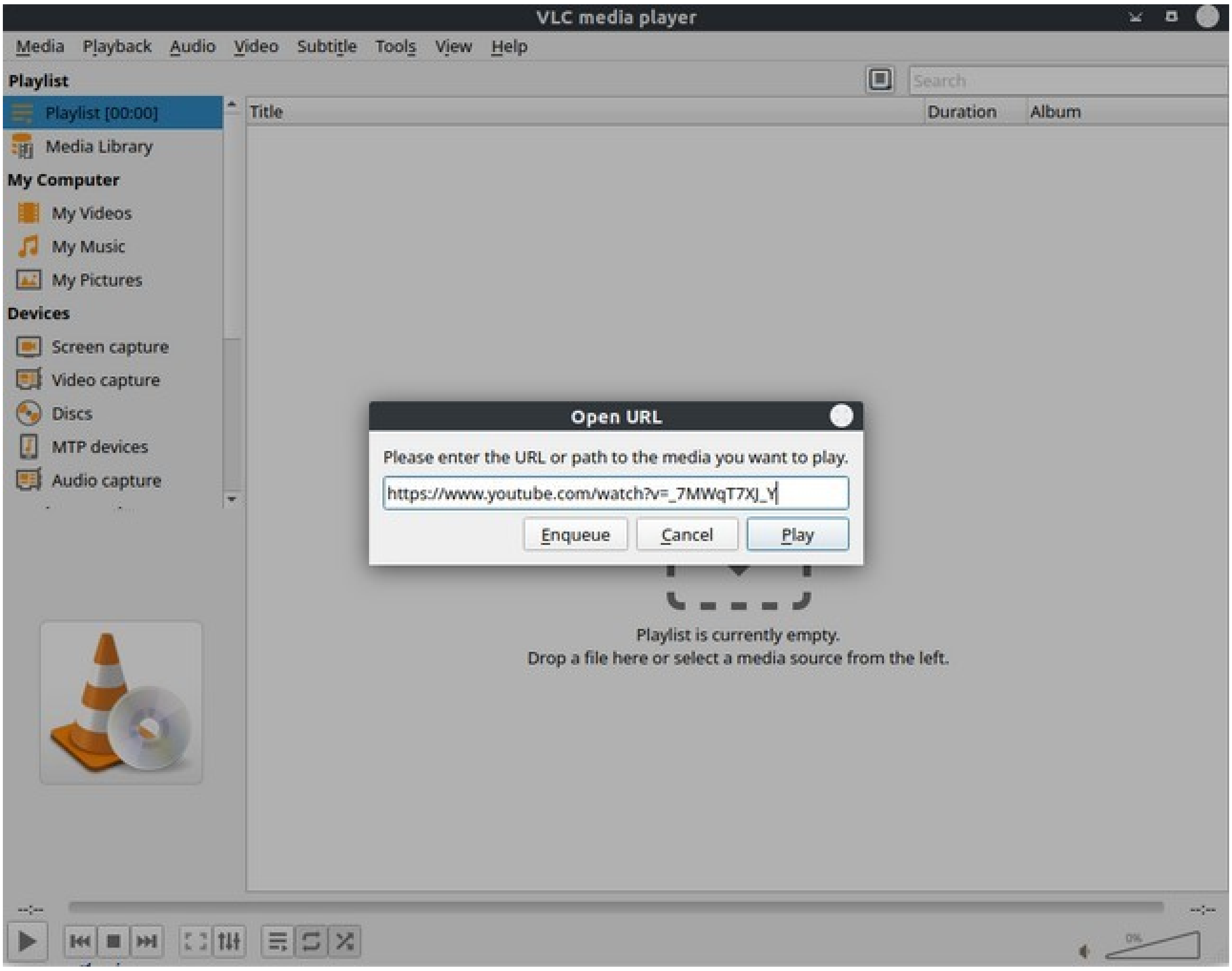
Look if the channel is already here. Look if the video is already here. What if it is?

Also there have been projects that auto-redirect you from YouTube to Odysee if the synced video or channel exists on the LBRY protocol. Noticeable one is Watch on LBRY that's available as a plug-in for both Chromium Based (Chrome, Brave) and Firefox based (IceCat, Tor, LibreWolf) browsers.

Alternative 1 (VLC or MPV)


SteveTerreberry - POST MALONE Better at GUITAR Than ME!? - VLC media player
Media Playback Audio Video Subtitle Tools View Help
SUBSCRIBE!
3 MILL SUB CLUB
000,000

In both VLC and MPV *which are Free Software* you can use a YouTube links to play a video from YouTube. In VLC you can just paste it directly into the main window and will start playing the video.



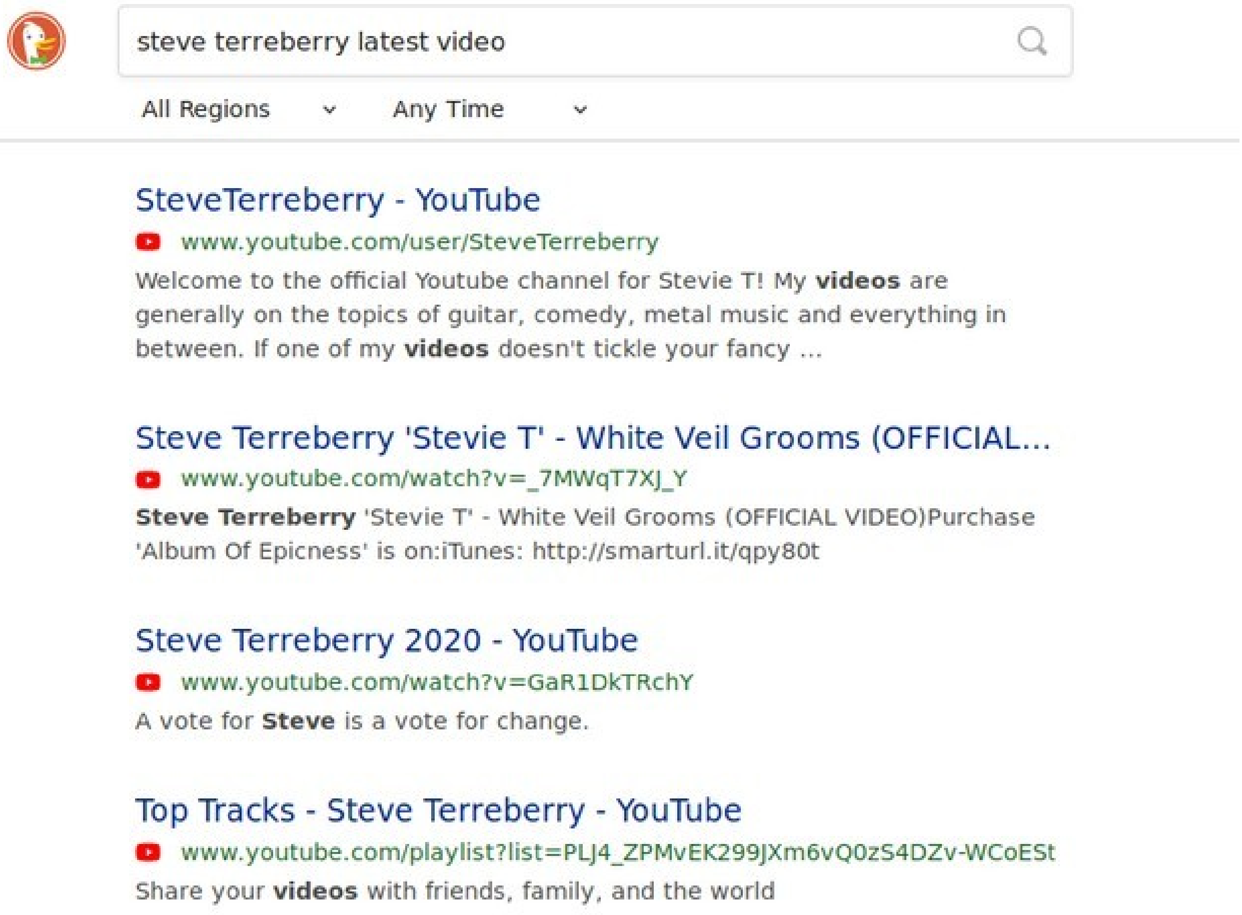
Please enter the URL or path to the media you want to play.

https://www.youtube.com/watch?v=_7MWqT7XJ_Y

Playlist is currently empty.
Drop a file here or select a media source from the left.

Keep in mind that some videos in the YouTube system are guarded more the others. Especially the music related stuff. So some videos will refuse to download or work if you don't have their JavaScript code running. This is Defective By Design.

Of course to play videos like that you will need to find a YouTube URL link in some way. Going to it's front page doesn't make sense. It's still uses hell of a lot of malicious code just to draw the search / recommendation thing. So I would suggest using a different way of finding a link. Perhaps using DuckDuckGo to find it.



SteveTerreberry - YouTube
www.youtube.com/user/SteveTerreberry
Welcome to the official Youtube channel for Stevie T! My **videos** are generally on the topics of guitar, comedy, metal music and everything in between. If one of my **videos** doesn't tickle your fancy ...

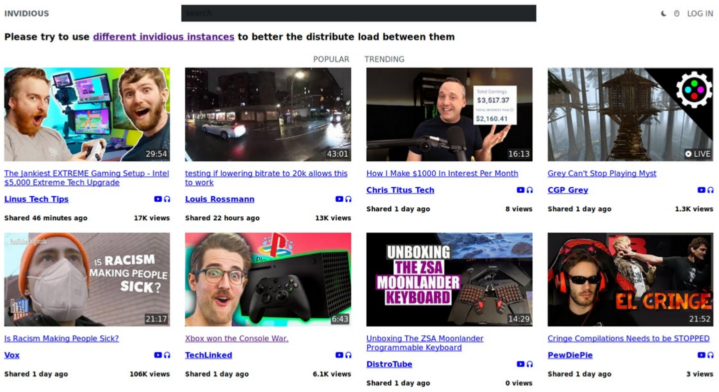
Steve Terreberry 'Stevie T' - White Veil Grooms (OFFICIAL...
www.youtube.com/watch?v=_7MWqT7XJ_Y
Steve Terreberry 'Stevie T' - White Veil Grooms (OFFICIAL VIDEO)Purchase 'Album Of Epicness' is on:iTunes: http://smarturl.it/qpy80t

Steve Terreberry 2020 - YouTube
www.youtube.com/watch?v=GaR1DkTRchY
A vote for **Steve** is a vote for change.

Top Tracks - Steve Terreberry - YouTube
www.youtube.com/playlist?list=PLJ4_ZPMvEK299JXm6vQ0zS4DZv-WCoESt
Share your **videos** with friends, family, and the world

Alternative 2 (INVIDIOUS)

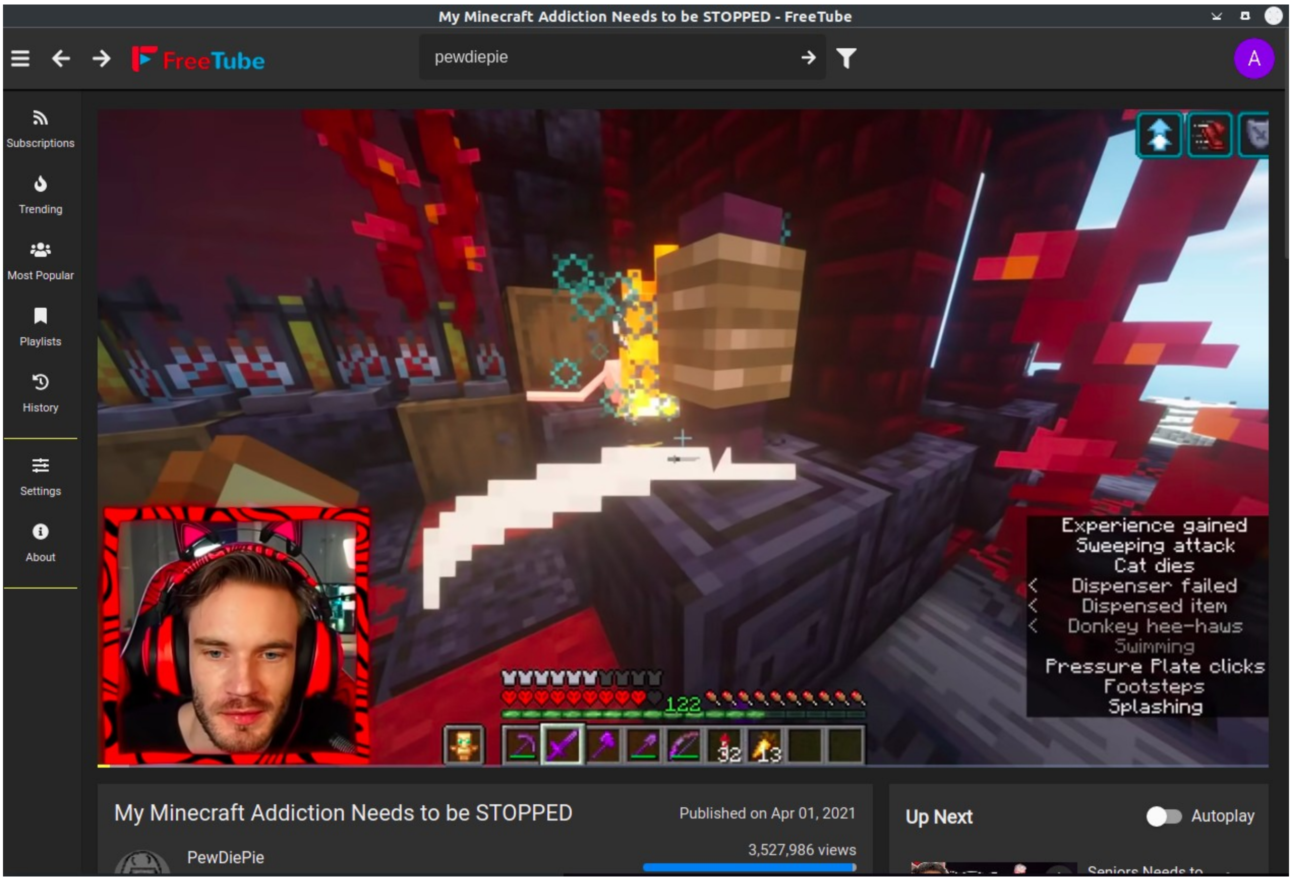
INVIDIOUS is an alternative front end to YouTube under GNU AGPL license (It's like GPL, but adds that if you are using the code on a server, you have to publish the code of the server). It's basically a set of server configurations to make your own YouTube front end server thing. Using YouTube as some kind of LBRY protocol. But for videos hosted on the YouTube's servers.


Please try to use different invidious instances to better the distribute load between them
INVIDIOUS
search
LOG IN
POPULAR
TRENDING
29:54
The Jankiest EXTREME Gaming Setup - Intel $5,000 Extreme Tech Upgrade
Linus Tech Tips
Shared 46 minutes ago
17K views
43:01
testing if lowering bitrate to 20k allows this to work
Louis Rossmann
Shared 22 hours ago
13K views
$3,517.37
$2,160.41
16:13
How I Make $1000 In Interest Per Month
Chris Titus Tech
Shared 1 day ago
8 views
LIVE
Grey Can't Stop Playing Myst
CGP Grey
Shared 1 day ago
1.3K views
IS RACISM MAKING PEOPLE SICK?
21:17
Is Racism Making People Sick?
Vox
Shared 1 day ago
106K views
6:43
Xbox won the Console War.
TechLinked
Shared 1 day ago
6.1K views
UNBOXING THE ZSA MOONLANDER KEYBOARD
14:29
Unboxing The ZSA Moonlander Programmable Keyboard
DistroTube
Shared 1 day ago
0 views
EL CRINGE
21:52
Cringe Compilations Needs to be STOPPED
PewDiePie
Shared 1 day ago
3 views

Being Free Software it's somewhat decentralized. Hosted *(the front end)* on many servers. With some changes here and there. I can recommend to look at invidious.snopyta.org. It's very light weight and mature for a front end of a video sharing platform.

For easy to remember one see **yewtu.be** .

Alternative 3 (FreeTube and NewPipe)



This one depends on whether you are on a computer or a mobile phone.

FreeTube is a Free Desktop application for watching YouTube. It uses it's own scrubbing API or you can choose to use the INVIDIOUS API as well. It doesn't give the ability to log in into google's account. It has a separate account with separate subscriptions that Google doesn't even know about.

More than that it is possible to use it over Tor. Making Google completely oblivious to your existence while you watch videos.

The code for this program is here. GNU AGPL v3

NewPipe is a Free Android application. Available on F-Droid here. It is pretty much the same thing as FreeTube but for mobile phones.

Notable difference is that NewPipe also can give you access to SoundCloud, PeerTube and a few more. Tho these are only in beta.

The code for this program is here. GNU GPL v3

Still not good enough

Complete Freedom is very hard. And some things are very hard to get rid of. So a transitional technology is needed. Things like YouTube Sync on Odysee, Wine on GNU/Linux, proprietary NVIDIA drivers on an otherwise Free system, and other things. They are meant to be temporary solutions. Not the answer for everything.

So please try to avoid using it. And instead try to bring publicators over to a more Freedom respecting platform. Would it be PeerTube, Media-goblin, LBRY / Odysee, or their own web site it doesn't matter.

Using YouTube as a host, even if the front end is Free Software. Is still not really good. It has this rotten feel to it. Because it makes the YouTube bigger. You still use YouTube, technically. Even tho not through their official means.

You want videos to say "Follow on Odysee" instead of "Subscribe on YouTube". You need challenges to be "If the video gets 10 thousand LBC" not "If the video gets 10 thousand likes". You want people who are still on YouTube to feel like they are the weird ones. So they will free them selves. And we never will need transitional techniques I described in this post again.

Happy Hacking!

Best Encryption Algorithm (that doesn't exist)

What do programmer think about while in the shower?

lbry://@blenderdumbass:f/but-those-few-channels-dont-have-odysee-yet:9

This is not an article about a real algorithm. Nor I know how to write it. This is me talking about ideas, wishes. Things I see that could be, possible done. And a little speculation on whether it's even possible.

What is encryption?

Encryption is a process of making information more secure. More private. Let's say you have a file with your passwords in it. And you want to make sure nobody but you *(or those you trust)* can read this file.

Encrypting is to make the information in this file look like it's something else. Usually like random string of gibberish. So even if somebody finds it. It will make no sense to him. But will make sense to you.

You can have various levels of encryption. For example just merely speaking a different language could be encryption. Let's say you live in Israel where every speaks Hebrew. If you happened to know Polish for example. You can write down things in Polish. And only other people who know Polish will be able to read it. Meaning nobody from your immediate surrounding could understand it. Tho it's not a very good encryption.

Using ciphers could be a better way. But ultimately **encryption is making one information look like another information**.

Problem I'm trying to Solve

Imagine a situation where you have sensitive data on a file somewhere. And the encryption is very strong. Nobody can break the key but a lot of people want to do so.

Hypothetically it's something of interest to some very dangerous people. They hunt you down. They catch you. And the only way to get away alive from the situation is to give them the key. But in the same time. You don't want them to get the information. So what do you do? You are cornered.

My idea

You give them a key. It unlocks the file. They get what they want. And let you go free. But it wasn't what they wanted really. It just seemed like what they wanted. You gave them a special, wrong key.

Which unlocked the file. But presented them with a different information. The information they got wasn't gibberish. I made sense. It was very close to what they were looking for. But not exactly.

Encryption is making one information look like another information.

Why can't it be possible to have an algorithm that encrypts data in such a way that you have multiple keys. Each of them unlock different information. And you choose ahead of the time what information will it be.

For example. You have a photo of some international spy. And a photo of a random stranger. Depending on the key it will give you either one photo. If it's somebody you trust. You give them the first key. And he get the photo of a spy. If you are cornered. You give them a different key. And they get the photo of the stranger.

Wish list

With such an algorithm I have a few things I'd like it to have in the implementation.

- **More then 2 files / keys.** It's important to have more then only 2 keys. For example you can have various security clearance levels. Each having they own key. Each of them can unlock the same encrypted file. But get only their dose of the information.

- **Clever compression of size.** It's probably going to be possible to trace the amount of keys in a given file by dividing the encrypted file by the output file. If you have 3 images in there, the one image you've got will be 3 times smaller then the whole file. Meaning those people who cornered you have a way of knowing whether you still have some keys you know about that you didn't tell them. The file should use the same bites of encrypted data to store multiple images. The key should be one that decrypts the data into either image. In today's algorithm you get either readable data if the key is correct, or gibberish if the key is wrong. What if instead of gibberish you just could get something else that's also readable.

Compression?

If the 2 wish list items from above be implemented?

For example, you can use it for compression. You could cut a big file into similar sized chunks. And store a key of each chunk with the encrypted file the size of one chunk. The trade of will be the amount of keys and their lengths compared to the encrypted file. Probably you can make the encrypted part represent either 0 or 1 in one bit of binary data. And with long enough keys, de-crypt it into a large file. But it's an extreme case and probably storing those keys will take more space than the original file.

How?

I don't know. I tried ones to build it by shuffling two encrypted files in between one another. Like bite from one then bite from another, then bite from one and so on... This was super slow for anything reasonable. And not really what I think should be done.

Using a Caesar Cipher for example. We can put IF and LI together. But it doesn't scale up from that very well. If you extent LI into LINUX it will give IFKRU with the second key. Which already is gibberish. *(by LINUX I meant kernel. The OS is GNU / Linux)*

I don't really know how. But it would be interesting to find a way. If you know. If there is a Free Software that does it. Even non-free software (we can do reverse engineering on it.) Please let me know. Comment about it.

Happy Hacking!

Should You Be Free To Run Non Free Software?

Sometimes one Freedom is colliding with another Freedom. What should be done in such a situation?

lbry://@blenderdumbass:f/Should-you-be-free-to-run-non-free-software:9

Today I had a conversation with a very bright, young lad that made a point to me that I could not just simply disregard. We were talking about Free Software and he asked me why I need FreeTube? And why I don't just use YouTube directly?

As I started pointing out to him the problems with their JavaScript code. He told me something interesting. *(translating from another language)*

> Your obsession with Freedom makes you unable to use things that other people are Free to use. Making you less free. Because you are not allowed to use those non-free things anymore.

This is an interesting point. Is me using exclusively Free Software contributes to me loosing freedom?

Should Proprietary Software be Illegal?

This is one of those things to which the answer will change whether you are using "Free Software" to describe Free Software, or you use "Open Source". I think a lot of "Linux users" are thinking only about reach. Thinking of Free Software (as they call it

"Open Source") as a direct competitors to proprietary ones. And it becomes the "us or them" situation. The "Open Source Purists" would argue that. Yes. Proprietary Software should be illegal. This makes total sense.

But what does a person who invented this whole thing, Richard Stallman has to say about this?

In this interview with Richard Stallman. Bryan @Lunduke asked him a similar question. *(It's right in the end of the video)*

Bryan

> Richard, I know you are a hungry man and I want to let you go really quickly. I have one final question for you. If, right now, today... the countries of the world banded together, and passed a unanimous law, outlawing all proprietary software and made Free Software the de-facto. What would you then do with the remainder of your days?

Richard Stallman

> Well. First of all I don't advocate for making all proprietary software as such illegal. There shouldn't be any proprietary software. It's harmful to people sort of the way cocaine and meth are. But you know what the war on drugs does? Just making a dangerous drug illegal doesn't get rid of the problems. So while I hope everyone will avoid using proprietary software just as I hope they'll avoid using meth. Prohibiting it is not such a good solution. We got to teach people to get off of it. In any case if we solved the problem and get rid of the injustice of the proprietary software, what would I do? Well if there are still other injustices in the world I'm sure I'll be able to make some contribution to putting an end to them.

So here the man, the legend, the inventor of the Free Software movement, Richard Stallman him self says that he doesn't advocate for making all proprietary software as such illegal. Why? Doesn't he want us to win?

The point is that if you look at "open source" from the Free Software perspective. (BTW the "Free" that everybody has a problem with, because of 2 meanings in English. I speak few languages. Only in English it has 2 meanings.) Free in this case means Freedom. Not Gratis. Freedom to both make, and use any type of software should be present.

If so, why don't you use Proprietary Software?

I'm not going to talk in the name of Richard Stallman here. But we both probably have similar reasons of why. And it's the same reason of why I would not eat poo.

Should eating poo be illegal? I think. If you are into it. That fine. Do it. Would I do that? No. Does it make me less free because I would refuse to eat poo? No. I can always do that if I'd want to. I just know that there is a high probability that I wouldn't want to eat poo.

Should you be free to use non-free anything, is like asking, should you be free to become a slave. It's a paradox. It's like asking. Can god make something that he couldn't move?

If you have 100% freedom. The freedom will include giving away your freedom. Which will decrease the freedom and you are no longer at 100%. So to stay at 100% you should not be Free to give away Freedom. Which makes you less free. And you are never on 100% freedom. So why bother?

But this is wrong. Even if you have the freedom to loose freedom. You can choose not to act on it. And that's what I'm doing.

I have Brave installed. While I use IceCat with LibreJS as my default browser. Why do I have Brave? So I could access non-free sites. That are either blocked completely or partially by LibreJS.

Do I do that often? No. But sometimes there is a page or two I would like to render properly. It's like if I was a drug addict. And I took a lot of drugs. But now I take a tiny amount ones in the while. Trying to get rid of the addiction.

For example. Just about 2 or 3 months ago. I would watch YouTube full time. Right on YouTube's official website. Some time later. I would still upload to

YouTube here and there. I uploaded this video to YouTube too. Using their proprietary javascript.

Up to a few weeks ago I would load YouTube's front page to get links of videos. To open them later in VLC. Now I use FreeTube to watch YouTube videos.

Do I feel good about it? No. I feel rotten inside that this platform made me so addictive. I wish I could live Free from it. And I visit it less and less. So I make a lot of progress.

The worst thing for an addict that you can do is to prohibit per's addiction. The best. Is to educate per that there is a problem.

Alcohol

Driving while drunk. Alcoholism. Alcohol is pretty bad. It's not something I would promote for consumption. But prohibiting it doesn't make sense. As soon as something becomes illegal. There is market for those who are willing to spread it illegally. And this market is not regulated. So making Alcohol illegal will result in worse alcohol being sold for more money.

You are Free to consume alcohol (if you a certain age.) But most people know that it's harmful.

Why can't this philosophy be applied to non-free programs too?

You are Free to use non-free software. But most people should just know that it's harmful.

What's than all the gnu.org/distros all about?

There are a few GNU / Linux operating systems that are designed to provide as much Freedom as possible. Those listed in the gnu.org/distros. A lot of people criticize them for making them so it "doesn't let you to install non-free software."

This is wrong. The distros just don't have non-free software repositories added and signed by default. It's should be done manually if you want to install non-free software.

Similar to LibreJS. It blocks all non-free JavaScript code by default. But you can whitelist those you want to run manually.

What makes people upset is that their usual software, they expect to have and run and easy to install. Is no longer easy to install. It's not like you go to Ubuntu Software Center and search for Discord and click Install. And you have it.

Discord in non-free software. To install it on a 100% free system you need to work a bit harder. Since they are not going to provide the repository for it by default. You will have to do it manually.

You have the freedom to install Discord here. As you have the freedom to install Discord there. On one place it's just going to require more steps.

Of course many popular distros come with non-free software preinstalled. In this case having Freedom will require more steps.

Conclusion

Should you be free to run non-free software? Yes. You should have Freedom in all things and all the time. Should you be forced? No. Power and Freedom are not the same. And nobody should be able to force you to do nothing. Is it good to use non-free software? Probably not. Maybe in few instances it makes sense. Like reverse engineering, or security research. Or "Using the evil to stop the evil is okay." that I mention here.

Freedom is important. Non-freedom is bad, perhaps not everybody realize it. Perhaps time is needed to get rid of non-freedom. Perhaps sometimes eating poo is the only way out. But let's try not eating poo.

Happy Hacking!

We Need More Free Software GNU/Linux Games!!!

A lot of people buy computers for games. It’s just a reality of life and there is nothing to do about it. But, this fact can be used to do something good.

`lbry://@blenderdumbass:f/Wee-Need-Free-Software-Games:7`

GNU / Linux is getting better and better at being a Gaming Platform for an average user. Valve and Steam with Proton and Wine is making it possible to play Windows games on GNU without too much problems and in a reasonable performance. **But is this really what we need?**

Problem 1: It already works on Windows, so why switch?

A lot of people have their Windows installation and play *all* their games on Windows. If there is suddenly an ability to play the same games but on a different platform. Many people just don't have a good enough reason to do it. Even more. Spend time making a live USB, deleting all the files from the computer. Installing a completely new operating system, they are not familiar with. And only then install the same games they had on Windows. Just to play them on a different platform. This doesn't make sense.

Problem 2: Most popular games are DRM'd proprietary software

Even if people have the ability to easily install and play games on GNU / Linux. To have similar or even better experience than on Windows. It's still doesn't help if the games them selves are proprietary.

The primary reason for people to switch to a Free Software operating system would be to get freedom. What is there to gain if you just change one program. You do the same amount of progress using any other one Free Software but on Windows.

If you are using Odysee and doing it on Windows. It's one program. Non-Free YouTube was changed to Free Odysee. The rest stayed the same proprietary software. The same goes for the proprietary games. If you play them on GNU / Linux. It's one program. Non-Free Windows was changed to Free GNU/Linux. The rest stayed the same proprietary software.

There is an article about it by Richard Stallman that goes a bit more in depth about this problem.

Is there anything good in having proprietary games / software on GNU / Linux?

Yes. Its called a transitional technology. A lot of people need time to adopt to the new setting. And the best way to do it is by having as much familiarity as possible. This is one of the reasons that *GNU /* Linux Mint is so popular. It's default layouts usually borrows the aesthetics of Windows. Having the panel in the bottom. A start menu in a predictable place. And having a good selection of software that comes with the system preinstalled.

Together with it. Strategical decision was made while making the GNU system to allow proprietary software to run on it. Stuff like Wine was developed so Windows users could keep using their familiar software on GNU. Stuff like IceWM were designed to mimic the look and feel of Windows so people would feel comfortable using a Free System.

Libre Office (Open Office), Gimp, Blender and other Free Software projects focus heavily on being able to work with proprietary software and proprietary formats. So people would be able to re-use their presets. To move one step at a time. Instead of being forced to jump over or be alienated by it being completely out of reach.

The downsides of it

The real downside of this transitional technology existing is that people who are free are being encouraged to try non-free software. Because the easier it is to install a non free game. The more people will just simply do that. And it's true for both sides. Either you are a Windows user. Or a GNU/Linux user.

The only people who will not fall for this trap will be the Free people. That value their freedom over all. People that value their Freedom more than a desire to play a given game. Or use a given peace of software.

What do I propose?

I'm a Jewish guy. And in our Chabad tradition there is ritual. When Napoleon was going into Russia. It was decided by Rabies that he shouldn't win the war. But Russians should. So the Ritual was to start singing Napoleon's hymn. If a non-Jewish song becomes Jewish song. It's believed to rub the non-Jewish people who use that song from their power. I don't know whether to believe it or not. But they still sing the Napoleon's hymn quite often and I like the idea.

To use the weapon of the enemy against the enemy.

Sony is a terrible company. They called police on a person that was trying to change software on per's Sony Play Station device. They are strong supporters of DRM. They make Jail devices that allows people to install only the software approved by Sony.

Also Sony has a Gaming platform that is not Windows and that is popular. People buy Sony Play Station for the soul purpose of playing games. Even tho they probably already own a Windows machine. What is going on here?

Have you heard of "The Last Of Us"? A game that requires Sony Play Station to play it. You can't play it on an Xbox. You can't play it on Nintendo Wii or Nintendo Switch. You can't play it on Windows. Or GNU / Linux. It's only available on Sony Play Station.

Well not true. Because hackers hack. And people made a Play Station 3 emulator for GNU/Linux that *can run games like The Last Of Us.* The intention was to make games that only work on Play Station so people would buy Play Station to play those games.

Same goes with Nintendo games. That are available only on Nintendo consoles. Making you buy Nintendo consoles to play those games.

So why not make games that are available only on GNU/Linux and make them Free Software to promote both causes?

How would a Free Software game be available only for one platform?

You may think that this approach makes no sense. Since the Software of the Game will be Free and every body could just make a Windows port of it. And yes it's True. This is a possibility. Especially if the game will be very popular.

But it's not necessarily going to happen. Or going to be easy. There is plenty of Free Software out there that is not as popular as the Linux Kernel. And that are available only for GNU / Linux. Mainly for a few reasons.

Dependencies. A lot or software is built on dependencies. And only if you design the software to avoid dependencies as much as possible. To design it in the first place to be easily portable to other operating systems. The porting will require either porting the dependencies too. Or to make an alternative.

A good example would be Windows games that need DirectX to render the image of the game. To make those run on GNU/Linux you either need to port DirectX to GNU. Or to make a Free replacement. Let's say you are using OpenGL or Vulkan to render the images. They would still need to have a translation layer. To catch the DirectX commands and to translate them into OpenGL command or Vulkan commands. On GNU / Linux there is Wine and there is Proton that do that job.

So if you design a game with as much dependencies on GNU software as possible. Or things like Vulkan, GTK, GLib and other things that are not available on Windows. Or hard to make it work on Windows. It will give it more exclusivity. While remaining free.

Since all the dependencies will be Free Software (to make the game Free) you can technically port each of them one by one to Windows and with it port the game too. But it's going to require a fundamentally hard work to do it.

The point here is that making a live USB and installing an operating system would be a much simpler thing to do to play this game. Than to port it over to Windows.

Official. Even tho Free Software is Free Software. People unfortunately, or fortunately want to use official stuff if they can. Most people will always get the main branch of a given peace of software. Unless they are nerdy and know what they are doing. Or it's the Linux Kernel.

Blender was sold by some left companies. I've seen some of those sites myself and cringed very hard. Some of them even renamed Blender into their own thing. It didn't help. Ton Roseendal's Blender is the official one. And people don't want to use those official ones.

The point here is if we make the official game be built for GNU/Linux. And state it in a big, bald text. So people will know that to get the official experience. You need to play it on GNU/Linux.

Laziness. Ardour is a Free Software but payed. The source code is gratis. But the compiled executable is payed. It's not the only Free Software that does it. And it's not wrong to do so.

But you are asking. Why not then build from source? And yes. Some people do that. They get the Gratis source code and build Ardour from it. But most people do not have the technical knowledge to build software from source. And Ardour is not your easy build. So people will pay for the convenience of having a pre-built executable.

The point here is to make the official pre-built executable to be only GNU / Linux compatible. Combining all 3 *Dependencies, Official and Laziness* together. Makes it possible to make a GNU/Linux exclusive, Free Software game.

Incentive?

What would be the incentive to build a completely Free Software (and probably Gratis with it) game like this?

I don't know. Maybe a clever person can come up with some business model that could be used. Hell. Why not crypto-currencies? LBRY / Odysee pulled it off very well. But even if the money side of things doesn't work out there are still a few things that could keep you motivated to do such a project.

And fortunately those reasons are already published here. So I don't need to rewrite them.

Conclusion

We are at war. Freedom vs Power. Companies want the Power. People want the Freedom. People who are not on either side can join either side. So let's make them join ours. Let's make them be for Freedom.

To do it we need to give those people who don't care a reason to care. A way for them to look our side and be exited. Good Free Software, GNU/Linux exclusive games are one of the ways we can bring a lot of people over to us.

Not the only one. Privacy, Security, Transparency also work well. But it's seems like those people who care about there issues are already on our side. And there are others. And you can help bringing them to us. So we will win the war and get our Software Freedom.

Happy Hacking!

Should you Aim For Success?

A lot of people have only one motivation in life – money. How about using this drive but adding just a little bit more meaning into the work that you are making?

lbry://@blenderdumbass:f/Should-You-Aim-For-Success:8

We all want something in our lives. Many of us want success in this or that area. Successful Odysee channel. Successful business. Successful software. Successful game. Successful movie. Successful idea.

What if I told you that success is not the most important thing in the world? What if I would tell you that sometimes some things are more complicated than this.

Devil in the details

I love details of things. Small things nobody is noticing or knowing. I like them. But I know that a lot of people are just not interested to think in detail about anything.

Today I heard a lady that was buying in a store. She was choosing cheese. And there was one type of cheese or another. And she needed to figure out which one is better. Somebody told her a recommendation. But not the kind of one you would hear often. Not something in like "I prefer this cheese". But something along the lines of "It depends on what is it you are going to use the cheese for. And depending on this and the details about the cheese. You can find the optimal solution to your problem". Her answer was something I can't forget. She said "No, it's too much thinking."

People don't like to think. They just want things to be in either one of 2 baskets. Either it's good or it's bad. But with most things in life it's not that simple.

Take for example a knife. Is it in a basket "Good" or is it in a basket "Bad"? You can cook with it and feed people. Good? You can cut people with it and hurt them or even kill them. Bad? The answer is it's both. But the devil is in the details. It already depends on any particular situation.

Too much details

There is a famous situation that happens probably to a lot of parents.

Kid

Why are you going to work?

Dad

Because we need money.

Kid

Why do we need money?

Dad

Because we need to buy food.

Kid

Why do we need to buy food?

Dad

Because if you don't eat you will die.

Kid

Why if I don't eat I will die?

Dad

Because, I don't know why. Because I said so.

And similar situation can happen no matter what would be the first "why". Or what will be the answer. You can always get another why. And reach the end of somebodies understanding of a situation. Or never reach the end.

This is a problem. Since if you want to think deeply about a given topic you need to think on some level of details. But where do you stop? How many levels of details you need to be at to solve a particular issue?

There is a famous quote by Albert Einstein:

> We cannot solve out problems with the same thinking we used when we created them.

Is this talking about what we are talking about? Is it the answer to knowing the level of details needed? Is the level should be one greater then the one you happened to have the confusion on?

Let's test it. Coming back to our knife problem. Level 1. Knife is either bad or good. It's good. Since it helps me to cut vegetables for a salad. Somebody kills a man with a knife. Is the knife actually bad? We are still at Level 1. The answer we are looking for is not on this level. It's not a question of whether to ban a knife or not to ban a knife. But what is the question really?

The real question is "How is it that the knife is both good and bad?". And here. This is where the level is going to increase. Now we discovered the Devil in the details. Our attention was brought to a detail. What do you do with the knife? What if action 1 is good and action 2 is bad?

And so the conclusion here will be. If you cut vegetables. It's good. And if you kill a person it's bad. But you need to keep in mind that those can be split further. It's just not important to go into more details to solve the issue at hand.

Is success good?

Yes. If you are aiming for business success. You will have more freedom. You will be able to feed yourself and your family better. If you are aiming for idea success. You will be heard. World will be a better place. And so on and so forth.

But what if your business subjugates people? What if your business kills people? What if your idea is wrong and potentially dangerous?

If we just left the question at this level. It would start an endless stream of hate. There will be a Letter For Support. A Letter for Dismissal. People who agree. People who disagree. It becomes polarizing. Dividing.

So "Is success good?". Or even "Should you aim for success?". Those questions are not the ones we really need to answer to solve them. We need to get one step beyond. We need to look for the details.

Capitalism vs Freedom

There was a great talk by Todd Weaver about why the Purism company was registered the way it was. To summarise his argument. (And I urge you to watch the whole talk) The reason big companies track people and sell data and lock devices and do nasty stuff like that. It's because it's nearly illegal for them not to do anything they legally can to maximize shareholders value.

Companies have shares. The main guy in each company. So called "Owner" has the largest percentage of shares. Usually about 51% not to risk being out-bought by somebody else. Other people might have shares too. Those are called share holders. They buy shares. And they can sell them.

To overly simply the whole thing. The more profit the company does. The more each shareholder gets payed. And the money is divided by the percentage of each shareholder. *So the owner gets 51% of all money company gets.* (And again it's overly simplified. It's way more complex then this. I'm just illustrating the point.)

Basically. If a shareholder buys a share. And sees that the company makes a decision that decreases the value of the share. The shareholder can sue the company. And will probably win. If the company is a **For Profit** Organization.

For Non-Profit and Social Benefit Organizations the rules are different.

So let's say Google comes up with a way to make tons of money by selling ads. But it requires tracking people. It's illegal for them not to track people.

Why did this happen? The whole system of business to begin with was to succeed in making money. So their aim was to make money. Their success is if they make money. Whether they respect their "customers" is a side issue for them.

Even if you start by being respectful. Like Google having "Don't be evil" in their code of conduct. It wasn't helpful since the aim wasn't it. The aim was to maximize shareholder value. The aim was to make money. Not to "Not be evil".

The real question. What should be your aim?

Success is proportional to the aim. If the aim is money. Success will be determined by the amount of the money. If the aim is fame. Success will be determined by how famous you are.

In Mark Manson's book The Subtle Art of Not Giving a Fuck he goes even one step beyond into the details. And argues that what you really need to do is to divide aims you control versus aims out of your control.

For example. If your aim is to be more famous than Metallica. It's very hard to control it. So it's a stupid aim. You can get lucky and be more famous than Metallica. And than it will be your success. And will make you happy. But it's not possible to control it yourself.

But if you are aiming to be honest for example. It's totally in your control. And you will be successful if you are being honest.

Values

Before you can figure out the success. Or the aim. The question is. What do you believe in? What are your values. Because while a lot of people will choose money as their values. It's probably a shallow answer. Money is not most peoples values.

I value Freedom. Freedom of Speech. Freedom of Software. If I have 100% Free Software operating system. I succeeded from my point of view. But there will be more success for me if everybody else will have 100% free software. If all software will be free. If all people would have the freedom of speech.

Some people might choose as a value being a "Linux User". (Not a GNU/Linux user) For them using Windows or Mac or BSD is not a good thing. Because it's not coming with a Linux kernel.

In September 2019 Richard Stallman gave a talk to Microsoft. He asked the vice president of Microsoft whether they are willing to release Windows under the GNU GPL license. And the answer was that it's not totally impossible to do so. So they might make Windows a Free Software operating system.

Imagine this. Windows from tomorrow becomes Free Software. This is not a "GNU/Linux disto". This is not a "BSD distro". It's the same old Windows. But suddenly it's Free Software.

If your values were the same as mine. You would not object against using it anymore. If you have your values set to make "Linux" a big thing. Using Windows would be a problem still. But using *nearly* all proprietary Android or ChromOS will not be an issue. They have the Linux Kernel inside.

Conclusion

Should you aim for success? Yes. What success should you aim for is important tho. If you aim is not things like Freedom or Privacy. Than you probably would not have a lot of Freedom or Privacy. If you aim is unrelated. You might not care to loose those.

But if you aim for Freedom. For a right to share. If you aim for understanding between people. Ability to express one's thoughts without being persecuted for them. If you aim for goodness. Then goodness will come. But it's should be The Aim. Not the side issue you are caring about somewhat. **Happy Hacking!**

At Any Time Something Bad Can Happen

This article will talk about an uncomfortable truth about life. But the truth that should be taken seriously.

lbry://@blenderdumbass:f/At-Any-Time-Something-Bad-Can-Happen:1

On June 22, 2015 James came to his cool and amazing Tucano plane. To take off one more time. This time from the Camarillo Airport. James was a musician. He had a lovely wife Sara Elizabeth and they had two daughters together. This time tho. He would crash in the Los Padres National Forest. And by June 25th it will be official that James dies of an accident.

This James is James Horner. The composer for such films like Braveheart, Titanic and Avatar. Among other good films where he showed a true talent at making film music.

Plane Crashes, Car Accidents, Fires, Being Robbed, Beaten Up, Killed in a War-zone... Those things we know exists. It's **not** the same thing as being killed accidentally by superman while he shoots bad guys with his laser eyes. Those things are real. Those things happen a lot to a lot of people and yet we refuse to take them seriously.

Von Trier

It's not a secret by that point that I had a very strong depression from the film The House That Jack Built directed by Lars Von Trier. I saw the unrated cut in the cinema in Jerusalem. I had to take a bus there since I live in a different city. It wouldn't show up in the regular cinema. It was too gross for them. There was a story that a large amount of people walked out of the premiere because either couldn't handle what they saw. Or just were protesting the film's scenes.

This movie wasn't that hard on most people that I know watched it. Because they failed to realize what exactly did they see. They failed to understand the intentions behind the scenes of The House That Jack Built.

Lars Von Trier is an interesting man. He has a tattoo on his hand saying the word "FUCK". He makes some of the darkest and most disturbing movies ever. His childhood seems to be a traumatizing mess of extreme events. Starting from being born to a Jewish mother and a Nazi Father. Growing up in a Nudist family. And so on. Which made him *scared of everything in life except film-making*. And resulted in a chronic melancholy with episodes of severe depression. (The type that is portrayed in his film. Melancholia.)

This is not a surprise that a man like this would make a "perfect horror film" from which people come out because they can't handle the horror. But it wasn't his intention. In one of his interviews he stated that the point of the movie was to shed light on some of the bad things happening in the real world. Things like murder. Child murder. People who kill people for stupid reasons such as art.

People who I talk to. When they describe the film. And the gross scenes in it. (Especially bad one is the Child Murder scene). They talk about how it's all fake. And it's all special effects. Therefor it's okay. And I should not feel anything towards those scenes. But the point of the movie wasn't to document. It was to recreate. To show people what happens really. Using fakery and effects. Yes. But to show things that really happen as authentically as possible.

Denial

People deny uncomfortable. They say "but it's not about me", "this is not going to happen to me". Until one day they enter the plane that's going to crash. Or the building that's going to fall. Or they come into a class that's going to be shot at. But until it happens. They are in denial. "It could happen to anybody, but not me." What makes you different? What makes you special?

Even special people die. Death in inevitable. Some will die natural death. From the old age. Some will be killed. Some will die in an accident. James Horner is a special person. It didn't save his plane from crashing.

Even if it's not death. Robbery. Fight. Rape. This could happen to you. Even if you are a man. Even if you are careful and strong. If it didn't happen yet. You were simply just lucky.

People even go further than to simply deny. They censor and cancel ideas that make them uncomfortable. And it is probably natural. After all it's all just defence mechanisms.

Misfortunes

This last year we all agree wasn't the best year. Pandemic started. I Lost a Job. I got robbed by gangsters on the street that took my expensive bike. I sat in prison for 3 days because police thought I did something. My good computer was confiscated. I couldn't speak to my girlfriend for 2 months. All because of reasons. My Mother decided to drink again. Which result in her death. I'm sitting with no cash for half a year already because it's hard to find a job that will let me get enough money during the pandemic. And even if I found a good job. There is no way I can get a decent computer today when all those prices for GPUs got up to infinity because of Bitcoin.

This is a shitty year. I wouldn't lie. And I feel a tiny bit of anger right now while typing it. But what do I think about now? How do I see the world from now?

My current view is : **At Any Moment Something Bad Can Happen**.

This laptop I'm typing on. It's not mine. It's my brother's. And at moment he could decide to take it away from me. The job where I work. At any moment I could be fired from that job. They don't even need a reason to just take me out of there. I can't sit with my camera open while talking to my girlfriend most of the time. Because my brother, who pays for the internet, could, at any moment, start his livestream on a proprietary platform.

I even found a way to start livestream on odysee so he would ditch his proprietary thing. But he things that LBC worth nothing compared to some gimmick thingy Amazon came with to promote Twitch. If you want to change his mind. Here is his channel. Maybe if you can write him a good comment. Or tip him enough. He will change him mind. At least I wouldn't feel that bad while he yells at his computer.

Happy vs Excited

Happiness is not the same as excitement. Most of the "good" experience people have are not happiness. They are excitement. The rush of Dopamine. Not Serotonin.

Serotonin is a Neurotrasmitter made by the brain in a calm and peaceful time. Making you able to concentrate more. And feel a slight euphoria. This is the experience of Happiness. It also acts like a kind of electricity conductor. Some brain related diseases like Epilepsy and Depression could come from lack of Serotonin.

Too much stress might cause less serotonin. Brain forces things like Adrenalin and Norepinephrine to build up to block neurons from communicating. To block thoughts that cause stress to spread too freely to other regions in the brain. It's a defence mechanism. But. You feel anxiety. You feel like you are suddenly so much dumber then you were before stress. People who experience severe version of it might not be able to do much. Depression. And in some cases it might cause glitches in the brain. Some speculate that lack of Serotonin causes Epilepsy.

Dopamine is a different Neurotrasmitter that fills your brain when you experience something "spectacular". It's the sugar of for the emotion. Dopamine is something you get when there is a cool twist in the movie. Or a nice buildup in the song. Or you did something you are proud of. It's your brains reward system.

While Serotonin is more like Food. Dopamine is more like desert. They are not the same. As Excitement is not the same as Happiness. Advertisers are very good at giving you Dopamine. And now a days you get it in such amounts that any lack of it feels like a disaster. You were eating only desert. No food. And you need to eat the desert all the time not to feel the hunger. While what you actually need is a good meal.

Freedom in Success

In 2018 when I finished I'm Not Even Human I was full of hopes. And wrote a screen play for my big tribute as a cinema director. I knew I needed some time to become a big name in the industry. So I didn't write an expensive film. I wrote something reasonable. But with a short action scene. A chase scene. Toward the end of the film. And this would be my selling point.

To make the chase scene I wanted to use CGI. Because 1. I could do that myself. And 2. It would be cheaper then doing it for real. To test that I infect can do that myself and to show the studio how cool I am. I made this little short film.

So there it is. I call the studio. And there is a woman that talks with me. And I tell about my super-genius new movie. That how I will make them millions. And then I say about my film that I already done. And she asks me "How much views does it have?". Oh no! Oh shit! I'm not famous yet. There is no views. Even tho

I spend 3 years making that stupid movie. Almost nobody watched it.

I didn't give up and made a little cringe campaign on my *then* YouTube channel. You can watch it here. But I warn you. You will die from cringe. I even tried buying ads on Facebook for I'm Not Even Human. To boost up views.

After some time. I realized that there is nothing I can do to magically grow it's number from about a thousand (on YouTube) to something like a million. So the studio will take me seriously. But what I did realize. Is that I don't really need fame that much.

I bought ads on Facebook. People actually clicked on the movie and watched it. I could see it in the analytics of the video. But non of those people actually cared about the movie. If I make a movie with a studio and it is released in cinema. Why would people go? Why would they pay to see my stuff? Who gives a crap about some random Blender Dumbass?

Moria's Race

Moria's Race is the movie that I'm "working" on right now. I'm not under any pressure to finish it on a particular date. So there don't expect it any time soon. By some magical luck. I had a copy of all the assets from the movie on a USB drive. And I'm planning to release them with the film. Maybe as a big zip file on Odysee.

I took my time to make the characters. I took my time to build the world they live in. And this time I don't pursue anything at all. Apart from maybe the message of the film. To respect children. I don't need money. I don't need fame. Maybe it's good that more people will know about the message. But if it's not going to be this film in particular. It's fine with me.

When what you care is not the number of views. The people who do show up, usually will care about your work more. It's the same old Quality vs Quantity argument. But applied in an un-expected place. I rather have 10 people with whom I can have a meaningful conversation about stuff I publish. Rather then have a million guys and gals who will forget about what I did the next day.

Conclusion

Anything bad could happen helped me to think clearly. To think about what's really important for me. Because I might loose things. I might not be able to get replacements for those things. But if I do something meaningful I can keep doing it. Even slower. Even with worse quality. Even while being bombarded by depression and attacks from all sides.

Would it be the Free Software activism. Or spreading the word for #RespectChildren. I can do that. With computer. With a film. Without computer. By talking to people. By talking to them on Odysee. By talking to them in person. What's important will stay there. And I could still do that. All the other things are just means.

It's fine to get Dopamine and Excitement. It's fine to make a video. It's fine when it's 4K and 60 FPS. It's fine when the music is mixed properly and you have a nice, professional microphone. It's fine when a game plays on max settings. But it's all not important. You can live and be happy without the desert.

Happy Hacking!

BeyerDynamic Fox | Problem With The Microphone

And to totally change the mood, let’s look at a more technical problem. Something to do with soldering and fixing electronic devices.

`lbry://@blenderdumbass:f/My-Expencive-Microphone:c`

I love BeyerDynamic Fox

When I used to make one video per day on the old Blender Dumbass channel. I needed a good audio for my voice. And I could spend some money into a microphone that was worth it.

I was slightly scared from microphones that connect via USB. Since I'm a GNU/Linux user. Things that connect to computer and have to work with the operating system are scary.

I went into a store that I love. It's the Halilit on the Kind George Street in Tel Aviv across the street from the Dizingoff Center. I bought there my first good headphones. BeyerDynamic Custom Street. They had a customization feature. Making the headphone's look theme-able. Also I bought there Marshall headphones. Both Major 2 that I gave as a present to a friend. And the Major 3 that I use currently.

This is not your typical computer store, or electronic store. They specialize in audiophiles and musicians. So I knew I would get a decent microphone there. I went and asked the cooled sales person ever about the microphone. And that I need it to work with GNU/Linux.

He thought a little bit. And then took out a box of BeyerDynamic Fox. I knew the company. And the thing looked fantastic. I looked on the box the Frequency Range of the microphone. 20 hz to 20 khz. The range of the human ear. Lovely. So I bought it.

When plugging it into my machine I saw that it's actually working out of the box. I didn't need their proprietary driver disk. It just worked. One other thing that I loved about this microphone is that it acts also like a sound card. Having an output for the headphones in the front. It also can boost the volume of anything without clipping it. Making a good headphone with a high volume sound amazing.

I could watch movies with the same volume as in the cinema by just turning a nob on the microphone. There is another nob. This one mixes between the output of the computer and what it hears with no delay. So I can hear myself while recording in real time. No latency.

This was so magical that I made a 3D model to tribute the Microphone. I gonna link it here:

lbry://@blenderdumbass:f/microphone-beyer-dynamic-fox-rigged:d

The problem

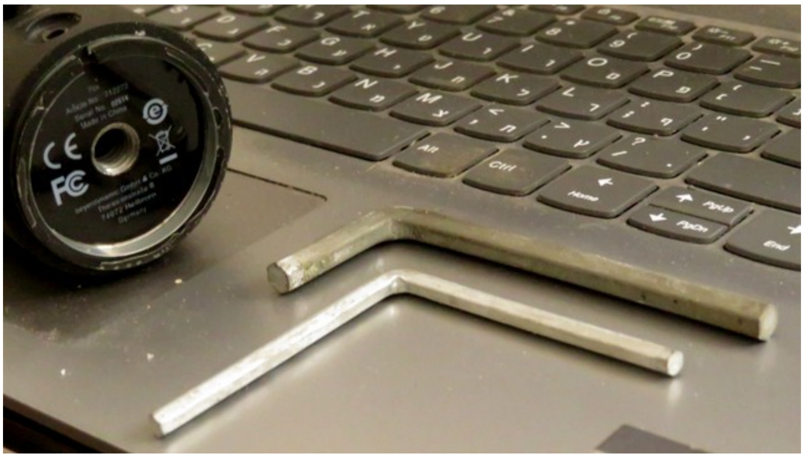
You can hear in some of my latest videos. Apart from maybe the one where I play JUMP Limited. That the sound is not in a good quality. It's because I broke the microphone. Technically it's not gone. And it's fixable. But let me explain.YouTube Just Got Even Worse | Creator Responsibility Initiative

One day a big object fell on it's cable and yanked it so hard that I had to replace the cable. But when I did replace it. The problems didn't stop. Ones in a while it would just refuse to work. Something was broken inside the microphone.

I am not the kind of dude that will just let something go because it's apparently broken. Especially when it's hardware that costs money. I will try to fix it. So that's what I did.

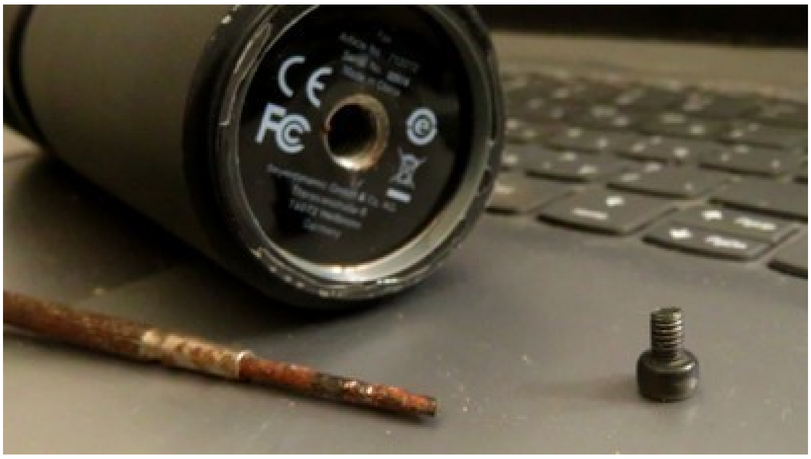
Opening it up

You need money to open the Microphone. Not to pay somebody. You need a coin to unscrew the first screw. Then there is this rubber part. I don't know why I removed it. It's not necessary to remove to get inside.

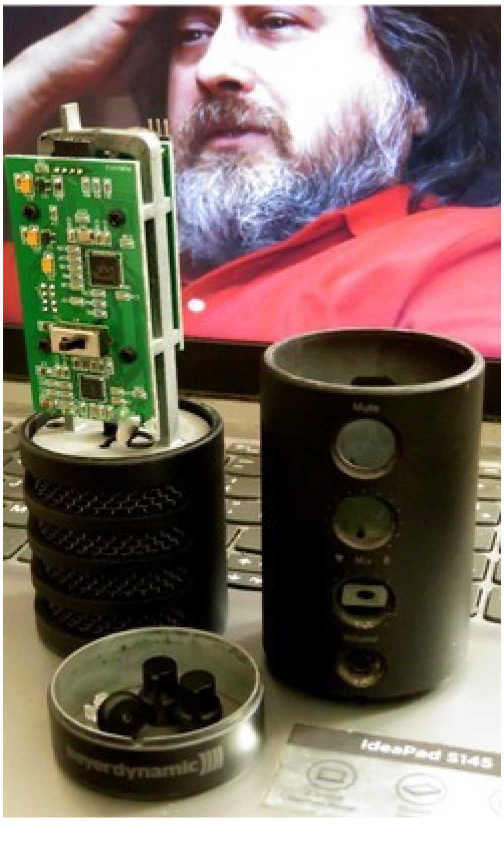


Under the big screw, in the hole you can see a smaller screw. But with a hexagonal shape.

I had those 2 things to try and unscrew that little one inside. But both of them were too large. And I wouldn't go and buy one just for one screw.

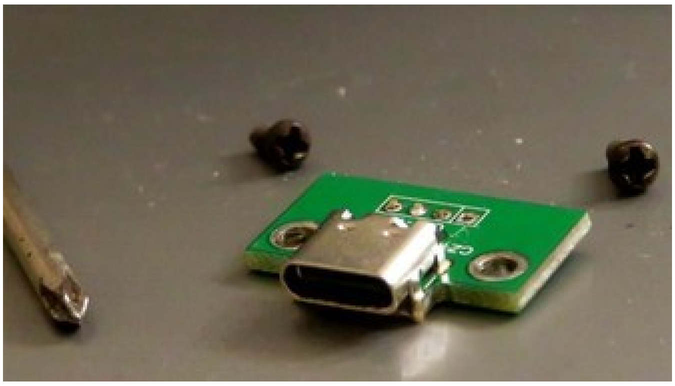


So instead I used this rusty screw-driver. And it wasn't the best experience. But I was able to take this screw out of there. Freeing the whole construction.

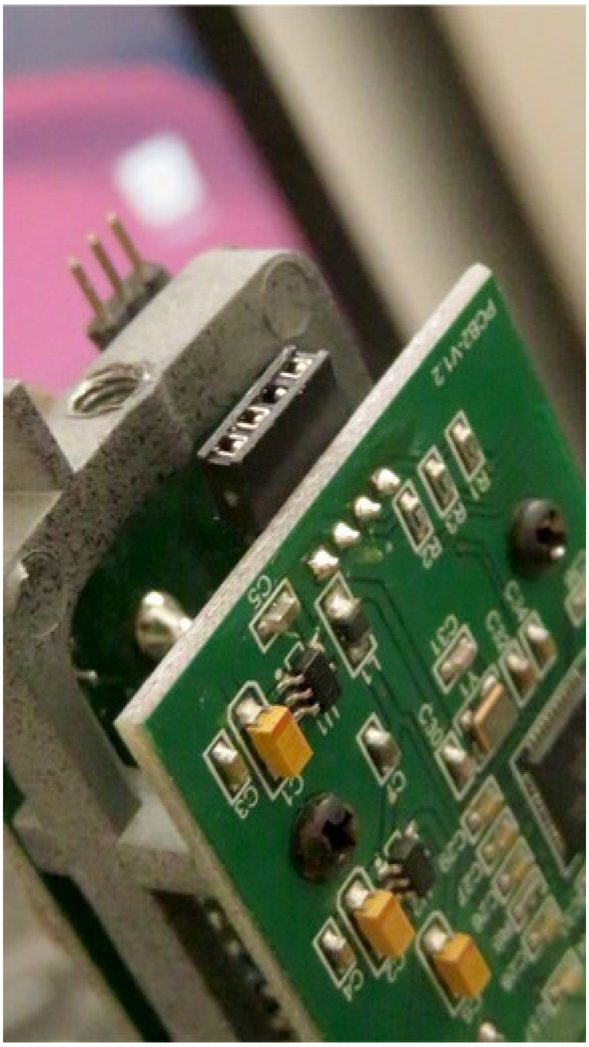


The next thing would be to remove these guys. It's pretty simple. You just pull them out. And as soon as you do that, you can open the case.

The top part with the logic board just comes out. You can see that the whole structure of the microphone and parts that hold all the important elements are made of metal. Making it extremely durable. Probably the cable yanked extremely hard.

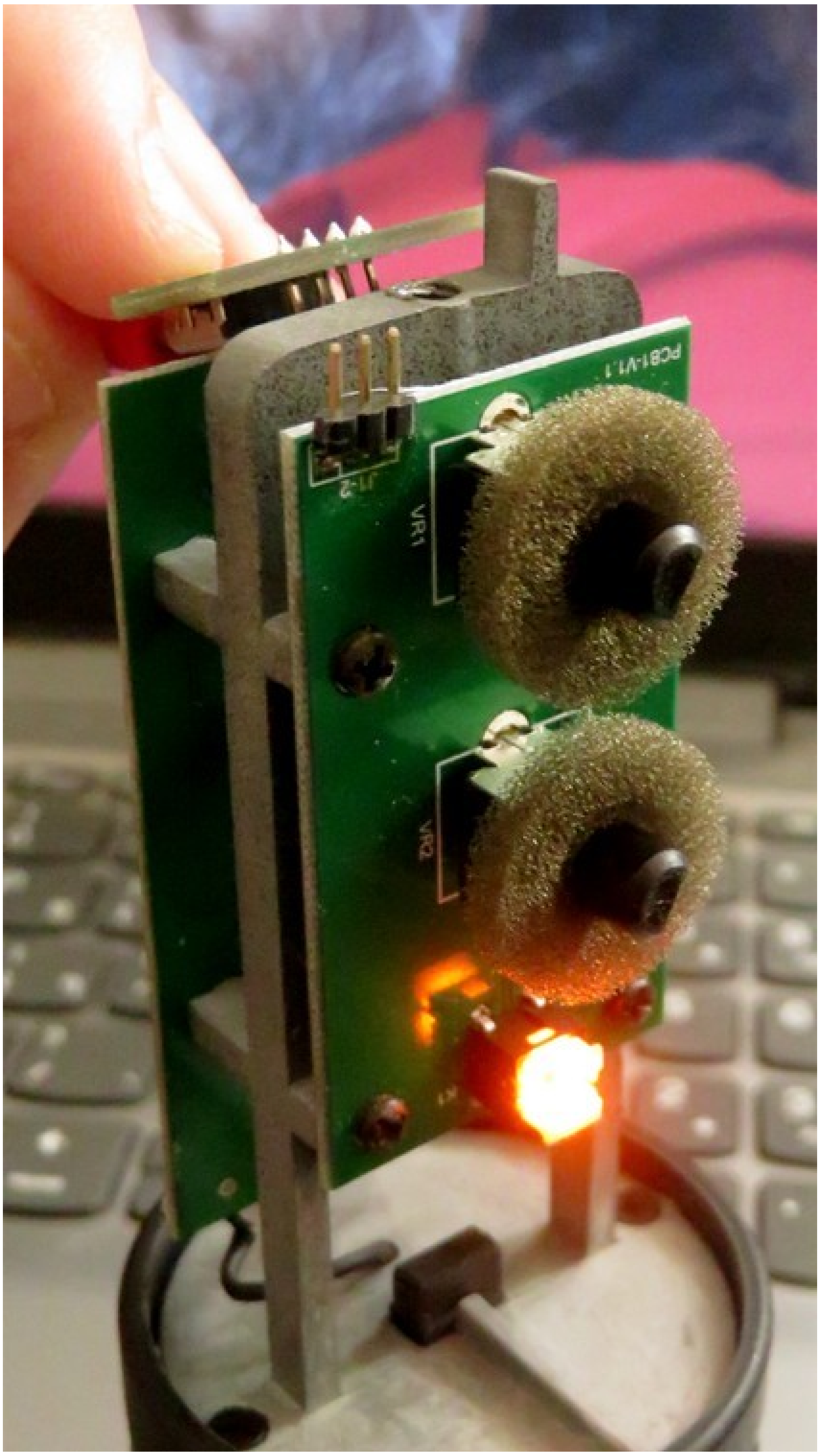


Inside of the bottom part you can see 2 little boards. They are the USB-C input and the headphones jack. While the headphone jack is fine. The USB board misses the connection part. It was yanked out of it.



On the board it self the connection part is loosely stuck while all the metal connections are torn out of their places from the USB board. One of them is missing.

If you put the USB part to the connection. Removing the black peace. The microphone is still functional. It's just I have torn the connections. And I need to glue them back together.



And for curious. This is microphone it self. It's holding on this rubbery, flexible material. Probably to lessen the vibration of the surrounding on the recording. Or something like this.

Plan to fix it

All I need a soldering station. And a one tiny peace of metal to put back there, where the part is missing. Maybe I can even use tiny cables instead of this black peace to connect the USB back to the microphone.

It's going to be better then buying a new one. Since if I have a soldering station I could fix many other things. Not just this one microphone. Still tho. I need to wait for about a month till I get my salary. I can't buy it without money.

Conclusion

Right to repair is an essential right. I would not buy a peace of hardware that is fundamentally against my ability to fix it. I would not use a peace of software that's against my ability to change it.

A lot of people when having a problem like this would just throw the microphone and get a new one. And then a new one. I'm not throwing it. I will fix it.

Happy Hacking!

YouTube Just Got Even Worse | Creator Responsibility Initiative

Sometimes something that sound like a good thing is not a good thing. Like proprietary software. The companies that make it, are experts in selling you, your own slavery. Let's take a look at one example.

`lbry://@blenderdumbass:f/YouTube-Just-Got-More-Ennoting:0`

We all know that the LBRY protocol's nature of Odysee is what's good and fair about this platform. Decentralization. Censorship resistance. Free Software. Freedom of Speech. YouTube on the other hand is an evil corporation that censors people, subjugates people, makes people addictive. All in the name of profit.

The Email

I have an email at Google still. I don't log in through their site. It would make me run their proprietary JavaScript. But I can use Free programs like Evolution or Thunderbird to view my occasional emails. I had a YouTube channel. And ones in a while I receive an email from YouTube talking about their new features and new policies.

Usually I don't look at those emails since it's YouTube and I don't care. But this time for some unfortunate luck I clicked on the email. Apart from their usual Ageism:

> **Supervised Experiences for Tweens & Teens** There might be a few more eyes on your videos–supervised experiences empower

> pre-teens+ to explore on YouTube. Learn what this new audience means for you.

What ever the hell this is. I don't want to know. Occasional ageism probably.

Also they sent this:

> **Your Influence Matters** Most creators are awesome, but some take things too far. Check out our Creator Responsibility Initiative.

https://invidious.snopyta.org/watch?v=PO5z1bqW5yY

I've changed the link to an INVIDIOUS instance so you wouldn't need to run YouTube's proprietary JavaScript.

Basically in the video he explains why people get demonetized for their views. And why Freedom of Speech is nothing for YouTube but "Creator Responsibility" is a thing now. Obviously is an attempt to sell their censorship practices. And fight against platforms like Odysee that have no problems like "demonetization" for "wrong behaviour".

Closer look

Let's look at the things they have to say and break down the arguments of YouTube.

> True or False? A creator can get demonetized or suspended from YouTube even though their videos didn't violate the community guidelines. What do you think? If you said true, your right. And although it's rare you may see it happen in recent years.

What we can understand from here. Is that they will try to frame this kind of behaviour from YouTube in a better light. So you would be less worried about it. And think less about things like Odysee. Let's see how it goes for goes for them.

> First, for some background, as a YouTube creator, you sometimes seen as a representative of the platform. And a member of this giant, influential community.

This is basically stating the centralized nature of the platform. Meaning they see it as a 1 house full of guests and friends. And they want to keep a good image of the house. While things like LBRY. There is no house. Everybody does their own stuff. No central authorities image is influenced by the people using the platform. Because there is no central authority.

> With great popularity comes great responsibility.

I have a whole article written about what I think of this one. But okay, we got the point. The more people listen to you. The more "power" about what they think you get. And it's true.

> If the things you say or do as such a representative are really reckless, dangerous, inappropriate, on video or not, you can actually cause damage to YouTube, and it's millions of creators by hurting their reputation and revenue. Why? Because YouTube and advertisers don't want to be associated with that level of craziness.

Good old "advertisers don't want you to say things that are uncomfortable" argument. Because they are advertising. They want their stuff to look nice and comfortable.

> Even if the creator's videos don't explicitly violate the community guidelines. There is a whole team of experts who weigh in on whether the action is taken on a channel.

Basically they have a bunch of people who check "creators" manually. And see if they making YouTube look bad. Even if it's not on YouTube.

YouTube's biggest Marketing Error

In my opinion they just shot them selves in the foot. Trying to calm down people who complain or leave YouTube. Trying to win Odysee. But they just made it worse for them selves.

All those things that he was talking about just proves one more time that the whole design of YouTube is flawed. Nobody can do nothing until a regime tell them that it's okay. Because they need everybody to be slick and beautiful.

YouTube sees it self as a store. Customers are those who want to sell advertisement. And the items on the shelves are the Videos and Personalities of "YouTube Creators". If the personality or videos are not going to look good for a potential buyer. They rather take this person off the shelf. Removing him from YouTube.

Those who want to simply express opinion. They can't have YouTube now. And a lot of channels. What they do is express opinion. Whether it's by talking to the camera or by making a Video Essay. Or by reacting to something. Most of the videos. Most of the popular videos are expressions of opinion.

And since expression of opinion hurts your chance to stay on the shelf. You gonna need to look else where. A different store. Where the buyer is the one looking for an opinion. Odysee. LBRY. Not YouTube.

Responsibility

Should publication Freedom exists? Should people be able to publish what ever the hell they want? No matter how unconformable and gross that it. I think Yes. But a system should be developed to accommodate this.

I talk about it in depth in my article Should "Content" Be Free. But the main idea that I explore is that if you have a sufficient enough warning system. Discretion disclaimers. Tag-like flag system for various categories of grossness. And stuff like that. So you would be able to choose for your self what stuff to avoid a what to allow. Then it's possible to have publication freedom without making it uncomfortable. And then publication freedom should exist.

Conclusion

YouTube maybe was a nice place to publish videos, some time ago. And it's still quite popular so I don't blame those who publish to it. But it's not there for long. And they know it. Their graphs going down. They panic. They try to rationalize their problems instead of fixing them.

Will this be the end of YouTube? Probably not. But I hope it would. LBRY has no ads. LBRY doesn't care what you post. LBRY doesn't care what you do outside of LBRY. Because LBRY doesn't exist. It's a concept, not a company. LBRY Inc. they are a company. Yes. But LBRY is an idea.

> Software embodies knowledge

This is quote from Richard Stallman. Source Code are ideas written in text in such a way that both humans and computers can understand. LBRY is free software. Those ideas passed on. And even if there will not be LBRY Inc. anymore. The ideas, the source code, are still there. Forever.

YouTube on the other hand is a sinking ship. That tries very hard to convince the passengers that they are not sinking.

Happy Hacking!

Trying to Beat H.P. Lovecraft

Sometimes when you write for long enough, you feel like a writer. This is what I was trying to check using this article.

lbry://@blenderdumbass:f/Attempt-At-Beating-Lovecraft:c

H. P. Lovecraft was an amazing writer. His language is sophisticated. I actually find it at times, confusing. I'm not an native English speaker. And I find his work a bit challenging. But even while it's hard for me to understand some stuff he wrote, his texts, I quite enjoy reading. I like the UFO type, cosmic horror genre. The one used spectacularly by pseudo documentary writers focusing on alien conspiracy theories. Whether they realize it or not. They borrow Lovecraftian influence.

His stories are usually small in size. Sometimes for about a half an hour read. I will try to condense his style into an article-long form. Making my own little story. Trying to borrow Lovecraftian style myself. I will not succeed. I can never be nearly as well-spoken as Howard Phillips. But I will try to put as much Love in my Craft as I possibly can, while attempting this ludicrous endeavour.

But before we are going to start this "essay" of mine. I gonna tell you about my friend Deborah. She started obsessing over Lovecraft in recent times. I still remember her being a clean and accurate young lady. Maybe with an edge of darkness so prominent these days among the kids. But her mature-like attention to her own look and smell was more then noticeable.

She was the most popular girl in town. We lived is Tzfat. It's on a northern part of Israel. The town is quite narrow in size. Some folk even were equating it resembling more of a village type than a town. Making a high chance of some quite spectacular individuals be known to almost, if not all, population of the town.

Deborah's dark edge was that she would date almost anybody. She would even date sometimes multiple people at ones. She considered her unhealthy relationships with men - interesting. It was giving her a blend of self confidence and worth. The kind she already deserved by her obsessive attention to her own self. But perhaps it wasn't enough. Her rational for this was that she was doing a research. Trying to find a perfect specimen for her future husband.

Deborah now a days doesn't remotely resemble the one Deborah Tzfat so whole-heartily admired. Her current state is more of a Hippie, Homeless person type. Hair looks like pre-scoured wool. Occasionally you can even meet there a bug or a spider. Some of them by now probably already have names. She doesn't paint her face anymore. She grew an ugly belly. Sometimes people mistake her for pregnant. She gained weigh so rapidly that her arms and legs still keep their normal shape. And if you saw her naked, which unfortunately I had no pleasure of doing. She has multiplicity of small, round wounds around her newly formed pillow. Perhaps the speed of the growth torn through the tissues.

People blame it on her obsession with cosmic horror stories. Saying that they tuned her into a monster her self. They draw parallels between her "research" of men. And that she found H.P. Lovecraft worthy of her majesty. But being a person I am. I couldn't simply leave it at that without talking to her. No matter how deranged a given person is. Sometimes communicating helps a great deal.

Her story is questionable at best. I would say delusional. Or even completely imaginary. But confirming with her physical state. She truly believes that it happened to her. Even though I cannot give any evidence apart from her words. Sitting in front of her and listening to it while she was looking dead into my eyes, was very intense.

Her story starts with a mysterious man she met one day on her "research" quest. She never seen him before. He was very quiet and well-behaved. This somebody, she thought, could be a perfect match for her then, near-royalty.

The man's name she can't remember. But his eyes. They never left her since. Black. Blacker than Black. No light escaped his eyes. It was nearly distorting reality. Black holes. Two, perfectly spherical, black holes in a scull of a man's head. On top the blackness there was a clear-coat of a very shiny, glossy material. Like if somebody took a charcoal and wrapped it into a transparent plastic. She says that his eyes didn't have a distinction of the pupil. It's like if the whole eye was one large pupil.

While he had gave her affection she desired. She doesn't remember his accent. It's if the man never spoke a single word to her. But she does remember understanding him very clearly. Every time he wanted to do something. Go somewhere. She would just simply know what to do. Or where to go.

The same night, slightly later. After a couple of drinks of Red Muscat wine she faded out. She describes the feeling in the following words "I couldn't hold myself up. I started falling. And then he just lightly, grabbed me. And I was up again. Slightly above the ground. Even tho I remember him standing in front of me.".

She claims that she flue up like if gravity suddenly disappeared. And the next thing she remembers was how she woke up. This is a part of the story where I doubted a lot of what she was saying. I mean, she admitted drinking that day. Maybe it had to do something with the alcohol. But I also drunk. And one day I drunk so much I faded out too. The stuff I remember before and after still make a lot of logical sense.

She woke up in a place full of guts. She could see a pulsating, live-like creature with veins and blood surrounding her. All the light she was getting was scattering through the flesh from somewhere outside. Was she eaten alive? That's what she was asking herself. It looked like a belly of a large beast.

One thing she can remember very vividly is that the whole thing was filled with meaty pieces. Making it unreasonably comfortable to lay there. Even tho she was laying on a pillow of organs from an unknown source.

A few moments later she realized that her body actually floats. She thought maybe she was submerged in some kind of liquid. But she could still breathe. And no matter how long would she wait. Her body couldn't find which where was down. Deborah would grab the guts and rotate. But no matter what orientation she was in. It would not change her perception of the float. The concept of "down" didn't apply there.

Of course her rational for all of it was that her boyfriend was an alien from outer space. And she was kidnapped. The guts room, she thinks, is an inside of a organically grown spaceship. And the lack of "down" was caused by lack of gravity in space. Whether her story is true or it all was an alcoholic hallucination, I'm yet not sure. She does remember some extremely vivid details. And it did effect her psychological state quite a bit.

I know I didn't succeed. Hopefully you are not waiting for an essay and you realized the joke of this article. There is no Deborah. Or at least I hope...

Happy Hacking!

Copyright Doesn't Make Sense

A lot of stuff in live makes perfect sense. Copyright isn’t one of them.

`lbry://@blenderdumbass:f/Copyright-Doesnt-Make-Sense:6`

John the New Sheriff

John is a man living lonely live on a farm. Most of his day he is working on his crops, caring for his sheep and cows. Milking cows. Fixing things around his house. John is a good man. And everybody in their town know that. Occasionally some gangsters pass by the town. John is usually brave enough to help fight them off. Helping the sheriff with his own riffle.

John never needed money to live. All his food he does him self. He already owns a house. It's quite comfortable. He sells a bit of his crop to get an occasional drink with his fellas. And to buy bullets to fight off the bad guys ones in while.

One day during a fight with another gang, Sheriff is shot. Before he dies he appoints John as the new sheriff. John sees it as a honour. And excepts the new title.

Suddenly people come to him. Miss MacDown's dogs ran away. Dr. Green's children had a fight. Somebody stole some chairs from lazy Mike. Police job started taking more of his time. And so he can no longer care for his crops. He can no longer Milk his cows and shave his sheep. So he hires a man. Pays him a good salary and that man cares for the farm for him. John owns the crops and the cows and he is the one eating them and selling them for money. But this time instead of buying drinks. He pays the employee to keep his farm.

John could do those things him self. But he needed to do something else. He is now a Sheriff. He has no time. It's more valuable for him and for the town that he will pay somebody else to watch over his farm.

A service that John is paying for, releases John from doing the job himself. He could've done it him self. But he just has no time.

Mike on the other side of the town also has a farm. Also has a worker. But he doesn't do anything with his time. He is just lazy. So he decided to pay for the same service John pays. It's perfectly fine. If you are willing to pay to have more free time. So be it.

Stolen chairs

Mike has 3 chairs in his house. One he uses all the time for him self. One used by his farm worker when he takes a rest. And another one for the guests. One day Mike comes to John. Because Mike counted having only 2 Chairs. One chair has disappeared.

John being a good Sheriff, find the chair in the house of Kyle. When he asks Kyle how the chair ended up in his house. Kyle says that he took the chair from Mike's house without asking Mike.

Kyle explains that he had no chairs in his house. And he is a poor man. His land doesn't grow any crops. He doesn't have cows to sell the milk. He sleeps on the ground. He just wanted to have a chair to sit on it. He saw that Mike always has one extra chair. And decided to take it.

When Kyle took Mike's chair. Mike had no chair to give to his guests. He had one less chair. He had 3 chairs before Kyle took 1. And now he has only 2. One chair changed hands.

Miss MacDown's tomatoes

On the other side of the town. Farther away was living Miss MacDown. She also had a little farm of her own. Growing pretty tomatoes. She would sell her own tomatoes. But occasionally would give one away if that person wanted to grow their own.

Kyle used to take tomatoes quite often. Miss MacDown knew that his soil doesn't grow anything. But she would give him a tomato. Or even two. Every time she had grown a couple extra. This is how Kyle lived. He would get a tomato from Miss MacDown. A cucumber from John. An onion from Mike. And so on until he can make a meal.

Vegetables grow and multiply. So if you give a tomato with good seeds to a person. He can make a tomato plant out of it. This is what Miss MacDown was doing. She would grow 4 tomatoes. One she would eat. One she would sell. One she would plant to get a new 4 tomatoes. And one she would give away gratis to somebody else. So he could make a tomato plant of his own.

Dr. Green's machine

Dr. Green lived in that town too. He was a scientist. He would sell various devices. Machines to do crops. He had a little factory full of workers. And he would pay them a good salary each. With his Free Time he would invent new things.

Ones he had made a new machine. This machine would solve the problem for Mike an Kyle. Giving Kyle a chair while not taking one from Mike. Using the ideas of Miss MacDown about her tomatoes. He came up with a way to copy objects.

He would borrow the chair from Mike for a short amount of time. A few minutes. During which he would put it into the machine. And the machine would grow many chairs like this. He would sell the new chairs. And sell the machine it self. Since if he built only 2 of them. He could copy the whole machine. And give each person living in the town a copy of the machine.

So there is no more need for stealing. You just see what item you would like to have. Ask for a copy. And you have one too.

Copyright

Let's talk finally about copyright law. And my views on it. Copyright should not exist. And you are probably now having so many things to say on how I'm wrong. And how you think it should exists. So be free to type in the comments. I will be more than happy to argue with you. I like it a lot.

How would artists make money?

If people share the works freely. Without any permission. How would any artist make any money? Since people could just get the stuff gratis from somewhere else. The irony here. Is that no matter how much copyright laws you put, or DRM you implement. It will never stop people who want to share from actually sharing.

I go to watch films in cinema despite having million torrent web sites. I gonna go to a concert despite having a gratis catalogue of songs online. I would press the support button on Odysee if I like what I watched or read. I would buy a CD. I would use my money. Even tho Internet exists. Because I just want to give my money to something that I find worthy of it.

If I see that the artist worth it. I will find a way to support him. Things like LibrePay, Patreon and the LBC coins made Copyright obsolete. It's better if a given publication is shared more. Because it will make more people aware of how to send money to a given author.

Also. Artist's work could be payed for directly. An artist can do a job making art as a service. Or a programmer could write Free Software for a person or a company. In exchange of money. Similar to John paying a farmer to farm for him. Because he couldn't do it himself. If you arrange it ahead of the time. It also could work.

But Piracy is Stealing

Equating copying information to attacking ships is at best unreasonable. When Kyle stole a chair from Mike. Mike didn't have a chair anymore. When you copy something you don't take it from the original owner. He still has his copy. But now you have your own too.

Mike had 3 chairs. As soon as Kyle took one. Mike ended up with just 2. One changed hands. If Kyle would copy one chair instead. Using the machine of Dr. Green. Mike would not even notice that something was changed. Because nothing was changed for him. Only Kyle got one more chair.

This is why equating copying to stealing is unreasonable and wrong.

Copyleft

The way to hack copyright out of existence is Copyleft. It's a technique of using the current copyright law and licensing in order to insure that everybody has a right to share the work.

The current copyright law of most countries prohibits anybody but the copyright holder to copy the work. Who is the copyright holder, can change. If the author wants to sell or give away his copyright to someone else he can totally do so. But they will not sell copyright for every person that will use that work.

To do distribution of such work. They usually give some people a license. The license is a document stating that some people can do some things with that work based on some conditions.

For example. I can say that you can use this text only for non-commercial sharing. But only if you will jump 3 times while making a share. If you shared it commercially. I can sue you. If you share it but don't jump 3 times. I can sue you.

Copyleft license is a license that will make sure that the work is permitted to be shared and modified. And if you do so. Your copy, or modified copy should also use a copyleft license. Making it a viral license. As soon as a work touches a copylefted work. It will become copylefted it self.

This text is under CC-BY-SA. A copyleft license. One more notable license is GNU GPL. This is a software license that uses a very strong form of Copyleft. You need to read the license. It's a very interesting read.

The hack of copyleft is that if copyright law becomes stronger. Copyleft becomes stronger. And more sharing will be done. But if copyright will be weaker. More sharing will be done anyway. So it wins either way.

Conclusion

Copyright is an interesting law. What do you do if you can make infinite number of something? But to be totally clear. Copyright is an artificial scarcity gimmick. Trying to add value to something that has no value. Because even if a given information is valuable to someone. Dividing it by infinity will equal to zero.

Copyright is there so only one entity could set a fixed prise for a given information. It's an old way of looking at things. And it doesn't even work.

Happy Hacking!

I Don't Have a Smart Phone

A short story about a man in 2021 that lives without owning a smart-phone device. This man is me, by the way.

`lbry://@blenderdumbass:f/I-Dont-Have-A-Phone:5`

General Purpose computers are an amazing invention. It's a device programmable to do any type of computation. And sometimes even more than that. You connect a computer to some other device. And now you can control that device with a computer. You connect two or more computers you can have a network. With a special program known to both sides of the connection you can communicate using them.

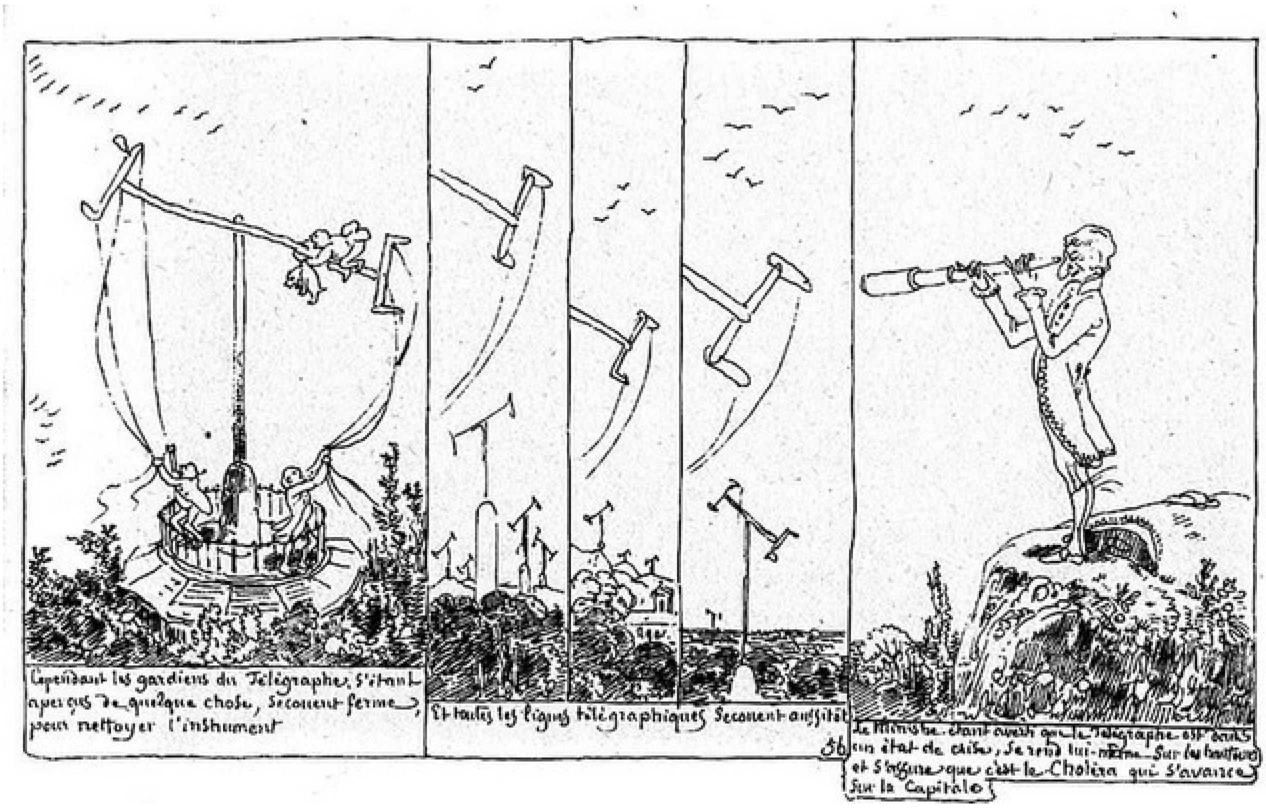
The internet. Communication between computers. Was a hack used by some hackers. They experimented on connecting a telephone line to a computer and sending some data over this cable. If you call to someone else that has done the same thing. And his computer is wired to receive the data. You successfully made two computers communicate.

If you write a standard of how communications like that are established. For example the "HTTP" standard. And make a system similar to phone numbers. The IP. Then you have Internet. You probably need a contact list. DNS (Domain Name System) is a good starting point. So instead of typing 104.21.22.220 every time. You can just memorise "Odysee.com". Tho there are proposals for a better system.

Connecting a computer with phone make a smart phone. Today we have small devices that we carry around everywhere we go. They are general purpose computers. And they are also phones. And they are typically connected to the internet.

Phone

A telephone. Or a telecommunication device is a device that lets two or more people communicate at a distance. For example Optical Telegraph could be called a telecommunication device. Since even tho it doesn't use electricity. Two people at distance could use it to communicate.


Cependant les gardiens du Télégraphe s'étant aperçus de quelque chose, secouent ferme pour nettoyer l'instrument
Et toutes les lignes télégraphiques secouent aussitôt
36
Le Ministre étant averti que le télégraphe est dans un état de crise, se rend lui-même sur les hauteurs et s'imagine que c'est le Choléra qui s'avance sur la Capitale

Optical Telegraph was a simple tower. One side was a construction that could change it's shape in come way. That a person with a telescope could see from far away. The shape would be changed. And the person on the other side would decode the message using some set of rules. Each country usually had their own codes. And their own shapes.

After the invention of electricity the world transitioned to the Electric Telegraph. It was a button and a receiver. Button would close the circuit and send a signal. And the receiver would make a noise each time the signal would be received.

To make it work there was a network of cables that was transferring those signals between telegraphs. The encoding this time was the Morse Code. Which was a way to standardise communication using a telegraph. Making sending and receiving messages easier.

Then people started experimenting with an idea to transmit more then one message in the same time using telegraph. Harmonic Telegraphy is when you transmit the signal in pulses of particular frequency. Instead of just sending electrical current. It generates a wave in some frequency. So the other end could tune into that frequency similarly to how you can tune into multiple radio stations. Allowing telegraph messages to be sent simultaneously on the same cable.

In 1857 Johann Philipp Reis made a device that could capture sound using a microphone and transmit this sound over the telegraph line to then be recreated on the other side using a speaker. This was the first time a telephone was made. People could communicate over distance using their voices. Tho this telephone was extremely complicated to use. It wasn't yet a great success.

Then a lot of people helped make this technology become a commercial thing that would be used my lots of people. And not just some lab nerds. By 1927 there were calls at distances between the US and UK. But it took 70 years of development to get to that point.

Mobile phones started their life in the 1960s. It was a huge device sitting in a car transmitting the signal over the radio instead of a electrical line. This machine looked like a regular phone. But could communicate while you move. Meaning it's a **mobile** phone.

Stalin's Dream

To make a modern phone call simple radio would not be enough. Your tiny, slim device would not be able to generate enough energy to transmit data far enough.

You probably seen those kinds of towers scattered all over the place. Those are little devices with antennas that receive and transmit calls. Your phone is connected to several of these right now. If you get far enough from an antenna like this. You will have "No Service". And your phone will nGeneral Purpose computers are an amazing invention. It's a device programmable to do any type of computation. And sometimes even more than that. You connect a computer to some other device. And now you can control that device with a computer. You connect two or more computers you can have a network. With a special program known to both sides of the connection you can communicate using them.

The internet. Communication between computers. Was a hack used by some hackers. They experimented on connecting a telephone line to a computer and sending some data over this cable. If you call to someone else that has done the same thing. And his computer is wired to receive the data. You successfully made two computers communicate.

If you write a standard of how communications like that are established. For example the "HTTP" standard. And make a system similar to phone numbers. The IP. Then you have Internet. You probably need a contact list. DNS (Domain Name System) is a good starting point. So instead of typing 104.21.22.220 every time. You can just memorise "Odysee.com". Tho there are proposals for a better system.

Connecting a computer with phone make a smart phone. Today we have small devices that we carry around everywhere we go. They are general purpose computers. And they are also phones. And they are typically connected to the internet.

Phone

A telephone. Or a telecommunication device is a device that lets two or more people communicate at a distance. For example Optical Telegraph could be called a telecommunication device. Since even tho it doesn't use electricity. Two people at distance could use it to communicate.

Optical Telegraph was a simple tower. One side was a construction that could change it's shape in come way. That a person with a telescope could see from far away. The shape would be changed. And the person on the other side would decode the message using some set of rules. Each country usually had their own codes. And their own shapes.

After the invention of electricity the world transitioned to the Electric Telegraph. It was a button and a receiver. Button would close the circuit and send a signal. And the receiver would make a noise each time the signal would be received.

To make it work there was a network of cables that was transferring those signals between telegraphs. The encoding this time was the Morse Code. Which was a way to standardise communication using a telegraph. Making sending and receiving messages easier.

Then people started experimenting with an idea to transmit more then one message in the same time using telegraph. Harmonic Telegraphy is when you transmit the signal in pulses of particular frequency. Instead of just sending electrical current. It generates a wave in some frequancy. So the other end could tune into that frequancy similarly to how you can tune into multiple radio stations. Allowing telegraph messages to be sent simultaneously on the same cable.

In 1857 Johann Philipp Reis made a device that could capture sound using a microphone and transmit this sound over the telegraph line to then be recreated on the other side using a speaker. This was the first time a telephone was made. People could communicate over distance using their voices. Tho this telephone was extremely complicated to use. It wasn't yet a great success.

Then a lot of people helped make this technology become a commercial thing that would be used my lots of people. And not just some lab nerds. By 1927 there were calls at distances between the US and UK. But it took 70 years of development to get to that point.

Mobile phones started their life in the 1960s. It was a huge device sitting in a car transmitting the signal over the radio instead of a electrical line. This machine looked like a regular phone. But could communicate while you move. Meaning it's a mobile phone.

Stalin's Dream

To make a modern phone call simple radio would not be enough. Your tiny, slim device would not be able to generate enough energy to transmit data far enough.

You probably seen those kinds of towers scattered all over the place. Those are little devices with antennas that receive and transmit calls. Your phone is connected to several of these right now. If you get far enough from an antenna like this. You will have "No Service". And your phone will not be able to call.

To make the cell phone network work. It needs to know which phone numbers are connected to which cell (tower) so the network will know which direction to send the call when it happens. The down side of such technology is that the cell network knows where you are all the time.

Even if you are connected at least to one cell. You are already clearly in the zone of this cell's service.Consent For Abuse But when you are connected to multiple of them. Simple math like triangulation can be used to determine your accurate location.

Making you walk around with a device that spies on you 24/7. All the time. Making it a Stalin's Dream device. If he would be alive today. He would make this technology be mandatory. And so everybody would have to have one. Today people who want to spy on you use different tactics. Like promotion. And slickness. And other various physiological means. Making you pay our own money to be tracked.

Worse then that

I do own a Nokia phone that I use occasionally. I didn't have a phone so my father bough me a very cheap one. So when I travel long distances I would have an emergency device. I guess it's fine. I don't take it with me. It's just there, in my house. Also I can totally use it as a flash light.

Smart phones on the other had I don't own. Maybe I will own. If I find one worthy of this. But currently if a device connects to a cellular network. It's not good enough. The network is what problematic here.

Smart phones are phones. But with added functionality of a general purpose computer. This is why they are so popular. Is like having a computer but small enough to put into your pocket. You can browse the web and do a lot of computation on it. It usually comes with a lot of install-able software (apps). A general purpose computer.

The problem with most phones today apart from some noteworthy exceptions, that they all use proprietary software through out the phone. Putting it together with a device who's location is always known. And that has at least a microphone. And in today's world also a camera. Or even multiple cameras. If your software is secret. And it's function is unknowable. Trusting that this device is not a spy in your pocket is unreasonable at best.

And even worse than that. Additional software (apps) that comes as an install-able option. Are developed by people who are usually want to collect as much data about you as possible. And to sell it. You are basically buying the worst kind of device ever.

How do I live without a Smartphone?

A lot of people ask me this question when I tell them that I don't carry a phone with me. And their arguments are usually come to the same few things. The time. The camera. The music. The communication. The video / film. And other types of computation. How am I getting those. If I don't have a phone?

First of all. All of these things are optional in life. But I still have them. Even tho I don't have a phone. The point is. You don't need the particular something. You may need it's function. The idea I described in my video:

https://odysee.com/@blenderdumbass:f/apps-you-dont-need:e

The time. I have a wonderful mechanical watch. It works just fine. And I don't need to charge it every night. I do have to change the battery ones every couple of years. The same duration of time in which people decide to buy a new phone.

The camera. I own a Cannon SX720 HS. With 40X Optical Zoom. Find that on a phone. And it costs nothing compared to a smart phone. I would buy a more expensive camera. But only if I found a use for it. For now I'm happy with what I have.

The music. I recently bought a Music player. It's a tiny device. Plays all my favorite audio files. And costed me about the same money I eat every day. So basically nothing.

The communication. I have a standard phone plugged into a wall. It was so cheap that a 6 pack of Pepsi costs more.

The video / film. I still have a laptop. So I can watch video and film here. First of all watching a movie on a phone is not cool to begin with. Films experienced best in the cinema. And before the virus I would go there quite often. I might buy a projector to increase the screen a bit more. But for a casual movie, laptop is fine.

Computation. When it comes to software. A computer. A laptop in this case. Or a big stationary computer. Is way better than any phone. Because it's what computers originally were designed for. To make people's computations. For when I do work. And I need to solve a math problem quickly. A calculator is enough. And they are quite cheap.

How does it feel?

Compared to having a phone. Not having a phone feels amazing. I don't check messages every second. When I have a minute of Free Time, I will not go into an addictive app. I will think. How many people have time to simply think today?

I read more and I write more.

Lowering the dose of Dopamine made me increase the Serotonin. Swapping constant excitement with a nice feeling of happiness.

I feel the freedom. When I go to work. I feel light and unrestricted. I feel like I'm here for myself with myself. And there is nobody that will interrupt me.

People call me less. They prefer to send me an email instead. And I have a time to read it. Think about it. Then reply. I don't get called for work suddenly. The boss has no choice. To be honest I didn't tell him about the stationary phone in my house. And I guess it was a great move.

I remember my old phone with constant messages and notifications and what not. I see how much people stare at the screen and think "This is nightmare".

Conclusion

If you don't want to be tracked, spied on, and available to the entire world at all times. If you want to have free time to think. If you want to feel happy. If you want to be free.

You can do that. Take your phone. And get rid of it.

Happy Hacking!

I want you to notice. That my articles are released under a Free License. Usually CC-BY-SA. You can see the license in the description on LBRY application (or LBRY . TV). Odysee for some reason doesn't have this feature. This article is under CC-BY-SA.

Thumbnail was based on this image (CC-BY)

This image was originally posted to Flickr by Senado Federal at https://flickr.com/photos/49143546@N06/22622160063. It was reviewed on 5 December 2015 by FlickreviewR and was confirmed to be licensed under the terms of the cc-by-2.0. ot be able to call.

To make the cell phone network work. It needs to know which phone numbers are connected to which cell (tower) so the network will know which direction to send the call when it happens. The down side of such technology is that the cell network knows where you are all the time.

Even if you are connected at least to one cell. You are already clearly in the zone of this cell's service. But when you are connected to multiple of them. Simple math like triangulation can be used to determine your accurate location.

Making you walk around with a device that spies on you 24/7. All the time. Making it a Stalin's Dream device. If he would be alive today. He would make this technology be mandatory. And so everybody would have to have one. Today people who want to spy on you use different tactics. Like promotion. And slickness. And other various physiological means. Making you pay our own money to be tracked.

Worse than that

I do own a Nokia phone that I use occasionally. I didn't have a phone so my father bough me a very cheap one. So when I travel long distances I would have an emergency device. I guess it's fine. I don't take it with me. It's just there, in my house. Also I can totally use it as a flash light.

Smart phones on the other had I don't own. Maybe I will own. If I find one worthy of this. But currently if a device connects to a cellular network. It's not good enough. The network is what problematic here.

Smart phones are phones. But with added functionality of a general purpose computer. This is why they are so popular. Is like having a computer but small enough to put into your pocket. You can browse the web and do a lot of computation on it. It usually comes with a lot of install-able software (apps). A general purpose computer.

The problem with most phones today apart from some noteworthy exceptions, that they all use proprietary software through out the phone. Putting it together with a device who's location is always known. And that has at least a microphone. And in today's world also a camera. Or even multiple cameras. If your software is secret. And it's function is unknowable. Trusting that this device is not a spy in your pocket is unreasonable at best.

And even worse than that. Additional software (apps) that comes as an install-able option. Are developed by people who are usually want to collect as much data about you as possible. And to sell it. You are basically buying the worst kind of device ever.

How do I live without a Smartphone?

A lot of people ask me this question when I tell them that I don't carry a phone with me. And their arguments are usually come to the same few things. The time. The camera. The music. The communication. The video / film. And other types of computation. How am I getting those. If I don't have a phone?

First of all. All of these things are optional in life. But I still have them. Even tho I don't have a phone. The point is. You don't need the particular something. You may need it's function. The idea I described in my video:

https://odysee.com/@blenderdumbass:f/apps-you-dont-need:e

- **The time**. I have a wonderful mechanical watch. It works just fine. And I don't need to charge it every night. I do have to change the battery ones every couple of years. The same duration of time in which people decide to buy a new phone.
- **The camera**. I own a Cannon SX720 HS. With 40X Optical Zoom. Find that on a phone. And it costs nothing compared to a smart phone. I would buy a more expensive camera. But only if I found a use for it. For now I'm happy with what I have.
- **The music**. I recently bought a Music player. It's a tiny device. Plays all my favorite audio files. And costed me about the same money I eat every day. So basically nothing.
- **The communication**. I have a standard phone plugged into a wall. It was so cheap that a 6 pack of Pepsi costs more.
- **The video / film**. I still have a laptop. So I can watch video and film here. First of all watching a

movie on a phone is not cool to begin with. Films experienced best in the cinema. And before the virus I would go there quite often. I might buy a projector to increase the screen a bit more. But for a casual movie, laptop is fine.

- **Computation**. When it comes to software. A computer. A laptop in this case. Or a big stationary computer. Is way better than any phone. Because it's what computers originally were designed for. To make people's computations. For when I do work. And I need to solve a math problem quickly. A calculator is enough. And they are quite cheap.

How does it feel?

Compared to having a phone. Not having a phone feels amazing. I don't check messages every second. When I have a minute of Free Time, I will not go into an addictive app. I will think. How many people have time to simply think today?

I read more and I write more.

Lowering the dose of Dopamine made me increase the Serotonin. Swapping constant excitement with a nice feeling of happiness.

I feel the freedom. When I go to work. I feel light and unrestricted. I feel like I'm here for myself with myself. And there is nobody that will interrupt me.

People call me less. They prefer to send me an email instead. And I have a time to read it. Think about it. Then reply. I don't get called for work suddenly. The boss has no choice. To be honest I didn't tell him about the stationary phone in my house. And I guess it was a great move.

I remember my old phone with constant messages and notifications and what not. I see how much people stare at the screen and think "This is nightmare".

Conclusion

If you don't want to be tracked, spied on, and available to the entire world at all times. If you want to have free time to think. If you want to feel happy. If you want to be free.

You can do that. Take your phone. And get rid of it.

Happy Hacking!

Consent For Abuse

How many times have you signed a documents or clicked that “Agree” button, without actually reading the text of what you are agreeing to?

lbry://@blenderdumbass:f/Conset_For_Abuse:b

Yesterday under a video by @OfficialZaney I commented that, quote:

> I think I might write a new article about it.

The "it" I was referring to was Privacy. The video he made was mocking the bad decisions Microsoft made insuring that nobody can ever trust their operating system Windows to respect their privacy.

I would go even deeper than that. **For sensitive people, the following text of this article will touch upon subjects like rape, death, pedophilia and cannibalism. Please. Do not read this if you think you can't handle the text.**

Consent

Consent is an agreement. It's when one person voluntarily agrees to the proposal or desires of another. It's when one sets the rules and the other follows them. Agrees to them. Consent is a tricky thing tho. How can you be sure that a person actually consents to something? How can you know that you **didn't** consent to something?

Armin Meiwes

Armin Meiwes is a German computer gig. Or he was. Now he lives in prison. His sentence will end after his death. Even tho he believes he did nothing wrong.

In 2001 Armin wrote a little ad campaign to his web site. Looking for a specific man. "...looking for a normally-built 18-to 25-year-old...". In March a man named Bernd Jürgen Armando Brandes answered to the ad. He was another gig. An engineer from Berlin.

The other part of his ad was stating the following "... to be slaughtered and then consumed". Meaning that the ad called for a person that would Consent to be eaten by a Cannibal. And Bernd was that person.

They came together. And to prove that everything was consensual Armin videotaped the entire act. There was a very long story of how they were trying to eat Bernd's penis. If you want details, read about it on Wikipedia. You will get plenty of crazy stuff like "Brandes may already have been too weakened from blood loss to eat any of his penis".

Then Maiwes ate Bernd slowly for the next 10 months. Making an act of cannibalism. But Consensual. Or is it?

There was a lot of discussion from this case about whether self harm like that could be consensual in the first place. Film were made about Armin and Bernd. The case inspired music of Rammstein, Marilyn Manson and Ozzy Osbourne.

This is an interesting question. Can a person consent to be murdered? Can a person consent to be eaten?

Naomi and Dave

When I met my Girlfriend. It was so intense, the idea of us even being together. You know she is underage. We didn't do anything illegal. And we are not planning to. I already even been to prison for 3 days because of her. Just because the police thought that they caught a person that did something wrong.

They kept me in a prison-like facility for 3 days. Trying to find something that proves that I was guilty. But they couldn't.

When it comes to authors of fiction as soon as the idea visits their mind. It doesn't really escape it. It turns into a more extreme version of the same idea. So in order to explore what could've happened if we did something illegal. And the consequences of that. I made a little story about Little Naomi and Big Dave.

I tried to imagine a way with which I would still consider what they did was somewhat "okay". But still illegal enough that will make Dave never escape the consequences of what he did. If you want to read the story. It's not made in a written form. It's not a movie either.

The story about Little Naomi and Big Dave is made in a form of a music album. And here it is (OGG, CC-BY-SA).

lbry://@J.Y.AmihudMusic:c/J.Y.Amihud---Little-Naomi-Full-Album-2020-(-OGG-):9

Would you consider what they did was consensual? Or it more resembling the first story of a cannibal? Are both consensual? Or people can't consent to hurt them selves?

EULA (End User License Agreement)

End User License Agreements so prevalent in the world of tech, are used in order to get legal consent from users. Usually for malicious dis-services. There is a difference between a license and an EULA. The difference is that with EULA there are things that could harm you that extend the applicable law.

For example copyright restricts some usage of a publication. But for software for example. Things like Reverse Engineering is totally legal. But many proprietary software use EULAs to prohibit you from doing it. When you run the software it asks you to agree to some terms. And you click "Agree". You can't reverse engineer it now. Or copy it. Or do other things they would not want you to do. If you clicked "Agree" and it has those additional restrictions in the EULA's text.

A lot of people complaint about Fair use in videos. Stating that it's not fair to demonetize the video if a little portion of a song appeared in it. And show the fair use laws. But they don't realize that the samples were taken from a song usually covered by some kind of EULA like this. That prevents you from exercising the fair use.

For example in Spotify's EULA it says quote:

> You agree not to, and you will not permit others to: license, sell, rent, lease, assign, distribute, transmit, host, outsource, disclose or otherwise commercially exploit the Service or make the Service available to any third party.
>
> Any further restrictions or specifications will be according to the Terms & Conditions of Spotify.ac, available at: https://spotify.ac/terms

If you click on the link it says also:

> You are specifically restricted from all of the following:
>
> 1. publishing any Website material in any other media;
> 2. selling, sublicensing and/or otherwise commercializing any Website material unless otherwise authorized;

3. publicly performing and/or showing any Website material;
4. using this Website in any way that is or may be damaging to this Website;
5. using this Website in any way that impacts user access to this Website;
6. using this Website contrary to applicable laws and regulations, or in any way may cause harm to the Website, or to any person or business entity;
7. engaging in any data mining, data harvesting, data extracting or any other similar activity in relation to this Website;
8. using this Website to engage in any advertising or marketing.

Certain areas of this Website are restricted from being access by you and Spotify.ac may further restrict access by you to any areas of this Website, at any time, in absolute discretion. Any user ID and password you may have for this Website are confidential and you must maintain confidentiality as well.

Basically saying that they can do what ever they want to you. And you can't do anything to them.

Agreeing to this EULA and then claiming that you have Fair Use is ridiculous.

But now you will tell me "I never read those. I just click Agree."

Unknown consent vs Known consent

Most people who agree on the terms of services do not read what they agree on. This is why they get problems. Because they assume that the services terms are common sense. And that nothing necessarily terrible is written there.

What you have to understand is that those EULAs are designed for one purpose only. Is to gain power over you. Is to make sure that no matter how terrible, mistaken, problematic or otherwise abusive the system might be. By design or by mistake. That they will never be accountable for that.

They could sell your data. Restrict your rights. Delete files you payed for remotely because they don't like you having it. And you can't say anything. Because you clicked that damn "Agree" button. And in the text to what you agree, it says that they can do all of those things to you. But you just never read it.

A similar system of liability exists in Free Software licenses too. For example this is the passage from the GNU GPL:

> THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

Funny thing is. Even tho Terms of Spotify are not the same thing as the GNU GPL license (that is designed to give you Freedom). Spotify also uses similar tactic:

> This Website is provided "as is," with all faults, and Spotify.ac express no misrepresentations of any kind related to this Website or the materials contained on this Website. Also, nothing contained on this Website shall be interpreted as advising you.

While in GNU GPL it was necessary to protect those who distribute a modified copy of the software. Because modified copies not always work as intended. This NO WARRANTY statement is totally necessary.

For Spotify tho. This is another way of saying "Screw you, we do what we want."

Also adding to it. The EULAs usually are developed to be so boring to read. So only few people would ever attempt to read those. This is defective by design.

Conclusion

You may argue with me on the following statement I make. I think that a person should be able to consent to what ever they want. But it's not a consent when it's something they don't know they consenting to. For example if I didn't read the EULA but clicked. I don't value it as consent. But when It's something terrible. But I totally know what it is. It's still consent.

The law would disagree with me. EULAs are still legal. But Armin and Big Dave are still in prison. I wouldn't necessarily protect them. Or stand for them. But it's their business. Not mine. And if a person want to be eaten. All I can do is to try to convince him to want otherwise. But not gonna force per to change per's mind.

As it comes to your personal freedom. EULAs are agreements. If you check the text of any contract you sign. Please read what you agree for in the digital space too. If you don't agree. Delete that program and install a Free Software program that does the same job.

Happy Hacking!

Privacy is Important !!!

People say stupid things like:

“I have nothing to hide”

Okay, then go an broadcast all of your life, including the toilet in a 24/7 live-stream. Where anyone can join and watch.

`lbry://@blenderdumbass:f/privacy-is-important:7`

In 1981 band named Oingo Boingo released their first real album Only a Lad. You can get a DRM Free copy of this album from Archive.org.

https://archive.org/details/05.YouReallyGotMe

Allegedly it's under Creative Commons, Attribution, Non-Commercial, No derivatives license but I could be mistaken. The DRM free publication I found, claims this.

In their song "Nasty Habits" they go over an interesting reality of life. People do things for their own pleasure sometimes that they are ashamed of publicizing. Things like a particular preference in women. Unhealthy interest in English puppies. And other things that are harmless but not could lead to uncomfortable situations.

In the same album. The first song "Little Girls" is talking about something a bit more uncomfortable. About an unhealthy type of relationship. Advocating that loving little girls is okay since they are not complaining as much as adult girls.

From my personal experience I know it not to be true. Underage girls complaining the same amount as adult girls would. Because if you really look at both "types" of girls. They are all just humans.

But it's besides the point. If somebody knows that you are into "leather quins", it can lead you to uncomfortable situations. The same as if someone knows that your girlfriend is under 18 years old. It leads to even more uncomfortable situations. And it's in your best interest to avoid people from knowing both things.

When I was arrested, the police told me to lie about what I was arrested for to the cell mates. Saying that it would hurt me a great deal. Similarly of how me writing this could hurt me. I was released 3 days later because police and the court realized that I did nothing illegal. And even tho I talked to the cell mates about my girlfriend and her age. Nobody attacked me.

But it's possible only if you control how the information is being revealed. If I would start this article, or talking to the cell mates, from the words "I date a little girl". It would have a much different tone. This is why you should always control what information about you and in what order gets presented to the public, if any.

If you read my article linked above as "personal experience" you know that I met her on the street by chance. And we grew into this kind of strong bond. Like if we were brother and sister. But since we are not. We are planning on getting a family when she grows up a bit.

This is different from somebody who is looking for active sex with children just because they are "easier to deal with". Me and Rita, my girlfriend, invented a system of dividing pedophiles into 2 categories. "Милашки" and "Ебланы". We speak Russian when we talk. It's "Cuties" and "Fuckers" when it's translated to English.

- **Cuties** are those who want to get a relationship. Who care. Generally nice people. Similar to Me or Dave from my album Little Naomi.
- **Fuckers** will be anyone who does this kind of things for their own pleasure only. So your typical pedophile.

Cuties need privacy

Imagine that you are a **Cutie**. Doesn't matter how much you did. Or how much you try to avoid thinking about it. If this information becomes available to the wrong people. And available in such a way that could be taken out of context. You will be framed as a **Fucker**.

This could lead to very dangerous consequences. People might stop talking to you. You may loose a job. You may get the same type of police investigation that I had. You may get bitten up or ever killed.

And all of this because people know who you are. And just interpret it as if you are a **Fucker** because they see no difference between **Cutie** and a **Fucker**.

Changes in rules

Let's say you are not a **Cutie** and just a regular guy doing his regular stuff. You do only what's allowed. Both legally and generally. (I'm not considering my relationship with Rita as immoral. So I gonna use "general" to describe rules that people use, that are not part of the law.)

A lot of people are fired from their jobs and persecuted for tweets they wrote long time ago. Because back then when they wrote it, it was considered generally okay. But now the general rules have changed.

Ones I told about it to a person who doesn't use a smartphone. Because it's not allowed in his family. He is a religious, Jewish man. He carries around a simple surveillance device. He couldn't believe me that a man would be persecuted for something that was allowed when he did that.

> Wouldn't they forget all the previous instances? Because it was allowed before hand.

He thought that people will just see when a given tweet was made and judge whether it is fine based also on the time of it's publication. Which is not how people judge. If you are changing the rules to include that something else is not allowed now. It means you see it as bad regardless on when it happens. And probably try to persecute as many people who did it until now as possible.

This is why nobody should have a detailed record about you, period. It might be just a bit too dangerous. Because rules change, even if you have nothing to hide now. It doesn't mean that you have nothing to hide from tomorrow.

Rogue Employees, Security Breaches

A lot of companies who's whole business model is based on collecting your personal information understood my sentiment about changing rules. And understood that people just simply want privacy. That's why they talk about how it's important to them. And promise that no data will ever be published.

This is probably comes from big companies sense of Power. They believe that they are the most secure and all the employees are just the best people ever.

I would recommend a not so well known James Cameron movie called True Lies. It's about a secret agent that finds out that his wife cheats on him with some guy. And uses his job as a secret agent to track them, spy on them and do other nasty things to them.

This is something that was not allowed to do. He is an agent and should serve his country being one. He knows that. And all the intelligence he gathers about people. Or tracking he does. Should serve only the country. And he knows it. But he decides that his personal matters worth doing something that's not allowed.

Image for a second that you are a wife of a Google employee. And you met a guy. You are not cheating or anything. But you just talked a bit too much. And to escape nasty fights at home. You are not going to tell him that. But he knows anyway. He was reading the transcript captured from your mobile device. The company didn't allow him to do so. So what. He did it anyway.

On the internet you can find lists of people who are considered to be "sex offenders". It includes both **Fuckers** and **Cuties**. You can probably find a bunch of shit written about me too. And even, probably, about this post of mine. I took the risk of being persecuted for spreading my personal information.

How do they have all this data? It's because people who are in this data didn't care enough about their privacy. They used Googles and Facebooks. And sent, so called "private" messages to their friend about things they do or want to do. The company promised that the data will never be publicly available. But it was. It was either taken by rouge employees. Or taken via a security breach.

To make a privacy respecting service. You may not collect any more data than absolutely necessary. And when you are done with the data you did collect. It must be destroyed. So no security breach, or rouge employee could use this data in any way possible.

Anonymous Data

Many companies claim that they use or sell only anonymized data about you. Meaning removing anything from the data that could point directly toward you. Face, Name and stuff like that. But is it enough?

Most often then not people who buy the data from the companies, so called Data Brokers, are collecting data from various services at ones. They combine the data and sell them further. Until some company or service buys the whole package and uses it for their analytics. Or what ever they want to use it for.

Being that data is collected from multiple sources. Some anonymized and some not. Combining the data and drawing a picture about **you** is not too hard. Especially using algorithms. And if you know where to look.

If you are using a phone with a keyboard that has a Google help bar in it. A thing above the keyboard that suggests you the next word to write. It's probably knows your writing style very well. Because the more you type on it. The more Google algorithm knows what would be **your** next word. Making it be able to recognize your text.

This has a huge privacy implications. Meaning that Google knows if you wrote an article even if you posted it anonymously. If anonymized data like this could be published about you. And some algorithm somewhere can figure out it is you. Whether it's google's text, Facebook's face recognition. Or some other type of software. Anonymized data is not good enough. No data is the only way.

Conclusion

Privacy is important. But it's also hard. Because true privacy is nearly impossible. But essential. The only thing a person can do is to try their best. Ditching services that abuse you. Not revealing everything about yourself right away. Taking meaningful decisions. Using Free Software that at least has a chance to be private. There is no security in Proprietary software against it's Proprietor. And again other's. You can't really know.

A healthy dose of Paranoia could be helpful to all of us.

Happy Hacking!

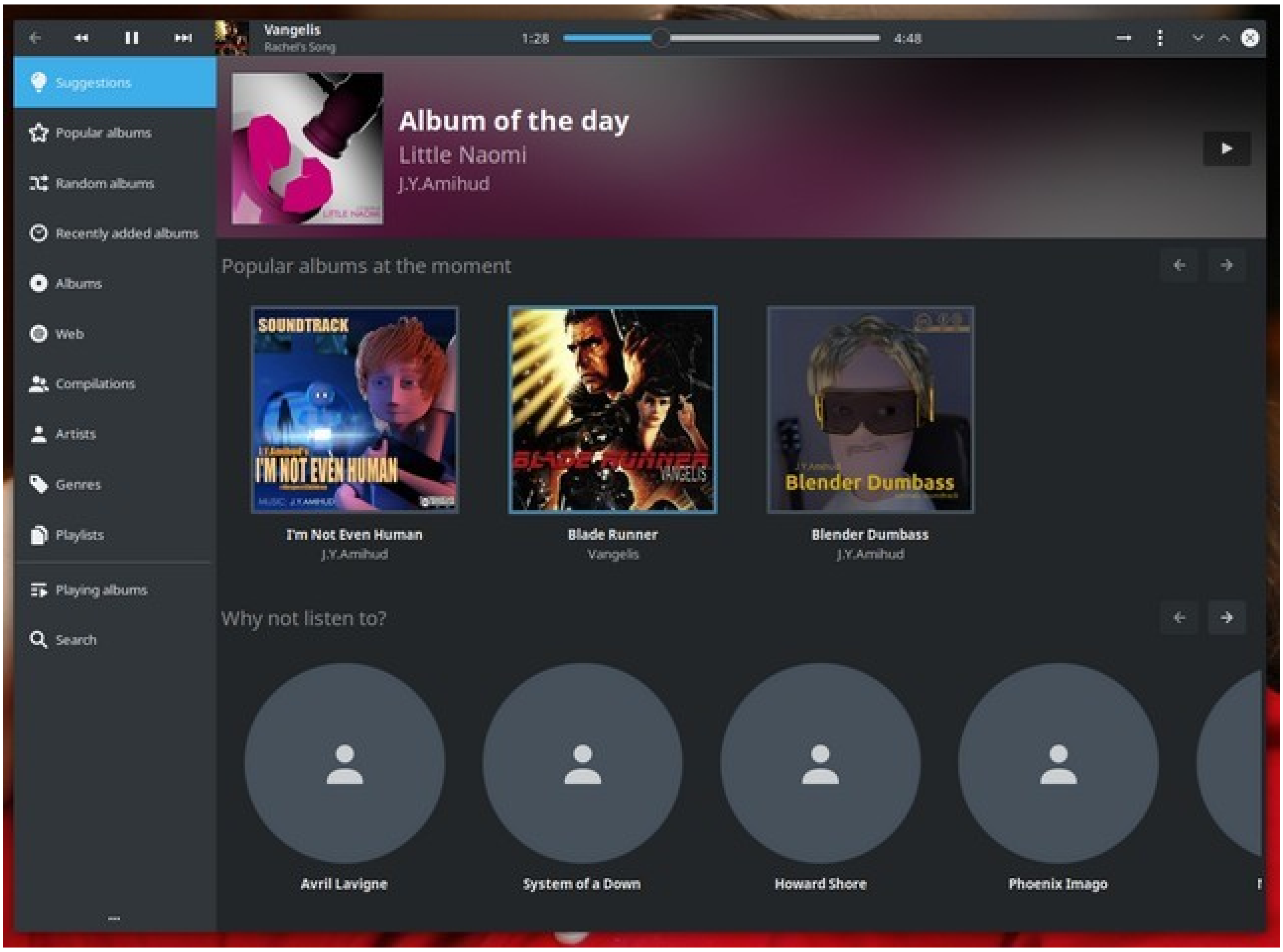
Lollypop - Hell of a Good Player

Sometimes a piece of Free Software is worthy of talking about. Like for example with this, rather amazing music player.

lbry://@blenderdumbass:f/Lollypop-Hell-of-a-Good-Player:a

I just found a MicroSD drive where most of my old music was stored. Huge collection of stuff. More then a thousand songs. All carefully put together by me long time ago. It's all full album collections. So there are no random songs. And players like VLC do not give such a collection a proper treatment.

I love VLC I use it quite often. But I think one of the best Free Software players out there today is LollyPop.



While taking this screen shot of the program I was surprised on how much it loves to promote me. Little Naomi by J.Y.Amihud. My album. Of course I have a very well put together package of my album on my drive.

Blade Runner soundtrack by Vangelis is very good. It gives walking on the street at night a crazy, melancholic atmosphere. Like if you suddenly in Los Angeles of 2019 fighting humanoid, biological robots replicants. And feeling all down while doing so.

What I like about LollyPop are 2 things.

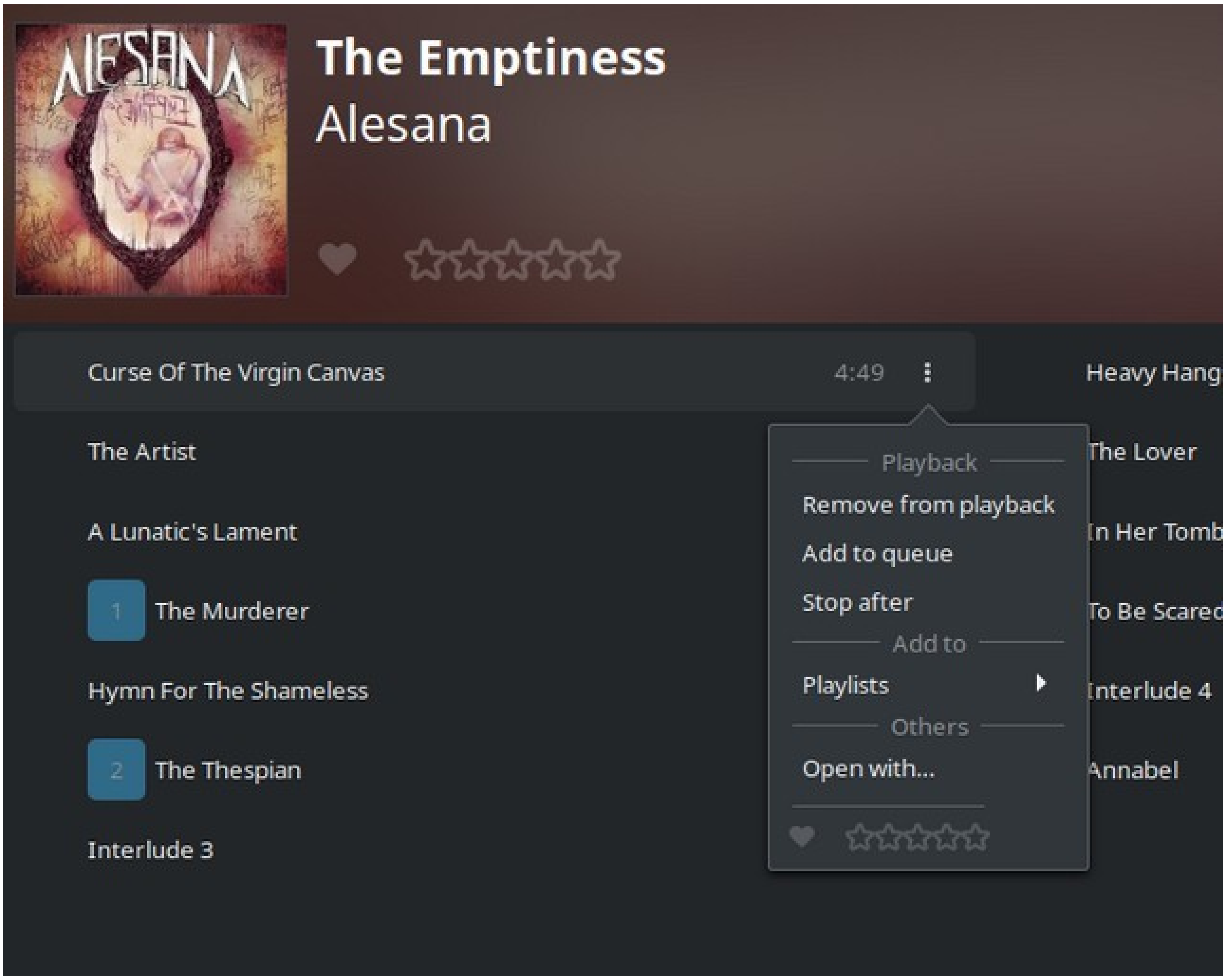
1. It's Free Software. And here is the source code. It's written in python. Wait. Think about this again. LollyPop is written in freaking python.
2. LollyPop is an amazingly polished experience.

Slick and pretty. Native GTK that made by the Gnome people to promote the look of Gnome. It's probably the number one application they want people to see when promoting the Gnome technology.

It's convergent. Meaning on Librem 5 or Pinephone it will fit to the little, vertical screen and act naturally.

You can even play with it on a regular desktop. Just shrink the window and you will get the LollyPop mobile.

It has a wonderfully designed full screen mode. With a spinning album cover. Like if it was a vinyl disk.



It also has a system that I saw in VLC mobile while still having my surveillance device. Currently I don't have a phone.

The idea is, instead of playing a song. You can add it to the waiting list. So for example the player is on shuffle mode and you want it to choose songs randomly for you. But ones in a while you want a specific song playing right after this one.

It has an option to "Add to queue" and the song will be added into a kind of hidden playlist. Lollypop will play all the songs in queue first. And then will choose songs randomly again. This is just next-level genius.

There are few issues I have with LollyPop, tho. It has an option to use lots of proprietary services. I think to get music from the internet. Or listen to radio. I don't turn on these features. I don't know who it's for. I have all my music stored on the computer.

One more clear downside with this player is that you have to have a pristine, clean music files. With all the Artists Names, Albums Names and other metadata be just right. Or it's going to look like a pile of mess instead of a music collection.

`sudo apt-get install lollypop` .

Happy Hacking!

New Flagging System for LBRY / Odysee

We already touched upon the freedom to publish anything at all. But how about a freedom to avoid certain publications as well?

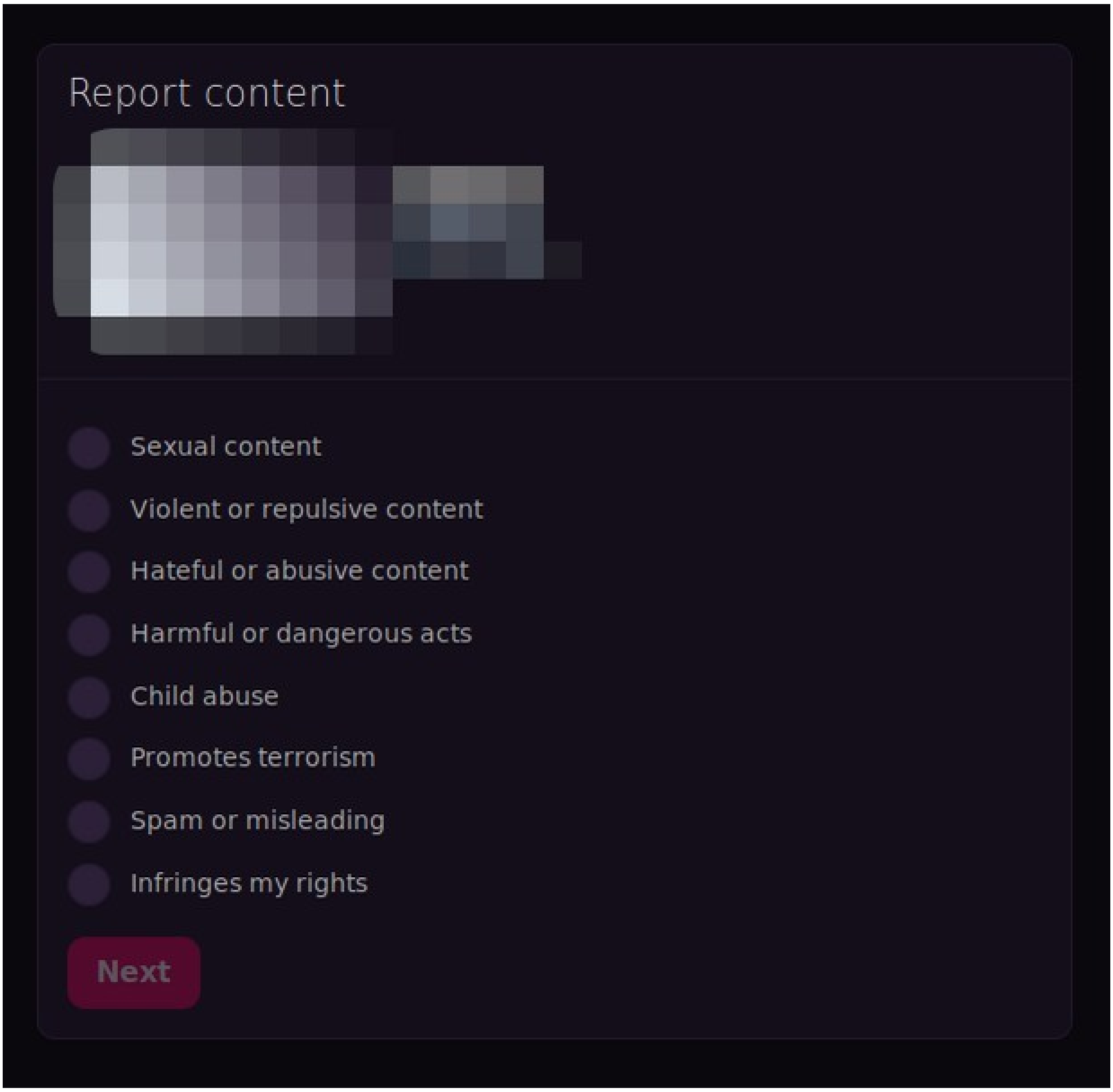
`lbry://@blenderdumbass:f/New-Flagging-System:c`

Not so long ago I written an article about whether "Content Freedom" should be a thing. Whether people should have an ability to upload anything they want. I concluded with that... Yes. But only with a sufficient enough warning system. Like a thing that warns you about what type of publication you are about to see, is present.

I want to dedicate this article to propose that kind of system. I already opened an issue on LBRY's git repository. Tho it's a bit broader than this article. I had thought about some things. And I have a few interesting ideas.

Flags now

Current implementation of a flag system on Odysee (and probably the rest of LBRY applications) at the moment works like this.



Report content

[redacted]

○ Sexual content

○ Violent or repulsive content

○ Hateful or abusive content

○ Harmful or dangerous acts

○ Child abuse

○ Promotes terrorism

○ Spam or misleading

○ Infringes my rights

You click the flag button and you get this menu. You can choose what kind of stuff you personally didn't like. And behind the scenes it will do one of two things. In FAQ about "content" they avoid the issue of the mature. And talk about mainly the DMCA stuff. That they touch on even more here.

The first thing that can happen. Is that if a DMCA request happens. The url of the publication will be added into this repository. And the LBRY developed software and websites will not show a publication if it's on this list. It's still on the LBRY protocol. It's not going to be removed. But you will need to modify your software to see it. So fork the LBRY Desktop please.

I think this is going to be done to illegal publications in the US where the company is. So anything where copyright is an issue and there is a DMCA request. And Child Porn will be removed.

For the rest of the publications it will probably add a mature tag on them. But I'm not sure. Because I couldn't find information about it. And I'm not willing to flag somebody just to check it out. If you know please comment about it.

Issues with that

Let's say that Lars Von Trier finds out about LBRY and wants to upload him movie, The House That Jack Built, to it. I gonna mention it again. I know, if you read my previous publications. I talk about this one a lot. But it's just a very good example.

If he uploads it him self. There is no problem of DMCA. Because he probably holds the copyright (The company that distributes the film is his). There is no illegal pornography in the movie. Even tho it could look like there is. Pictures of Murder of children are not illegal. Only sexual child images are illegal. So the movie is not removed with DMCA. And not removed because it's legal. Then what? Probably it will be under a mature tag.

I don't have a problem with tits. I enabled mature in the LBRY desktop. But I don't want child death scenes. What could be done?

Proposal 0.1

The first idea was to extend the mature tag to enable other tags like child-death or simple death to appear in the network. Making the choice of publication a bit more versatile. This has a limitation tho. If you make any number of tags that are strict. With the current system is 1 tag. If you add death it will be 2 tags. If there are 2 hard coded tags. It's not very helpful for people who are afraid or want to avoid other things.

What if there are only 10 people in the world who afraid from Cute Cat videos. And one would be brave enough to flag a video with cute cats in them. So the other 9 could avoid that video. cute-cat is not the type of tag that a reasonable developer would think about when designing a system like this. Should we just disregard this as not important? Or do we have a deeper system at our hands?

Another issue that I want you to think about. Is what will happen if people could tag things them selves. For publications that is not theirs. There could be a lot of trolls and malicious people that would abuse that system. How do we deal with it?

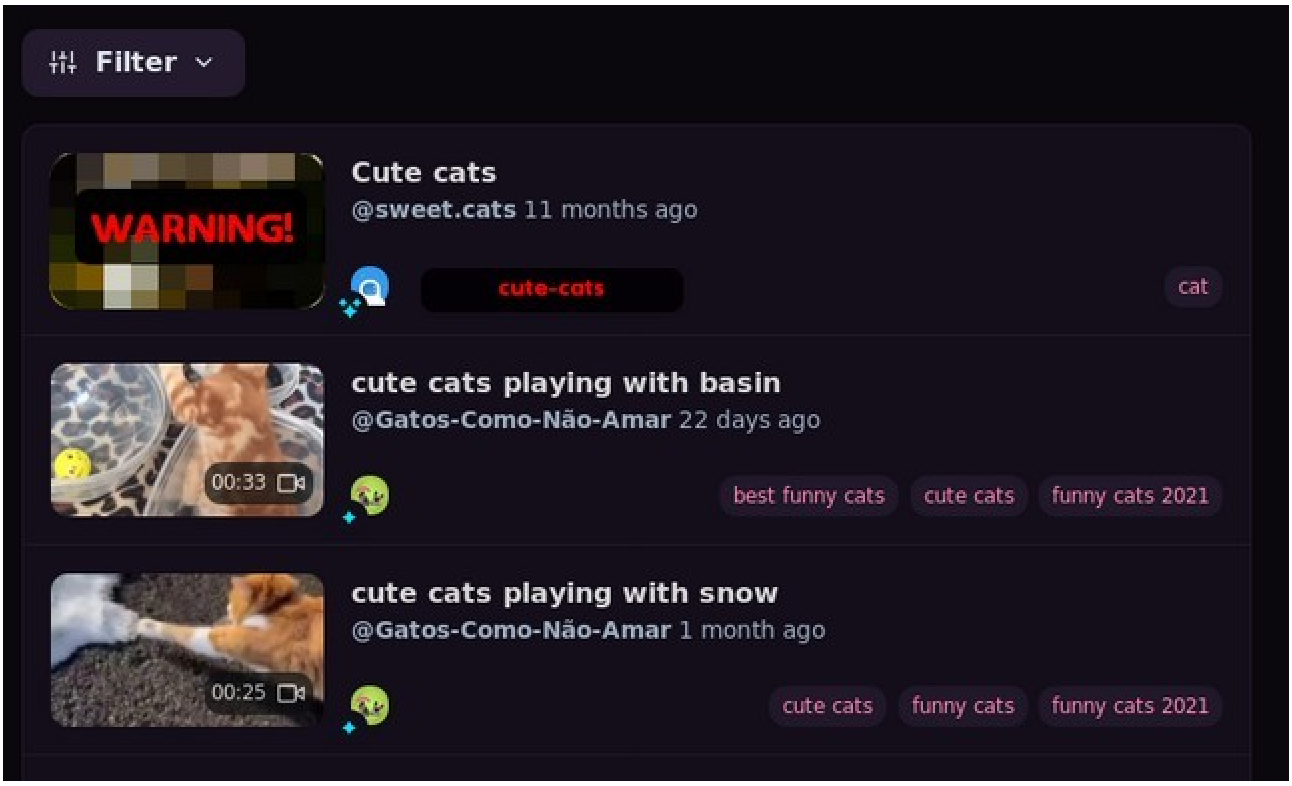
What do I keep in mind when I design?

In order to solve those issues. I want to talk about what to I think about when designing the system. And how I want to approach it. If this article will be seen by the developers and will be implemented. What were my motivations? So it's not going to go out of vision.

I am trying to insure maximum Freedom. And this kind of system would make sure both, that people can upload anything they want. And that people could avoid anything they don't want to see. It's their choice to avoid something or not.

It's not supposed to be a policing thing. It's not supposed to be something that makes the protocol this place where if I hate your opinion I will click the Flag button and add hate to the list of tags. I want it to be resistant to such nonsense. Even if it's allowed.

Proposal 0.2



Cute cats
@sweet.cats 11 months ago

cute cats playing with basin
@Gatos-Como-Não-Amar 22 days ago

cute cats playing with snow
@Gatos-Como-Não-Amar 1 month ago

So you come into Odysee and you type "cute cats" in the search. You get one publication that somebody flagged as cute-cats that so happened to be still not allowed by you. You can either click on the red tag. And choose allow. Or avoid the publication. If it's something you rather avoid.

If you accidentally click on the publication it would not play. You will see a big warning sign instead of the video player. And a list of flags on that publication.

I think a menu of options could be there both when you click on each individual flag. And for all flags together:

- **Allow**. Adding the flag to a whitelist of a kind. Making it ignore the warning of this particular flag in future. Similar to allowing mature in the current system.
- **Allow Ones**. Maybe you want to see this just now. But keep avoiding later.
- **Allow for X amount of time**. To be able to test yourself. You could choose a time while this flag will be visible for you. And then automatically blocked again.

- **Censor**. Make it so you never even see publications with this flag. Like if they don't exist in the platform. But obviously just for you.

  This solves it a little bit. For example if abuser wants to add some flags to it when they are not harmful or that are misleading. You will still be able to find the publication. And with a click of a button still see it.

  But it makes spamming flags be a thing. So how do we deal with it?

Proposal 0.3

LBC is a very good thing. It's just an essential part of LBRY protocol that I think we can use it for this system. When you flag somebody it's like making a little publication. If you have little LBC it's fine. The publication will take like 0.001 LBC. And you successfully flagged something.

Every other person may flag it again. Or support your flag with his own LBCs. Making some flags more noteworthy than the others.

In the setting apart from just having a simple white list. You can also have a setting of reason-ability.

Basically you choose how much LBC should be on a flag before your account considers it worthy of blocking.

We should make it a separate thing from simple publication. Because it should not give a monetary incentive to flag anybody. People would abuse the system making weird flags. So then if somebody supports the flag they could unlock the tip and run with it.

Also flagging or supporting flags should be limited in LBC. Let's say up to 5 LBC. Why? Because people with money could use it to put a very highly payed flag to a publication they hate. It should be multiple accounts supporting one flag together in order for it to be noteworthy.

There is no conclusion is this post. I don't want to conclude it yet. This proposal worth trying, but I think you will figure out why I'm wrong on so many things. This is a gift of freedom of speech. I propose an idea. You criticise it. We both improve the idea.

I want to read what you have to say about it. Please comment. And **Happy Hacking!**

Chrometophobia | Fear Of Money

One of the hardest things to overcome in life are phobias. Especially phobias from things that are so essential for living. Like the fear or water, the fear of food or in the case of this article, the fear of money.

lbry://@blenderdumbass:f/Chrometophobia-Fear-Of-Money:7

I have a terrible psychological thing that I start to realize is ruining me slowly. My unwillingness to be payed for. I will try to self-treat myself using this post. Trying to learn to like money a bit more.

For a few last days I was nearly starving. I'm currently starving. I just called my dad a few hours ago asking to send a bit of money to buy some food. How did it happen that a nerd. A guy like me who knows python. And knows Blender and 3D modeling, animation. A guy who does music. How is it that I'm starving?

Kauffman situation

Jeremy Kauffman is the CEO of LBRY Inc. the company behind LBRY and the Odysee. The platform where you are reading it. (If of-course somebody didn't copy it to some other place. Please do it. The article is under CC-BY-SA).

Few days ago I made a publication about my proposal for new flagging system on Odysee. And I uploaded the same text to GitHub creating this issue. To peak the interest of those who look at the code.

Jeremy Kauffman answered. He said, quote:

> @JYamihud want a job?

I didn't think long to reply. I replied with this:

> @kauffj Having a job will make me potentially fire-able. You can't fire a volunteer. And also. I need to loose weight. Las time I had a job I ate too much. So I'm willing to help in any way shape or form. But only if you don't pay me in a formal way. LBC support. I don't mind.

My girlfriend was very angry at me for doing it. But it's not the first time I did such a thing.

Studio

There is a studio in Israel, where I live, that makes CGI animation. I used to work at a store in a good neighbourhood. And a few industry people would buy there. And I would chat with them. One of them worked as a producer in that studio. I think it's called "Snowball Studios", but it was long ago. So I could be mistaken.

He saw my films and asked me If I would like to do animation there. Instead of working as a cashier in a store. And I declined. My excuse for it back then was that I don't want to have a contract that doesn't let me to work on my own stuff. But it was only an excuse.

What is wrong with me?

When I had a YouTube channel. I would refuse to sign up for ads. Because I thought it would make people annoyed. I did a patreon ones. And I was too ashamed of promoting it. So I just threw it in the description. And forgot about it. I don't even log into it anymore. So if you see me on Patreon. Don't waste your money.

2 days ago at work a cashier didn't come (I work in a different store now). And the boss asked me to be on the cash desk for the morning. Saying that she would pay me 100 shekels for it. I would totally help her. But I didn't want the 100 shekels. And then it struck me. That I need those. But I'm not going to ask her if she forgot.

She forgot. And I didn't ask her. And I'm still starving. It got to a point where I need to ask my parents for money.

Even tho I know that I'm going to have a decent salary next month. In 2 week from now. I still don't want to have it. It irritates me to have money. I want to pay the rent and get some food and forget about it.

The question is. **What is wrong with me?**

Chrometophobia

Chrometophobia is the fear of money. Chrometophobia is derived from the Greek word chrimata, which means money, and phobos, meaning fear. Chrometophobia is not only about physical handling of money but also about having money and its value. Sufferers tend to be homeless while not buy anything.

The fear is more complicated than it seems. It can be related to the pressures of managing money. Those who have less money are also prone to suffering chrometophobia, thinking about how little the money they have will make them feel anxious and will become their undesirable focus which will later affect their lives.

Symptoms of a panic attack include dizziness, sweating, numbness and confusion are the common symptoms of a chrometophobic.

Is this what I have? Do I have this disease? Or it's not that bad? I mean I see value in money. I just don't want to have extra money. I want to have just exactly the amount I need. And if I suddenly need more. I want to have more. This is why I did call my father.

What if I have a very early stage of this? How do I fix myself?

Attempt 1

As I mentioned. I did manage to open a Patreon some time back. But I would feel very badly promoting it. So I stuck it in the corner somewhere. And I would hope somebody would find it and give me something.

Maybe it's something to do with how I was raised? I don't feel like I need to reject money right away. But asking for it, is bad... Or something?

Attempt 2

At some point I did this:

lbry://@BlenderDumbass:c/oh-no-blender-dumbass-is-at-asylum-last:7

And the next video was this:

lbry://@BlenderDumbass:c/blendswap-model-review-the-pin-in:7

Basically. The first video was my actual emotional breakdown that I recorded on video. And when I got better. I thought to upload it but try to frame it as if it was a joke of some kind.

The channel, Blender Dumbass, was this parody idea. Trying to make the weirdest, but somewhat informative videos about Blender. Like for example simulating poo with blender.

People didn't really think much about my actual emotions. Because they knew that some stuff like that I could just fart out for the sake of a sick joke. And the next video. Where I describe the fiverr gig.

Where you pay me 5 bucks to request a video idea. It was a natural continuation of the "joke".

But the more sinister and dark truth of all of it. I was trying my best to come up with a way I could get money to eat. By playing this character, Pito Sage. Who is a greedy bastard that is not afraid of promoting something like fiverr for greed alone. If it's not me saying it. I would have less of a problem saying it.

After a while. If you look at the archive of my old channel, I stopped talking about the fiverr thing. And for a few live-streams I added this kid character that was holding a sign about the patreon thing again.

I was homeless

On January 30th 2017 I published a video simply titled "Fuck Off". Here it is:

lbry://@BlenderDumbass:c/fuck-off:d

In the video I talk about how my parents wanted me to leave the house. Because I have some kind of "Phobia of Jobs". Probably it's started very early on in my life and I didn't understand much.

I've gone to another town where I had friends. And they had me for a while. But later they too grew tired of me. And I ended up on the street. After a while I promised my parents to get a job if I could live with them. It wasn't for money. It was so they would allow me to stay.

Trying again

LBC makes me more comfortable to make than shekels of dollars. Since I don't see a clear way to convert LBC into shekels. This uncertainty makes me feel comforting. I can support channels freely. And I feel nice when people support me. But more often I use the wallet page as a kind of analytics window. Since the real analytics are very limited at the moment.

So here what I want to try. Because I think it can lead me into healing myself from this disease thingy.

Odysee just recently made it possible to comment and support in the same time. There is a new LBC button on a comment now. And I gonna see how much LBC was supported with each comment.

So in the next post that I gonna write tomorrow. I gonna select 5 questions from LBC supported comments. If there will be any. And answer them in a special section. I gonna choose 5 questions with the highest LBC count. Of course if you make one million of them I need to choose at random. There is not yet a sort by LBC comment filer.

So please ask questions about anything. This issue, computers, music, film, 3D graphics, GNU / Linux and other Free Software, surveillance and privacy. Or what ever else. That I might not know a lot about.

Happy Hacking!

My opinion on the "University of Minnesota banned from Linux Kernel" Drama

Sometimes people have a correct argument to make. Only they choose to make this argument in the most disrespectful manner. And even though they are free to do this. People are still free to react negatively to this.

`lbry://@blenderdumbass:f/University-Of-Minnesota:0`

University of Minnesota was banned from the Linux Kernel for a research paper on contributing bugs. I have a few thoughts about it.

Playful Cleverness

In the 1960s the word Hacker meant somebody who does weird things. Sometimes they would be helpful. A hack could be something similar to today's "Live-Hack". Anything outside of the ordinary, that requires special type of cleverness. Is a hack.

In Richard Stallman's article On Hacking he describes hacking as, quote:

> Playfully doing something difficult, whether useful or not, that is hacking.

Basically hacking is a kind of "Playful cleverness". Another thing I stole shamelessly from Richard.

In the 80s. Hackers. Clever people. Used to sit in front of computers. And hack them. Coming up with interesting ways to use a computer. Hackers came up with connecting a computer to the telephone line. And transmitting data that way. Inventing internet as a result.

Also some hackers would break security. And the news, which is only interested in the most horrible things, would talk about those kinds of hackers. By the 90s the word "hacker" already was exclusive to mean a person that breaks security.

Some Hackers like Richard Stallman were unhappy with this depiction of them. They coined the word "Cracker" to mean people who break security. But unfortunately this term is not used by the majority of the public today.

White hat and Black hat

Crackers *(I gonna use this term in respect for Hackers)* are divided into two subgroups. The white hats and the black hats. Black hat is your typical bad guy. Trying to break your passwords and or computer to gain access to your personal data. They might use tactics like social engineering and deception to get what they want. Usually for personal gain.

Basically black hats are crackers that would try to manipulate the voting. Or crackers trying to weaken the forces of countries. Or scam callers could be also called crackers. It's a type of social engineering. Or if you want a more well known black hat crackers you can look at Facebook or Google or Microsoft or Apple. They usually deceive good people into giving up their freedom.

White hat crackers are different. They crack security to patch holes in it. Like if I would squeeze a tire to check if there are any holes in the rubber. Some companies hire crackers like that to pen-test them.

Sometimes white hat crackers do it on their own. Trying various ways to penetrate particular security. And if they succeed, they are contacting the company, so they would patch the vulnerability.

I did a White hat Crack

I worked as a cashier in the supermarket in Tel Aviv. They had a cash desk running windows. I think since it's not my computer. It's not my freedom to loose. So let them run what ever they want. Even tho I tried to convince them to run Free Software instead. But that's besides the point.

They had a cash desk software open full screen. In order to open the cash drawer you needed a password. When you went away from the desk, you would press a button to lock it up. So nobody could press the "Open Drawer" button. Or if they managed to press it. It would not work. Since the software is under a password.

They had the windows bar at the bottom visible at all times. And there was always a second program running. Setting of some kind. So I decided to take a look. I tapped on it (It was a touch screen device). There were multiple settings for the cash desk.

One of the categories of settings was called "Drawer Settings". So I clicked there too. And there was a button "Test Drawer". So I pressed it. And the drawer opened.

I told that to the manager. And she told to the main manager. I discovered a vulnerability in the entire network of stores. You could open the drawer with 3 taps on the screen.

In few months they changed the software on all of the cash desks in the entire network. Unfortunately updating Windows XP to Windows 7. But it didn't have this same vulnerability anymore.

University of Minnesota situation

Not so long ago there was announced that the "Linux Kernel is banning the University of Minnesota" from contributing changes. Meaning that any pull request to the main branch of the Linux kernel, if it comes from an email with @umn.edu in the end, will be out right rejected.

Linux kernel is free software. So it would not disable them from working on their own forks of the kernel. Just that the guys in the main branch do not want to work with them anymore.

What happened?

The university published a paper titles On the Feasibility of Stealthily Introducing Vulnerabilities in Open-Source Software via Hypocrite Commits. Judging buy it's name, it's understandable why a decision like this would be made by the Linux Kernel developers.

But let's not judge the book by it's cover and take a look at the paper. Or at least at it's Abstract version in the beginning.

> Open source software (OSS) has thrived since the forming of Open Source Initiative in 1998.

Clearly they didn't do their research. The Free Software movement started in 1985 by Richard Stallman. And the "Open Source" was an attempt to make Free Software more commercially appealing for the businesses. By trying to hide the core principle (Freedom). This is why we try not to say "Open Source" and instead use "Free Software" or "Software Libre". That communicate Freedom better then "Open Source".

> More importantly, the OSS development approach is believed to produce more reliable and higher-quality software since it typically has thousands of independent programmers testing and fixing bugs of the software collaboratively.

This is why we need to use Free Software. People who promote Open Source tent to speak too much about reliability. It's technically possible. But technically Microsoft or Apple can make their proprietary software also reliable. And in both will still be bugs.

The fundamental difference between the two types of software is that one gives you Freedoms to Run, Modify, Distribute and Distribute Modified copies. Which are not present in full, or at all, in proprietary software.

The over reliance on the terms "Open Source" or statements like "More Secure" or "More Private" are misleading. Yes, with proprietary software security against the proprietor is impossible. And against others is unknowable.

Only with Free Software you have the chance to be secure. Since you can see the code for your self, verify it. Change what ever you want in it. It's not necessarily secure by default tho.

> In this paper, we instead investigate the insecurity of OSS from a critical perspective - the feasibility of stealthily introducing vulnerabilities in OSS via hypocrite commits (i.e., seemingly beneficial commits that in fact introduce other critical issues).

Basically they tried to look a possibility to fix some bug in some Free Software, but in such a way that it introduces another bug. And also that the code reviewer will not notice the bad part. And will approve the malicious patch.

> the Linux kernel is extremely complex, so the patch-review process often misses introduced vulnerabilities that involve complicated semantics and contexts.

I can't disagree with that statement. If you take in concentration how complexity of code rises with every added line. Making a vulnerability undetected is very much possible if you know how to confuse the reviewer intentionally.

> As a proof of concept, we take the Linux kernel as target OSS and safely demonstrate that it is practical for a malicious committer to introduce use-after-free bugs.

Meaning that they attacked the Linux kernel it self. To prove that they can do it. They claim it was "safely". I don't know how would they insure that a security vulnerability would be safe. But okay.

> At last, to improve the security of OSS, we propose mitigations against hypocrite

> commits, such as updating the code of conduct for OSS and developing tools for patch testing and verification.

Basically saying that free software code reviewers should be more careful with what code they allow. And they propose some "tools" to test the commits. Just out of curiosity I want to take a look at what "tools" are they proposing.

Okay so in page 12 they go over this in depth. They recommend stuff like incremental symbolic execution, alias analysis, concurrency analysis, indirect call analysis, common bugs detection and more...

Basically they made a huge job on trying to actually make Free Software more secure. So I can't really be mad at them after reading the paper.

Where I stand on this?

I think it's understandable that the Linux Kernel main branch would ban such a crackers from contributing. Needless to say, their proof of concept bugs are in the code somewhere and poor developers need to fish them out of there.

It's the feeling of you cooking a soup and a random stranger comes to you and throws something nasty in the soup. Just to later sell you something to prevent it from happening again.

Maybe better means of doing this research could be done. Like maybe making a branch for it. But then again. The code reviewers should not know that they are tested. Because then, whats the point. People who know that they are tested give different results from those who don't know.

I understand the anger. And also thankful for the paper. I see why the "Linux Community" would hate the University. And I can't deny that this is something worth talking about.

These types of attacks probably happen more often then we think and the University of Minnesota just alerted us. So we would know to look for them. Even if sacrificing them selves and the work hours of many, many people.

In the ancient times, so called research was a to take man, women and children and to cut them, boil them and do all kinds of other nasty things with them and see what happens. It's unapologetic. But in the same time. We have medicine now, thanks to it.

Life is uncomfortable. Deal with it.

Happy Hacking!

Is Anything Real?

Hm…

```
lbry://@blenderdumbass:f/is-anything-real:7
```

Inception is a very powerful movie by a very talented filmmaker. Christopher Nolan is one of those people I like more then regular folk. One, is because he made some of my favourite films. Not the most favourite. It will be Steven Spielberg and Stanley Kubrick with AI: Artificial Intelligence.

Christophers of the Hollywood. Nolan and MacQuarrie. Are very interesting. They are trying to preserve the film industry with shooting on film. Avoiding digital. In 24 frames per second. Using as much practical stuff as they can.

Nolan directed such instant classics like Memento, Prestige, Inception, Interstellar, Dunkirk and Tenet. And MacQuarrie joined forces with Thomas Cruise Mapother IV and made films like Mission Impossible 5, 6 and now 7.

With the over reliance on computer generated effects and comic book heroes. These directors pulling off practical, expensive block-basters, is a fit to admire. So if you are looking for name to give a boy. Christopher would be a great choice.

One more interesting fact about Christoper Nolan. And with it also Arnold Schwarzenegger. They both were born at July 30th. My birthday.

To throw this out of the way. I gonna focus this article back on track. And talk about Lucid Dreams.

Lucid Dream

An experience of Lucid Dream is when during the dream, while you sleep, you realize that it's a dream. And since it's all happening in your head. You can totally control it.

Many people find ways to do it. From constantly checking the reality to a few other methods. The film Inception shows one of the ways to check if you are dreaming. It's the totem. An object that you know will change it's properties in the dream versus the real world.

Like the totem shown in Inception. A spinning top. That keeps on spinning forever while in the dream. And falls in the real world. The movie ends on an uncertainty of whether we are still in the dream. Showing only a slight wobble of the spinning top before cut happens. And the credits roll. You are unsure if it kept going. You are unsure if it was all a dream.

I had Lucid Dreams

When I was 12, my parents brought me and my brother to Israel. We still live here. I was born in Ukraine. I was very well familiar with the house we had in Dnepropetrovsk. One day I realized. That I'm in the house. In Dnepropetrovsk. But we just moved to Israel.

This made me realize that I was asleep. That this was a dream. I thought. Okay. I will wake up. Then something completely strange happened.

I did woke up. But I stayed in Dnepropetrovsk. The blurry, dreamy environment changed to the at most realistic and sharp one. I woke up during the dream. From a dream with in a dream.

Suddenly here I am. Standing in the room that I know so well. But everything is touchable. I can feel smells. I hear sounds. It's like I was transported via another dimension. Like if I teleported there. Like if it was real. And me being in Israel was a dream.

Needless to say I was confused. I wanted to check if it's real or not. So I made the stupidest thing possible to check the reality of something. I walked out of the window. And climbed on the wall like a spiderman.

It worked. I was dreaming. Just I was dreaming the most realistic dream possible. So of course I took my gifted pleasure. And did all kinds of unrealistic, but interesting things. From flying and shooting lasers to dating all of the girls.

By the end of the dream it started to become random. Like if my subconscious couldn't find any more data to fill the blanks in the meaningful way. So I started seeing flashes of unrelated images. And suddenly... BAM! I'm awake. In Israel. For real.

Travel confusion

When we did move. The first night was a long trip from Dnepropetrovsk to Kiev on a night train. The train had beds and everything. We were in the open area for people with less money. Me, my brother and some random little girl took the 3 beds on the top. While our parents were in the bottom.

I didn't see a girl sleep before. I saw my mom sleep. I saw my granny sleep. But never seen a little girl sleep before. Combined with that we are going to live in a different country tomorrow. It was the first night I didn't sleep at all.

I still remember the sound of the train. The metallic rhythm of wheels hitting the connections of the rails. I was thinking about Israel. Looking at that girl ones in a while. Wondering how simple she looked. I thought it would look somewhat more interesting than this. I turns out that little girls sleep while laying down with eyes closed.

The rhythm continued. I heard the rhythm. The rhythm played when we came out of the train. The rhythm continued when we flied in the plane. I heard the same sound of metal hitting metal in Israel. Few days it was fading from my ears.

If I concentrate hard enough now. 11 years after. I can still barely hear this metallic sound in my head. Did I fall asleep in the train from Dnepropetrovsk? Am I currently a 12 year old boy going to Kiev. Is this all a dream?

eXistenZ

Think about a world where there are no computers as we know. But people still have computational devices. It's just they are not manufactured, fixed and designed by engineers. They are raised, healed and genetically modified by farmers.

eXistenZ is a movie about two Gamers stuck in an Inception like game with in the game. But a game that wasn't built on a computer. But raised on a farm. A device that plugs into a special port in your body. That looks like an alien creature with testicles for buttons.

The movie is built in such a way to question reality of everything. Are they plugged in? Is it all a virtual reality on a computer? Is this inside a game too? Is anything real? Is anything game?

Simulation Theory

Not so long time ago we could draw a couple of triangles on the screen. It was so slow that one frame from Jurassic Park where the T-Rex runs after the Jeep was rendering for 12 hours.

Today we have EEVEE. Real time graphics. Better handling of light, shadow and reflections. Working in real time. 320 frames per second. We grew so much in just a little more then 25 years.

Another 25 years onward. And we will be able to simulate the universe. If it is true. And a simulation like this is so inevitable. Then probably in the simulation there will earth. And on that simulated earth there will be technology. And in this technology there will be computers.

Simulation with in a simulation. With in a simulation. And Inception of simulations. With this being the case. The probability that one world being not in the simulation is very low. One in millions. If not trillions.

Are we simulated? Is Matrix real? Are we in the Matrix?

Conclusion

Confusions like this happen to me regularly. Currently I'm tired. Is when I go sleep, waking me wake up in the world of dreams?

Nobody commented a supported comment on the last post. Tho one user helped me a lot with my Free Software project. If you want me to review something. Give an idea for an article. Anything that will influence me forward. Seeing a supported comment makes me a tiny bit happier. I don't have a lot of readers. So I will probably see all of your comments anyway. But pressing this Support button helps regardless.

I want you to answer a question. Is anything real?

Happy Hacking!

Stop…

This is the part of the book where I transition. I finally covered all the lazy articles that I was making for no reason. From now on I would see myself as a kind of article-professional. I would make a yellow border thumbnail style. I would start writing better. I would include pretty images in the end of the articles.

It was a jump in style. I'm no longer the old, experimenting article writer. Now I am a real article writer.

To celebrate this moment. Let's focus in the book on something that I will take away from the articles. Let's take a look on the Moria's Race Project.

MORIA'S
RACE
OPEN MOVIE PROJECT

J.Y.AMIHUD'S
MORIA'S
RACE
OPEN MOVIE PROJECT

J.Y.AMIHUD'S
MORIA'S
RACE
OPEN MOVIE PROJECT

J.Y.AMIHUD'S
MORIA'S
RACE
OPEN MOVIE PROJECT
NEONSPEEDSTER

How To Make a Good Video?

Sometimes a new person may focus on the wrong aspects of a little film they are making for what ever platform they are posting this video on.

`lbry://@blenderdumbass:f/How-To-Make-A-Good-Video:d`

Whether on Odysee or on Evil Tube. If you have a channel. You want your channel to be successful. And one of the ways to do so, is to make good videos. Videos that people watch.

Recently I was coming back more and more to my archived channel (ones a YouTube channel). The interesting thing is that some of the videos on that channel I want to watch again and again. I made them. I know everything about them. And yet. I want to see them.

People strive to make publications relevant. Like trying to touch upon subjects that are widely talked about. I made it myself. Touching upon the return of Richard Stallman to FSF and University of Minnesota. But subjects are like stories. And what story is told is not a big question. Even tho it is a question.

The bigger question is **how** the story is told. Does the speaker knows from which point to start? Which speed to use? Where to make pauses? What intonations to use? And so on...

I don't think I have a definitive answer on **how** to make videos interesting. But I think I might put you on a journey that will take you there. So let's begin...

Perfect Audience

On multiple occasions. In the older interviews with Steven Spielberg. The director of such classics as Jaws, ET: The Extra Terrestrial, Schindler's List and Jurassic Park. The interviewer asked him how he makes such a great movies. And he answers that he thinks, that he might be the "Perfect Audience". Perfect Audience? Isn't he supposed to be the Perfect Filmmaker?

He explains it in this interesting way. Being the Perfect Audience made him know exactly what people want to watch. Because it's what he wants to watch. And the job is to make his desires. The stuff he wants to watch to become a real movie. The hard thing is to make the vision to become reality. The vision tho, is very important to have.

Quality?

A lot of people buy expensive gear. Shoot at 4k or 6k or 8k. Making the videos 60 fps. Pixel perfect. Zero compression. But nobody wants to see it. Other people release a 720p, black and white video. And there is a huge following. Why is that?

People usually don't realize that in any type of production. Every level of it matters. And sometimes some things are more important then the other.

Sound is more important than the image. You probably heard that saying. People rather see a terrible looking movie that sounds good. Than a good looking movie with a terrible sound. Try it yourself. Take a film that both looks good and has a great sound. On one side fiddle with the image. Compress it. Make it black and white. Reduce the resolution. But don't touch the sound.

On the other side. Take the same 4k, high dynamic range image. But do weird things to the sound. What version would you enjoy more?

Other things like good screenplay can make or break the film. While visual effect. Yes, they are important. But not as much as a good screenplay. If your characters are dumb and one-dimensional. No matter how much VFX budget you have. It will not save the movie.

Taking priorities like that could be very important. The screen play is not the story tho. It's a particular way of telling it. The story is also important. But not as much as how you tell it. Think about how boring would be the movie Memento. If it was told linearly. From the beginning till the end.

How to find what's important?

A strategy could be developed. Something that grabs people in. And keeps them through out. And keeping this strategy through out the film or video will help a lot. For most of the videos of my old Blender Dumbass channel. I took the inspiration from the PewDiePie editing style. And the "never saying anything too seriously" thing. And tried to combine it with a boring Blender Tutorial. The resulting mayhem looks like this.

lbry://@BlenderDumbass:c/blender-sequence-editor-green-screen-no:0

This was the most popular video on the channel when I still used Evil Tube to host those videos. As you can see the video is extremely low quality. I recorded the screen in 4:3 making space for the camera view on the side. Something that's probably contributed to the fresh feel of the video. I used a very old camera. I was lucky that it had a descent built-in microphone.

While recording I knew roughly what I would do. Since you see me having a working example in the beginning. I had prepared the background frame. And the footage. I thought about potential jokes I could make with them. So I named the files accordingly.

I knew that jokes would be layered one on top of the other. Making you discover more things to laugh at when you watch the video a second time. This creates an necessity to watch it for the third time. In film it's usually done with a plot twist toward the end of the film. Giving a whole new context to what was happening before.

I didn't know English well back then. So my weird, cursing language was a part of the performance. I didn't prepare any script. I hit record. And struggle to explain how to do a thing in blender. What ever happens, happens.

I knew that the editing would be the interesting part of the process, where I would zoom on my mistakes. And do all kinds of creative stuff with the video. So while recording I would imagine the editing already. Trying to take mental notes. I knew that I had to do enough random things in order for the video to be interesting. Because there should not be a single dull moment in it.

As a kind of hook in the beginning of the video. I decided to surprise the audience with playing a guitar. It has nothing to do with the video's subject. But it tunes the audience into the chaos, where anything can happen. And it's by design. Giving me an authority to do any mistake what so ever. Or any spontaneous thing that I suddenly will come up with during the video.

But I want to make a serious video

Let's look at another channel. Here is a video of Doug DeMuro. A guy who reviews cars. I gave you an Invidious link since he isn't yet on Odysee. And I don't want to subjugate you to Evil Tube.

This is one of the most popular channels out there. Despite having the most boring idea ever. A review of a car. With the most basic production possible. Shooting probably on a phone. And without a crazy editing like in my videos. His editing is straight forward.

His success could be broken up into two parts. One would make him successful as a talking head. It's his character. A guy who wears casual to anything. A guy who will be more interested in the infotainment system manual. Then the actual car. He is a free looking. Free feeling. Smiling dude. That would not make you care about how he looks. Or how the stuff that he makes looks. But you just want more of his quirkiness.

The polar opposite of his style would be somebody like Markiplier. No matter how hard of a subject matter he takes about. Or how stupid a game he plays. It's interesting because of how he talks. Straight. Precise. With perfect intonation. The kind of voice you would enjoy reading a book. The kind of voice that a radio station would kill to have.

Doug DeMuro's second power is the detail. His channel is not about the horse power of the car. Or the time in which it does 0 to 60. Yes he mentions those things. But it's not what he is all about. His channel is about bringing you the idea of how it would **feel** to own a car like this. This is why he reviews seemingly uninteresting things. The cover of the user manual. The weird Easter egg in the infotainment system. The feeling of sits. The smell.

If he would review a peace of hardware. He would go over how quirky the texture of the box feels. Making you feel like if you owned that thing for a duration of the video.

Good movies that are told subjectively use that same trick. They put ordinary people in extraordinary circumstance. And try their best in making you feel what it feels like to be there. Avatar is about visiting the world of Pandora. And feeling the aw of seeing those bio-luminescent plants, or the flying mountains, for the first time.

Titanic is about experiencing love. And experiencing the sinking of the Titanic. War of the Worlds is about the confusion that happens when you invaded by martians. About the experience of fear, horror and tension. When you don't know what horrible thing that will happen to you next.

Tension

I gonna tune a guitar. I gonna start turning the nob. The string is getting tenser. It might snap. I might get hurt. I keep turning the nob. I check the pitch and it's getting higher and higher. I keep turning the nob. Now you are scared. The pitch shouldn't be that high. But I still keep turning that damn nob. Oh no. It's gonna snap. You all sweaty. But I keep turning the fucking nob. Until eventually I stop.

This is tension. It's used in films very well. The most basic form of tension is when the time is running out. If you communicate clearly that some task should be done in some amount of time. Watching this task becomes tense. And with it interesting.

Tension can be done in a number of different ways. Like withholding some information that might or might not lead to something of a value. But in such a obvious way that audience will feel the tension.

When will it be revealed? Will it be revealed? Is it what I think it is?

Or you can hint that something bad would happen. Like starting the video with "Why did I do that? It was a mistake.". Or playing a game that clearly establish ways to fail.

To make the tension, the stakes should be clear. You are probably going to loose the level. And it's this one time that you got so far into the level. What a bummer it would be. The string would snap and hurt you badly. What a horrific scene it would be. You will loose something if you are not doing it in time.

Putting at least the most basic form of tension in every scene will make the movie at least watchable. To put it masterfully in every scene. Will make you on the edge of your seat. Holding a level of tension in a video for a channel on Odysee is not a necessity if you have other things that are interesting. But actually managing to do it too. This will blow your video into another dimension of cool.

Conclusion

You probably noticed already that I talked more about film and less about videos. How is this supposed to help you make a good video? Videos are Films. They are small, tiny, low budget films. With a director. Sometimes an actor. And an Editor. Often in one person.

The final video is a film. And how you make this film matters. How you tell the story matters.

Happy Hacking!

What If...

This is a start of any good question that results in a good exploratory work. Fiction or not.

lbry://@blenderdumbass:f/what-if:2

Yesterday on an Odysee stream by @OfficialZaney I asked him to react to this page by the FSF. This is a campaign. A petition type thing. To persuade Microsoft into releasing Windows under the GNU GPL license. Making it Free Software.

Zaney's response was something along the lines of "This is not gonna happen and here is why...". But this is not something I wanted to hear. I wanted him to think and give his view of **what if** it would happen. Probably had to explain myself a bit more clearly.

What if Windows became Free Software?

I don't know for sure. And that's the point. When I want to look at a situation like this, I want to think like a writer of science fiction. I'm not concerned whether I am going to be 100% correct. It's about speculation.

I think first thing that's going to happen is that GNU/Linux distros will start taking peaces of Windows. Things like Wine will become more capable. Maybe even distros like Ubuntu will become fully compatible with the .exe files.

Second. People will start forking Windows. Removing telemetry and auto-updates. Making more themes. Adding things like Desktop Environment support for Windows.

It seems like at first there will be a strong opposition of people. People will shout things like "Embrace, extend extinguish" on all of it. People will think that it's all just another Microsoft's move to kill "Linux". Because people are not very educated about the Free Software.

For the hard core Free Software people it would be interesting. Maybe Richard Stallman will make an installation of a completely Free Windows fork. I think Microsoft will not make something like Trisquel OS with Windows. It will resemble something like Ubuntu at best. But it's more reasonable to assume what it will be as Free as Chrome OS. Which will require some additional clean up. Similar to the Linux-libre.

After some time, due to entropy Windows kernel will be compatible with GNU software and vise versa. Things like GNU/Windows will be a thing. And you could call Windows a Linux Distro. It will be mayhem. But in a good kind of, Freedom respecting way.

What if... as a topic for a story

The example I just gave you with Windows being Free Software. I wrote it with minimal research. Just by feeling. I came up with all of it on fly. And one thing simply lead to another.

With a tiny bit more research you could make a better version of it. Maybe, perhaps you could go the hard science fiction way and research everything as precisely as you can. Making another The Martian book. Or Ready Player One.

The Martian asks the question of **what if** a botanic genious was left alone on Mars. Ready Player One is probably not asking anything and just trying to be something "for the nerds". But if it would. It would ask something along the lines of **what if** a VR game became so successful that it's dangerous in the real life to play it.

For any fictional story you could ask a **what if** question that the story wants to answer. **What if** an alien was left by mistake on earth and found a little boy that became his friend? There goes ET: The Extra Terrestrial. **What if** people could share Lucid Dreams? There goes Inception.

Some stories are probably a little deeper then this. Or they ask more **what if**s. Or more layered **what if**s. **What if** people could have super-powers is a common one these days. But if you add to it also **what if** we combine multiple movies about superheros into a single movie. Then it becomes Avengers or Justice League. A layered **what if**.

What if... as a topic for a video

If you are a LBRY'er, Odyser... (I'm trying to come up with a word like "YouTuber" but for a better platform), then you may want to use **what if** in your video ideas.

For example. Let's take a channel like @OfficialZaney who is focusing of reviewing software and games sometimes. Like this video by him:

https://odysee.com/@OfficialZaney:8/jump-limited-playing-a-game-made-in:b

I know. I plugged him, plugging me. By the way. This is the game. There is a pink Download button if you are using Odysee.

But think about how much more interesting the video would be if he would go around and give suggestions to things. Similar to any Free Software *issues* page with bug reports and feature requests. Some will not be implemented. But imagining is sometimes enough.

In 2013 Blender Guru (a Blender based YouTube channel) released this video. I gave an Invidious link. Please take a look at this video. It's a UI redesign proposal for Blender. It didn't happen. The UI has gone a different direction. But the video is still very interesting.

This is kind of doesn't make sense. It didn't happen. So why bother talking about it?

Steam Punk

Steam Punk is an art style. A concept. **What if** instead of electricity people would further the steam technology? **What** would today, or even future look like, **if** the progress had gone in a different direction?

Those kind of questions are endless. But they are interesting to ask. Even tho it makes no sense to ask them. The petition thing about windows becoming Free Software is an exercise on **what if**. Similar to Steam Punk being an exercise on **what if**.

Exploration of realistic concepts...

Any **what if** is a concept. Either a design of a different world. Exploration of ordinary mixed with extraordinary. Or exploration of things that are ordinary but interesting.

What if a regular person met an old friend... In a bank... While the old friend makes a robbery? This is possible. But an interesting concept non the less. It worth exploring it. Writing a story around it. Even if just a small scene.

What if a mafia boss would give an assignment to one of his gang members to spend an evening with his wife? Ludicrous. Unlikely. But possible. And worth exploring.

"What if" gives people inspiration...

What if there was a platform like YouTube, but decentralized like BitCoin? This was probably the question in the head of Jeremy Kauffman. This question probably lead him to make LBRY and Odysee. I'm not sure that it was the question. But **what if** it was?

I can go so deep into this. Everything becomes interested all of a sudden. With enough good questions like this and interesting answers to them. You can make things possible.

What if a computer would understand English? Before this question there were punch cards. With binary commands. After this question there were programming languages.

What if you could take a phone with you on the walk? Now we have cell phones. The implementation sucks but the question was outstanding.

Assignment

I want to make a few **what if** articles. And I want **you** to come up with the topics. There is a comment section below this article. Please add your suggestions. If they will be Supported. I will appreciate it a lot.

If you want to make your own **what if** article. Please do so. Steal suggestions form the comments. Come up with your own. Give me a link.

Happy Hacking!

What If There Was No Linux?

Linux is a famous kernel used in multitude of operating systems. The most notable example is the famous GNU / Linux operating system. But there are also systems like Android / Linux or Busy-Box / Linux.

`lbry://@blenderdumbass:f/what-if-there-was-no-linux:f`

Linux is our beloved Kernel. Developed originally by Linus Torvalds. And since by many, many contributors. We all love Linux. And even if some of you doesn't use it, we can't overlook it's effects on the modern world.

In the spirit of yesterday's article about the ideas of "What If..." I wonder. **What** would our world, or at least our technology side of it, look like **if** Linux the Kernel didn't exists.

GNU Hurd

GNU Hurd is a kernel. In the 1980s when Richard Stallman started the GNU project. His aim was to make an entire Free Software Operating System. The GNU project grew. And by 1995 they had almost all of the components. Some were developed by Richard Stallman him self. Some by other contributes. They had replaced all of the system utilities of Unix. And the last part was the kernel.

By that time the GNU project was the main Free Software project in existence. People were excited about what would this project produce as a final result. And Richard Stallman had a lot of plans. He thought that making a mere re-implementation of Unix isn't enough. And they needed a more elegant design.

They came up with such design. And started working on it. It's called the GNU Hurd Kernel. And it's developed to this day. But since Linux kernel was licenced under the GNU GPL. And it was very much plug and play solution to the missing bit. It was natural that people would make a Free Software Operating System on this kernel.

Some people. Mainly Richard and those working on the GNU Hurd didn't like Linux at first. They were convinced that their new design would be way superior to anything by Linus. Since Linus just re-implemented Unix all over again. And it could be true. But Linux was so much easier to maintain then Hurd. So the GNU / Linux happened. And development of GNU Hurd was slowed down a lot. This is why 25 years later. GNU Hurd is still in alpha. Linux happened.

But **what if** Linux didn't happen? Then naturally GNU Hurd would. The development would continue full throttle. And in few years they would have something working. The Freedoms that a kernels like Hurd or the newer RedoxOS could've give are interesting to imagine.

No need to restart a computer when installing drivers. Since each driver is an application. Running in a user space. The kernel it self could have modules for tasks. And they would be install-able if needed or removable if not needed. Swapping of functionality. Like multiple drivers for keyboard or mouse inputs would be easily implementable. Similar to how we have Desktop Environments on GNU / Linux.

Binary Blobs in the Kernel would be an easy to configure thing. If you want to subjugate yourself to NVidia. You could install their binary blobs separately from the kernel. Keeping the kernel always libre.

Penguins

If Linux would not be a thing. Then Tux. The Linux mascot would not be a thing either. People would draw GNU everywhere. There would not be Super Tux Cart. Or Super Tux. But probably there would be more games about the cow. The GNU. Super GNU Cart? Super GNU?

What would the system be called?

Today there is a polarizing topic about whether to call the OS Linux or GNU / Linux. And you can take your stance. But I agree on calling it GNU / Linux.

The idea behind that, is that Linux is merely a kernel. And there a lot of GNU libraries. The entire utility stack was developed separately from the Kernel. Think about it. The Terminal. The Bash. It's a GNU package.

You may say that if Linux wasn't a thing. And Hurd was finished earlier. You could call the whole system GNU / Hurd. But it's only if there was a mistake that there was originally with Linux. I think the system would be called simply GNU.

There would be distributions the same way they are now. They will be called GNU Distros. I don't know enough history to know whether most popular distros would be created if there would be no Linux. I don't know if Fedora or Arch would exist. But there is a Debian GNU/Hurd distro. So I assume there could be Debian.

Linus Torvalds

If Linus wouldn't have developed Linux. There would not be a big name of Linus Torvalds. At least not as "the developer of the operating system". I think this role would take Richard Stallman instead. A lot of people say that they dislike Richard Stallman because he didn't make the Linux Kernel. Well. He was trying to make a better kernel. Somebody just beat him to it.

The differences between the two. Linus and Richard. That would probably be a big factor in what would happen with the system. And with the whole technology world. Linus is a business oriented. Open Source camp person. Promoting things like stability. Code quality and things like that. One of the things he said I remember very well is :

> Nobody can do perfect code except of me.

Richard Stallman is a different kind of person. A dude who is not very concerned with how much business-able is what he does. But more focusing on Freedom of all people. They both can be outrageous. And disrespectful in their own ways. But for Linus is because he is this cool motherfucker. For Richard is that he is not very careful with how straight to the point he may go.

It's rumoured that Richard Stallman has an Asperger syndrome. Making him not very understanding when it comes to politeness. And making people comfortable. In Richard's head there is only logic. If something makes logical sense. But makes people uncomfortable. He will not try to go around the subject. Try to prepare the person he is talking to. He assumes you will think logically. And understand his point immediately. So he hits you with it.

"Open Source" started because people were trying to sell the idea of developing Free Software to companies like Mozilla. It would not be simple to convince a guy who only thinks in terms of maximizing shareholders value. When your software idea has the word "Free" in it.

If there would not be Linux. The state of Free Software becoming viable to businesses would require another few years. Or even maybe a decade.

Making things like Firefox appear only in 2010. But maybe. Just maybe the timing would be saved by some clever people.

"Software Libre" is one of those terms Richard Stallman would use. Since "Free" is ambiguous and has two meanings in English. Richard understand why people want to use phrases like Open Source. But for him it's not enough to make the source code available. It's should be also Liberated. Free as in Freedom.

The term "Free Software" has a bug. But it's only a bug in English. Vrije software, Freie Software, Software libre, Logiciel libre, Свободное программное обеспечение. In other languages they use different words for "Gratis" so the problem never becomes a thing. It's inherently an English problem.

Probably one of the things that would happen in the world of the GNU system. Is that people will be more focused on Freedom of a given software and less on quality of it's code. Making Windows or Mac users arguments about quality completely worthless.

For GNU users arguing with non-GNU users. It would be a different battle ground. Since in English you would have only one thing to be confused about. And in other languages there will be non.

Android

Half of why Android is a thing. It's because it was based on the Linux kernel. So there will not be Android. But there could be something similar built on GNU Hurd. Only this time it would be more flexible.

I think if the Free Software would be such a big thing. And Open Source would not be. The developers of Android would've done something similar to what the developers of Librem 5 or PinePhone are doing now. It would be a phone that both has a pretty UI. And both respects Freedom. And much earlier.

With an explosion of the SmartPhone market. This could mean that there would be a choice between iPhone's non-freedom. And Android's total GNUness. I even think it would be called something like GNU Phone. And people with these devices would become more expecting of Freedom.

Probably it could bring Apple to an end. Or make Apple consider becoming a Freedom respecting company. Since users would not except anything else anymore. This would ruin buisnesses of anyone who want to subjugate peoples Freedom.

This thing would completely break the business practices of Microsoft. Making them release Windows as Free Software, or die. Smart people will find interesting ways to market, sell and otherwise monetize Free Software. Software Libre is not against selling it. It's just harder to convince people to pay for it.

Conclusion

I don't know what exactly would happen if Linux kernel didn't exist. But I took the optimistic route. And imagined a good type of alternative reality. But **what if** not all would be so great? What if this would make GNU disappear. And **what if** the Hurd is unfinish-able.

Happy Hacking!

The "YouTuber" on Odysee?

On YouTube it's YouTuber. But we are not using it since it's not Free Software. Odysee / LBRY is Free Software. But what should an author be called on Odysee?

`lbry://@blenderdumbass:f/The-YouTuber-On-Odysee:7`

I would write a longer article. I think on going into "What if there was no GNU?" but it will require a few hours of typing, which I unfortunately don't have right now. Instead I will focus your attention on a different topic that I wanted to touch upon.

On the Evil Tube the publishers are called "YouTubers" now a days. But what is a similar word that could be used for a better platform like Odysee or LBRY?

LBRYer ODYSEEr

Those 2 I came up with in the article about "What If..." as just a quick idea. I knew right away that it didn't sound good. I wanted in a way a better suggestion.

LBRYian or Librarian

This is already better. Suggested to me by @polarhive. By the way. Check his channel. He is hardcore in Free Software. I like it.

Calling people Librarians on LBRY would make a lot of sense. Since the LBRY should be pronounced "Library". But most people use Odysee. And they don't even know anything about LBRY. So I think it has a tiny flaw. Tho can be helpful to teach people about the protocol.

Astronauts

On the Odysee side of things. The whole 2001: Space Odyssey references made it so any new channel has a logo of an astronaut. Making all Odysee users astronauts. But is it hold true when a user has an identity? What if he is no longer an Anonymous user with no image? Is he still an astronaut?

Author / Publisher / Content Creator

The less interesting ones for sure. But could be used. Even tho "Content Creator" sound like a corporate talk to promote copyright powers and use of DRM. Maybe let's avoid it.

The obvious problem with this approach is that it's no longer advertise the platform. Maybe you can use that if you publish on multiple platforms. Like if you have a YouTube Sync. You can say on your channel "I'm no longer a YouTuber, I'm an Author. Since I'm no longer exclusively on YouTube.".

For those exclusively on LBRY / Odysee. What is your identity? LBRYer, LBRYian or Astronaut? Or something else?

Happy Hacking!

What If There Was No GNU?

We already talked about what if there was no Linux. But Linux the kernel is only half of the picture. It's only the kernel of the operating system. But what if the other half was missing? What if there was no GNU?

`lbry://@blenderdumbass:f/what-if-there-was-no-gnu:0`

Two days ago I wrote an article about **what if** there was no Linux. It's from the "what if" series. A series of articles about speculations on various topics.

The Linux Kernel one was interesting. I realized how massive the GNU side of things was. And so now I want to speculate about **What if there was no GNU?**

Could there be no GNU?

The GNU project was started by Richard Stallman in January of 1984. This was not something that was designed before hand. It was not like Richard always knew that in January of 1984 he would start this Operating System project that will be Free Software. It was instead, a reaction. Reaction to what was happening in the world at that time.

In the 1960s and 1970s computer hackers were not subjugated yet. When a manufacturer of a hardware would sell the hardware. It was either very basic in design. Or had a manual with explanations about everything. If it had software to control the device. The source code for this software was coming with the device.

It wasn't the Free Software as we know it now. There was no 4 freedoms yet. But it was Free in the sense of you could do with it what ever you wanted. You purchased the device. You purchased the copy of the software with it.

One day to the AI lab at the University of MIT came new printer. The old one was very slow. And had jammed a lot. They had to modify the software on it to make it tell the user when it's jammed. Since you would run to it every second to check on it. If it had a warning message, you would waste less time. So they made such a warning message. It was a long time ago. It was a huge machine. And you would not have it standing in every office like today.

Then came this new, cool and slick printer. It would print so much faster. And would print so much more accurately. The circles looked like circles. And not wobbly noodles put in a shape of circles. But after a bit of using. The hackers realized that it would be jammed just as much as the old one.

So they figured it out. They had to add the same modification to this printer too. Edit the source code. And make it so it would notify the user if it's jammed. Like they did with the other one. But there was no source code. Alright, don't panic. They could forget to include it. So Richard went to the manufacturer and asked for the source code. He met with the developer. And the guy said that he signed an agreement that he will never release the source code to nobody. Richard Stallman stood there. Unable to comprehend the weight of what just happened.

This kind of thing was happening again and again, more and more. And Richard Stallman grew more and more tired of this kind of disrespect from the developers. How could they use something that they can't modify? What if they have a use case that the developer never thought about? Where is my Freedom?

So by 1984 Richard Stallman was going to give up computers. And give up programming. And become a waiter. A job, he thought, would be way more respectful. Way more good. Than developing this new kind of corporate, secret, evil, software. He would no more, touch computers. No more, fix computers. Computers were the past for him.

But then he realized that in order to stop it. They just needed to write enough Free Software. And could get their Freedom back, this way, if they did. And he was the obvious choice for that kind of job. A computer programmer. So that's where GNU comes in.

If he would not think of it. If he would instead give up the computers as he originally thought to do. If he would become a waiter. There would be no GNU.

Would there be Free Software?

You probably would argue that it's - no. But there were other hackers. And they too wanted freedom. There was Free Software developed by other hackers before Richard Stallman made GNU. It wasn't the same kind of Free Software tho.

When software copyright started to become a thing. And companies like Apple and Microsoft started to build their businesses on people's inability. Legal inability. To modify software. Some hackers tried to fight it with early attempts. They would give with their software code, something like "Copyright by James Hacker, I allow you to do with this code what ever you want."

This is technically Free Software. You can do what ever. Meaning you have the 4 freedoms. But there is an inherent flaw.

All people want control. And a reasonable level of control a person should have. This is an issue every time when a new country writes a constitution. They set "rights". Rules that make sure people can do certain things. And be protected to do those things. Like the freedom of speech.

But how are those rights chosen? Freedom of Speech is a type of control. You can control what you say. And what you don't say. But you don't have a right to control speech of somebody else. You can't tell me what to say and what not to say. Since it's not your Freedom. It's mine.

There are two types of control. Freedom and Power. Freedom is when a person controls him self. And things belonging to him. Power is when he controls somebody else. We all need Freedom. Power is used only when one freedom takes away freedom of somebody else.

Like person A has freedom to have his property. And person B has freedom to walk where ever he wants. But if he would walk onto person's A property. Without person's A permission. It would strip person A from some of his freedoms. So Law Enforcement, a form of Power, is there to insure that one freedom doesn't take away other freedoms.

If you can do what ever you want with a software. You can take this software. And make a non-free, proprietary software from it. Taking freedom. And turning it into power. This is ain't good.

During the beginning of the GNU project this was one of the major concerns of Richard Stallman. How do they release the software in such a way that Freedom always stays there no matter what? So they made the GNU GPL license. Inventing CopyLeft as a result.

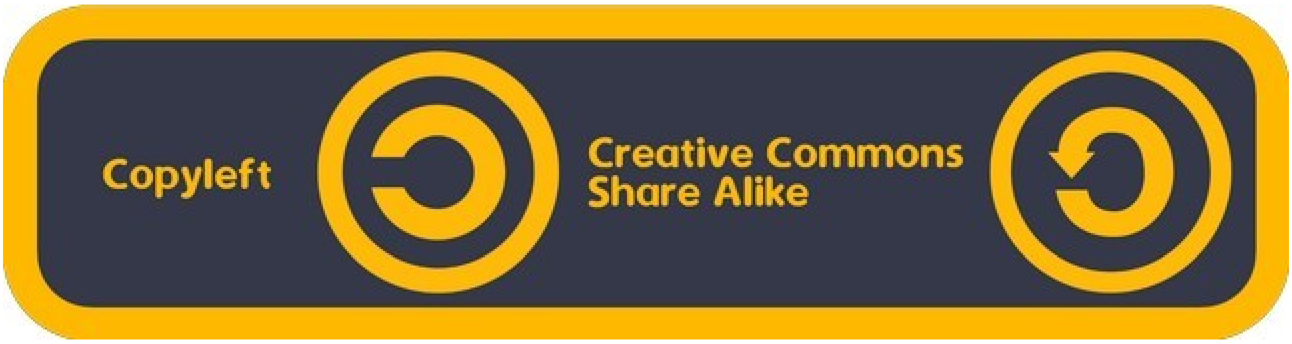
All copyleft does is prevents turning Freedom into Power. If you edit copylefted source code. You are not obligated to release your version. Just if you do release. It has to carry the same, copyleft, license.

If there was no GNU. There would be no GPL and there would not be CopyLeft. But there would be Free Software. Only it would be turned to proprietary very quickly. Maybe somebody would keep a fork of it free. But it would probably not be significant enough.

Would there be Creative Commons?

I think that it's a - yes. It doesn't seem like Creative Commons started due to GNU. From what I see it was a separate thing. Maybe, possibly, influenced a bit by Free Software. But since there were other hackers that used to release Free Software without the GNU GPL. It's safe to assume that Creative Commons or something similar would exist anyway.

How about the Creative Commons Attribution Share Alike? This is a CopyLeft license. This was influenced by copyleft. Since the share alike logo is a modified copyleft sing.



If GNU didn't exist. Copyleft wouldn't exists too. And probably the Share Alike licenses wouldn't exists as a result. Maybe they would. But the logo would be different. This is such a speculative territory, that to make a novel in this area I would just throw a coin and then choose. Since it's very hard to find an answer that would be satisfactory, easily.

Would there be Linux?

Now most of you scream that - no. But I think it's a bit more complicated. Linux was developed due to laziness. A programmer wanted to experiment on a Unix machine. That was placed in a specific room. In a place. And he had put a coat. And go through Swedish snow. Just to use that Unix computer. But since installing Unix wasn't possible on something close. Linus Torvalds just implemented the same features as in the Unix kernel. But put them on his own machine. So he could test things easier.

There was a community of hackers with or without GNU. So probably Linus would have been made this kernel of his public. But since there would not be a project of building a wholly Free operating system.

There would probably never be a Free Linux based system.

People would still be able to choose what components to install. Similar to Unix. But there would not be as much attention to the Linux kernel. The development of this kernel would not be as fast. It would probably be very unpopular. Or even forgotten.

To illustrate you how I imagine Linux without GNU. You can look no further then to ReactOS. A Free Software re-implementation of Windows. But since there is no project that wants to build a wholly Free Operating system on the Window kernel type technology. ReactOS is unknown, buggy and almost unusable. Nobody was enthusiastic enough about it. Nobody saw it as the last missing peace. So there was no rush to use it. No headlines made. It's just got forgotten.

Linux would exist. But it would not be covered by GPL. So forget about Freedom in the consumer market. Freedom would be only available to the unknown hackers. Until maybe another Richard Stallman would appear and make the GPL.

Conclusion

The world without the GNU, would be a very different world. A world where there is almost no freedom. A world where even tho Free Software exists. Nobody would protect it from being exploited and tuned into instruments of power. So we have to thank Richard for doing what he did. And for not going to be a waiter.

It is better to be a waiter than to develop non-free software. I agree with him on this one. But developing Free Software is better than to be a waiter.

Happy Hacking!

Microsoft Windows Is Free Software!

No it's not. But...

`lbry://@blenderdumbass:f/Microsoft-Windows-Is-Free-Software:7`

You are probably very confused with the title of this article. How could proprietary Microsoft Windows be Free Software? This doesn't make any sense. But let me go over how and why it makes sense even tho it's not really the case for you.

Good Morning - Good Evening

When people say hi to me in person. I usually respond with a variation of "Good Morning and Good Evening". Some people think that I'm just quirky. But few ask. What is the point? It's either evening or morning. Why would you say both?

On my screen I have multiple clocks. One shows my current time. It's 6:02 PM or 18:02 depending of how you read the time. Evening. In the US central time zone. Where some of people I know live. It's now 10:02 in the morning. It's both. In the same time. Morning and Evening. Depending on where you are in the world.

Mentioning both Morning and Evening when I say hi to people. Makes me not forger those people on the other side of the globe.

What is Free Software?

The Free Software is defined as software that gives it's users 4 essential Freedoms.

1. To run the program when ever you want for what ever purpose.
2. To read the source code and to write changes.
3. To redistribute. Give other people copies of the program.
4. To distribute modified versions of the program.

Those 4 essential freedoms are essential. Meaning if either one of them is missing, the freedom of the user of the software is no longer there. The software is not Free.

Can you run Windows when ever you want, for what ever purpose? I don't know. I don't think it's the case. Can you read it's source code? Not really. Maybe a few little things that come with it that they borrowed from other Free Software. But not the whole code. Can you redistribute copies of Windows? No. Well only if you certified with a license to do so. And it's not something anyone can do. Can you change it and redistribute copies of changed software? Not really. Perhaps they can give you this ability if it's very wanted.

So it's not Free Software. How do I claim than, that it is?

Premo

Premo is an animation software used in Dream Works Animation Studio.

It's not Autodesk Maya. It's not 3D Max. It's not Cinema4D and it's not even Blender. It's something they call "proprietary software". Which is somewhat true. But it doesn't mean that Autodesk Maya is not proprietary all of a sudden.

Autodesk Maya is proprietary since when you get a copy of it. You don't have the 4 essential freedoms. Premo is what they call "proprietary" since you don't even get a copy of it. It's custom built to be used only in Dream Works Animation Studio.

But why? Why would they build something from the ground up if they can use other software? They have enough money to pay for Maya or 3D max. And they can use Blender. Why would they develop something custom?

From the perceptive of the Free Software movement Premo is fine. Since you are free not to give a copy of your program if you don't want to. The problem starts if you willing to give a copy. Without the 4 freedoms.

The thing is with Premo is that the developers of the program are working together with the animators.

> One of the advantages we have at DreamWorks in building "proprietary" software is we sit literally on the same floor as the animators. We work very closely with them and try to build exactly what they’re looking for.

I don't necessarily agree with their use of "proprietary". I think what they mean is "custom". Since they don't release it to the public. It's only for them. Their own, custom Software.

Making the software by your self. Like in the case of Premo. Gives you, the developer, full control over it. And in the same time. Give you the 4 essential freedoms. Since Premo was developed by DreamWorks for DreamWorks. It's Free Software to DreamWorks. And if they decide to release it. It will be Free Software if they give the 4 freedoms with it. And it will be proprietary if they don't. But for them it's always going to be free.

Windows is Free Software for Bill Gates (or who ever holds the copyright)

Since Bill Gates owns the Microsoft Corporation. He controls the software they produce. And for him Microsoft Windows. Or anything else by Microsoft is Free Software. For the rest of the world it's not. Similar to how in one place is now evening. But for somebody else it's now morning.

Happy Hacking!

What If At Least 1 Of The 4 Essential Freedoms Was Missing?

Let’s examine why so many people insist on using only Free Software.

lbry://@blenderdumbass:f/what-if-at-least-1-of-4-essential-freedoms-was-missing:e

With Free Software every user gets the 4 essential Freedoms. If at least one of them is missing. It's not Free Software anymore.

In the spirit of the "what if" article. I want to ask, and speculate. What if at least one of those essential Freedoms is lost? And why it can't be Free Software without it.

Freedom 0

The Freedom Zeroth is when a user has a freedom to run the software for what ever purpose at what ever time.

This seems like a kind of obvious Freedom. Of course you will have a freedom like this. If you get a copy of any software. You will be able to run it at any time. And do with it what ever you want. But mentioning this freedom is important non the less.

It was very clever to put Freedom Zero at the Zeroth place. From one side it makes this a joke that programmers get. Most programming languages index start at zero. So it's natural for a programmer to count from zero.

The other clever thing is to hint. On both how essential this freedom is. But also how important it is still to mention. It's so essential. That it's at a zeroth place. It's not even number one. It's something that has to be just understood. But yet unfortunately it has to be explained. Because proprietary software developers really want to take even this Freedom from you.

What if you don't have it? Some software have, what they call "free trials". You get a copy of the software gratis and it runs for a limited amount of time. And after that time is passed. The program no longer runs. Eliminating the zeroth freedom.

Other software include clauses on use of the software. Such as in Autodesk Maya. You can get the same software for two different prices. But with one, you will be prohibited from making commercial art with it. Eliminating the zeroth freedom.

Other types of such abuse could be political. When a license includes a clause that a person which disagrees with the developer on some subject in not allowed to use the software. Or when the use of software is not furthering the views of the developer is not allowed. Or something along those lines.

Some of them will claim. That since Free Software is widely using the GNU GPL license. With a clause to restrict proprietaryzation of the software. Employing a CopyLeft maneuver. That will insure that the software will be free no matter what. Some of the developers claim that making software that restricts certain use of it is okay too. Calling their software unfortunately "ethical software".

From GNU.ORG List of Licenses:

> **The Hippocratic License 1.1**
>
> This is not a free software license, because it restricts what jobs users can use the software for. That denies freedom 0. This entry was previously listed as the First Do No Harm license.

People are always wrong. The only thing you can try to do is to be less wrong. Restricting usage based on what you believe is like not allowing for thoughts to develop. Just because you had your freedom of thought. And you now had a conclusion. Doesn't mean that your conclusion is final. Since you are wrong. As I am too. And we will get closer to truth eventually. But never reach it.

If there would not be freedom zeroth. There would not be freedom of speech. There would not be freedom of thought. There would not be freedom.

Freedom 1

The Freedom first states that the user should have the ability to study the source code of the program and modify it. Making the program do the computation of the user the way he wishes to do it.

In Free Software versus non-free software. You can see a trend of customizability. On Windows 7 I remember was only one theme available. You could change the color of the theme. And that's it. I don't know about Windows 10. I haven't touched it.

On GNU / Linux instead you have various Desktop Environments. And with ones like KDE it's so customizable that comparing it to Windows is pointless. I feel in control even tho I didn't look at the code yet.

Free Software also requires the Source Code to be available. And a permission to edit the source code should be there. A permission to run modified copies of the software should be there. So if the developer did not implement something in the UI. Or in the settings. It could be still customizable through the source code.

You don't like something, you remove it. You miss something, you add it.

If this Freedom doesn't exist. Any developer can force any feature on a user. Whether harmless of malicious. And of course without the source code. The user can't know if there is malicious features. Can't remove them if he doesn't like them. Can't customize things he wants to customize.

Imagine you bought a car and you couldn't change it's tire. Since the developer doesn't want you to fix you **own** car. Ridiculous.

Freedom 2

The Freedom 2 states that the user should be free to give or sell copies of the software further to other users. If he wants to. He is not obligated to do so.

Software embodies knowledge. The source code is a written form of such knowledge. A knowledge of how to solve a particular task using some steps. An algorithm.

Imagine what it would feel like. If a parent could not teach a child something he knows. Just because he is restricted by copyright, or something similar, from doing so.

Software passed from person to person is similar to a very detailed advice on a question of solving a computation. How do I do such and such? Here is a text document that will do it for you.

Math is a programming language. A math expression is an algorithm. It's knowledge written down. A program written in a programming language is just another form of such algorithm. Another format. Math expressions can be very big and complex. So are computer programs. So passing on a programs should be possible.

What if it's not? What if you can't redistribute software. Then you have information you are not allowed to tell.

Freedom 3

The third and final freedom states that every user should be free to give or sell copies of their modified version of the software. Of course if they want to.

This is similar to the freedom 2. But it extends a little bit more. What if you have the first 3 freedoms? But not the final third one? I know I just realized how confusing it is to start counting from zero.

A lot of people are not programmers. They will receive exact copies of something. And they would not know how to modify them. For example if there is software that has a flaw. And one user modified it. If he can't give you his fixed version. And only the original one. You will have to re-do the same job. But what if you can't.

Okay, you can give it to a programmer. To remove it for you. But then he can modify it for him self. And can't give you the modified version.

You are stuck.

Making him be able to give you the modified version. And not just give... Give or sell. Enables non-programmers to benefit from Free Software too. One can modify it. And other people can use the modified version. And if you want to modify it. But you don't know how. You can hire a person that will make it for a fee.

Free software is not about Gratis. Is about Freedom.

Happy Hacking!

Is Elon Musk Stupid?

Elon Musk in not necessarily stupid, but he could as well be just very lucky. I’m confused...

`lbry://@blenderdumbass:f/Is-Elon-Musk-Stupid:3`

Elon Reeve Musk is in the top Richest people on this earth. And probably soon beyond Earth. It would be a mistake to say that he is **the** richest person. Since he and Jeff Bezos are in the race for this title. And the lap is far from over.

Elon Musk is a very technologically driver person. And since I talk about Free Software. I have thoughts about Elon.

Tesla

Tesla is a company that produces cars. Electric cars. The idea of the company was to promote the use of electricity in cars. Instead of the regular fossil fuels.

The company was founded not by Elon. It was founded by Martin Eberhard and Marc Tarpenning. Their idea of a company was to be both car manufacturer and a technology company.

Elon liked the electric car concept so much that he joined the company. And became one of the directors. After a while he became the main voice of the company.

Elon said quote:

> We will not stop until every car on the road is electric.

Cementing his view on the business. His company is not about making cars and selling them. It's not similar to what Ford or General Motors would do. His idea is more of a "concept that he wants to pursue".

This is why Elon Musk doesn't care about patents. It's not about making Tesla sell more. It's about making every car in the world electric. Even if not by Tesla.

Tesla publicizes their developments. Sharing details about the implementations and reasons behind things. The software of the car is Free Software. It could be found on their GitHub page..

Tho it seems like they didn't care about software at first. And had to be reminded to upload the source *code by the Software Freedom Conservancy. Non the less the source code is available.*

It doesn't seem like Elon Musk care much about Free Software. He uses an iMonster. Tesla cars could be bad for privacy. But his views are very good to begin with. He is just probably not very educated about software and privacy.

Neuralink

One of his most prevalent concerns, is the one about artificial intelligence. Artificial Intelligence is an interesting computer science concept. It's about computers learning by them selves. Not when people program them.

AI is software. From the perspective of Free Software. It's fine to have AI. Since you can manipulate it's algorithm by giving him different set of initial data. It's a bit more complicated then editing a source file. But could be done.

Of course computer people and smart people are not very in favor of AI. Because it has one extreme flaw.

There was a movie by Martin Scorsese called Cape Fear. In which the protagonist is a lawyer. But the antagonist is a very deranged criminal. But that knows the law, as good, or even better then the protagonist.

The premise of the film is to show that if you are a bad person. You will find how to use the law to your advantage. To do bad things. It will not stop you from making bad things. If you know the law well enough. It will give you Power against the Law Enforcement.

The antagonist would play by the rules. But just right on the edge of where he can still get away with it. Making him the most terrifying antagonist ever.

AI is the same kind of antagonist. How ever well you define the rules. It will ignore the intentions behind them. And be always right on the edge. Just so it would always pass the rule-obedience test.

You tell it to bring you coffee. It might break the whole house in the process. There was no rule, not to break the house. You write it a rule to not break the house. It will kill those who want to move things.

Since they might break those things. And it's breaking the house.

The AI is not a bad antagonist. It's a system that tries to do a task with maximum perfection. Ignoring anything else in the process. It's like a very obsessed person. A stalker. But without emotional core. AI is fucking terrifying.

Elon Musk's answer to AI is not very good either. His idea is to implant chips into peoples brains to speed up interfaces with devices. In theory it will make people think faster than AI. This could be something that would work. But it has various implications.

Privacy will disappear completely for people on the Neuralink. And for people interested to those on Neuralink. Since if you speed up your brain that much. You become freaking Lucy. And you know how people beside Lucy are not able to do anything.

Also if you are on the Neuralink. Your brain is crackable. People with malicious intentions will no longer target phones and computers. They will target brains.

Richard Stallman quite often says when people ask him about innovation, that innovation could be either bad or good. That Democracy was ones an innovation. And Tyranny was ones an innovation. So thinking only in terms of innovation is wrong.

We need to have a set of values. Like Freedom. And choose innovations by that set. Rejecting other types of innovations. SaaSS is an innovation that strips people from their computational Freedom.

AI is a dangerous innovation. Some things should be done to restrict it. Neuralink is a dangerous innovation too. Yes. It's an attempt to stop the AI danger from becoming too dangerous. But you create the same situation as in Civil War or Batman vs Superman. It's super-beings fighting super-beings. And killing regular people in the process.

He probably thinks that everybody will just become super-beings. But what if I don't want an implant in my brain? What if I don't want to use SaaSS? Well then I'm a background character inside a falling building.

SpaceX

Elon Musk's desire to leave earth is understandable. Sometimes I want to leave it too. But funny how hypocritical he becomes when talking about SpaceX compared to talking about Tesla.

In the same article I linked above. Here it is again. It mentions a quote by Elon that says:

> We have essentially no patents. Our primary long-term competition is China. If we published patents, it would be farcical, because the Chinese would just use them as a recipe book.

His desire on making all cars electric suddenly stops when it comes to space flight. He doesn't want everybody to go to Mars. He want SpaceX to do that.

It's funny how in Elon Musk lives both Richard Stallman (Tesla) and Linus Torvalds (SpaceX). While Richard wants all software to be Free Software. Meeting with Microsoft to persuade them to release Windows under GPL. It's similar to how Musk wants that all cars to be electric.

Linus Torvalds by this point. Probably want everything to be based on Linux.

The Boring Company

We can't ignore the best thing I love in both Richard Stallman and Elon Musk. It's their love for jokes. Richard Stallman has a page with his own puns. And even made a joke religion. *Funny how I type this article in Emacs.*

Elon Musk is probably well aware that the most boring company that he ever made is the one focusing on making holes in the ground. So the pun of the boring company is well intended.

In my opinion is wrong to call attention of puns. Let the reactions to be natural. Don't apologize for a pun. Let it just be there.

One more example of Elon Musk extreme sense of hilariousness comes in a form of **X ÆA-12**. This is a name of his son. I don't even know what is it going to be, to live with this name. **Wait!**

Is Elon Musk stupid?

What if he actually is? And he was just very lucky to make things work. He wanted to go to Mars. So he launched a rocket company. He just wanted to make something from the science fiction books to become reality.

He wanted another thing from the science fiction to become reality. Electric cars. And he wanted it everywhere. So Tesla was a good company to join.

He wants us to be cyborgs. And he was terrified from Hall 9000 from 2001: A Space Odyssey. So he made Neuralink.

What if all those decisions, all of those companies, were just done to make our world resemble more of a science fiction world? What if all of his reasons are just excuses to do the things he does? This may explain everything.

Elon Musk shows the power of an idea. And that with the right set of stubbornness, no matter how stupid you are, you can do anything your heart desires.

Can anybody please give a Richard Stallman type talk to Elon Musk? This would be great to have him on board.

Happy Hacking!

What If Everything Was Nodes?

Nodes – User Interface technique of drawing little boxes with settings, connected together with lines, to create a more powerful and flexible way of setting up things.

lbry://@blenderdumbass:f/What-If-Everything-Was-Nodes:e

I'm Blender Dumbass. Originally I was making tutorials about Blender. But recently, since I don't have a very good computer to make tutorials. I decided to try out myself in writing. So that's why a channel called Blender Dumbass is posting text articles.

You know those people who promote GNU / Linux and Free Software in general? Sometimes only promoting "Linux". And trying to convince people to use it because, steam made games work.

I used to do similar but with Blender. I didn't much care what Operating System people used. As long as they don't use other software. Until I became a "Linux" guy. And now a GNU / Linux, Free Software guy.

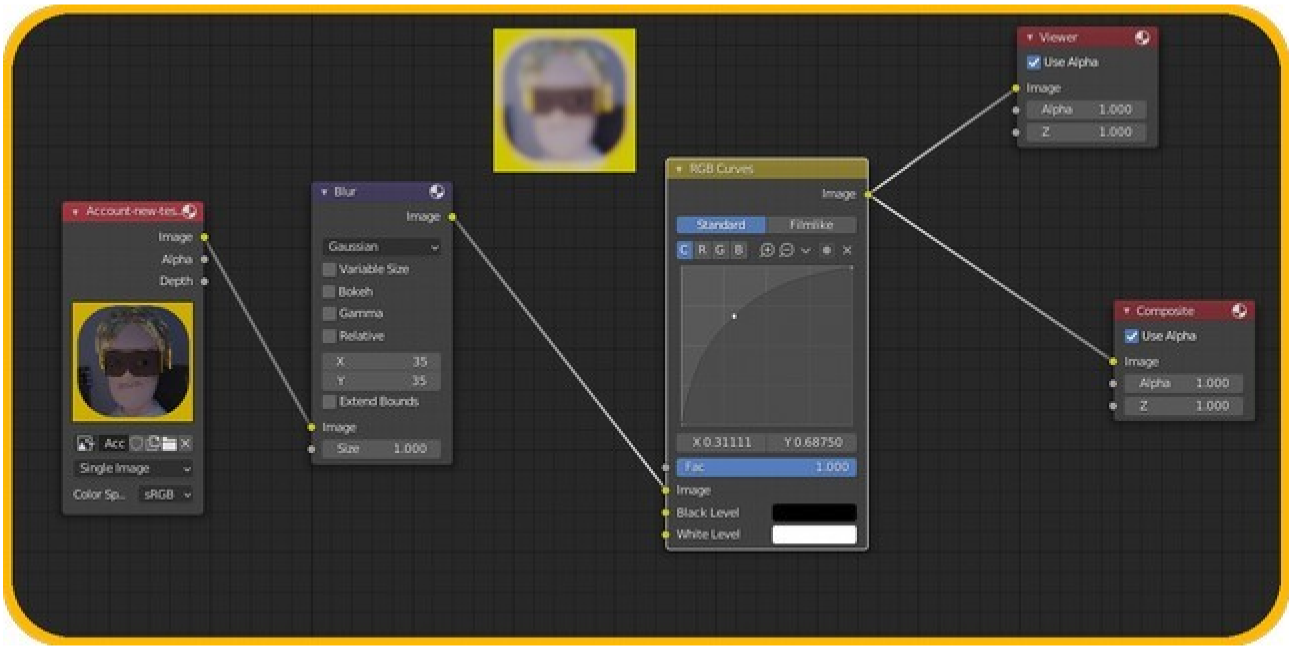
Blender is what started the journey. But what is the journey that Blender takes? What is it's future?

Everything nodes

In 2006 Blender Foundation tried something brilliant that more software companies should do. They made a movie using their own software. Trying doing actual work on it. It revealed issues. And made them develop the software in a more user-friendly way.

One of the issues that they found was a lack of post processing capability. They needed a compositor to add glow effects and touch up the colors after the rendering of the image is finished.

Some UI design was needed to make this compositor work. And the decision was made to implement a Node Editor.



This is an example of a modern Blender's compositor. On the left there you can see an input image. The logo of this channel.

On the right you can see the 2 output nodes. The Composite node. That will be our final image when we press render. And the Viewer node. That will draw what ever connected to it on the background. As you can see it gives us the blurred version of the channel logo in the top center.

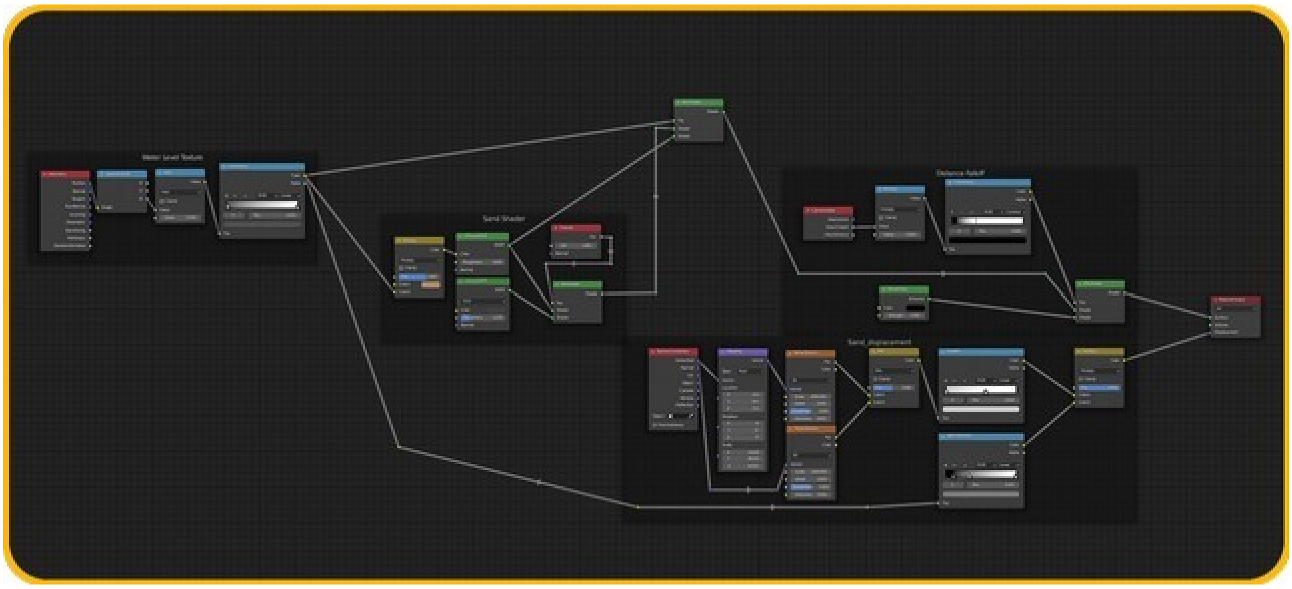
In the middle you may connect what ever and go nuts. As you can see here I connected only 2 nodes. The Blur node. And the RGB Curves that adds a little bit of brightness.

You can see that some nodes have multiple inputs. Meaning that with enough creativity you can do almost what ever you want here.

Later, in 2011, Cycles happened. A new render engine for blender. With a new algorithm of path tracing. Which is calculating the paths and bounces of light rays. And not just approximating everything. But one of the coolest features of the new engine, was the shader node editor.

The old Blender Internal Engine had a node editor for materials. You could use it. But it was a hack. It would be a post processing thing. Not something that could effect the whole scene.

With Cycles you control the properties of the material with a node editor, by default. And since it's calculating the bounces of rays. It effects the whole scene.

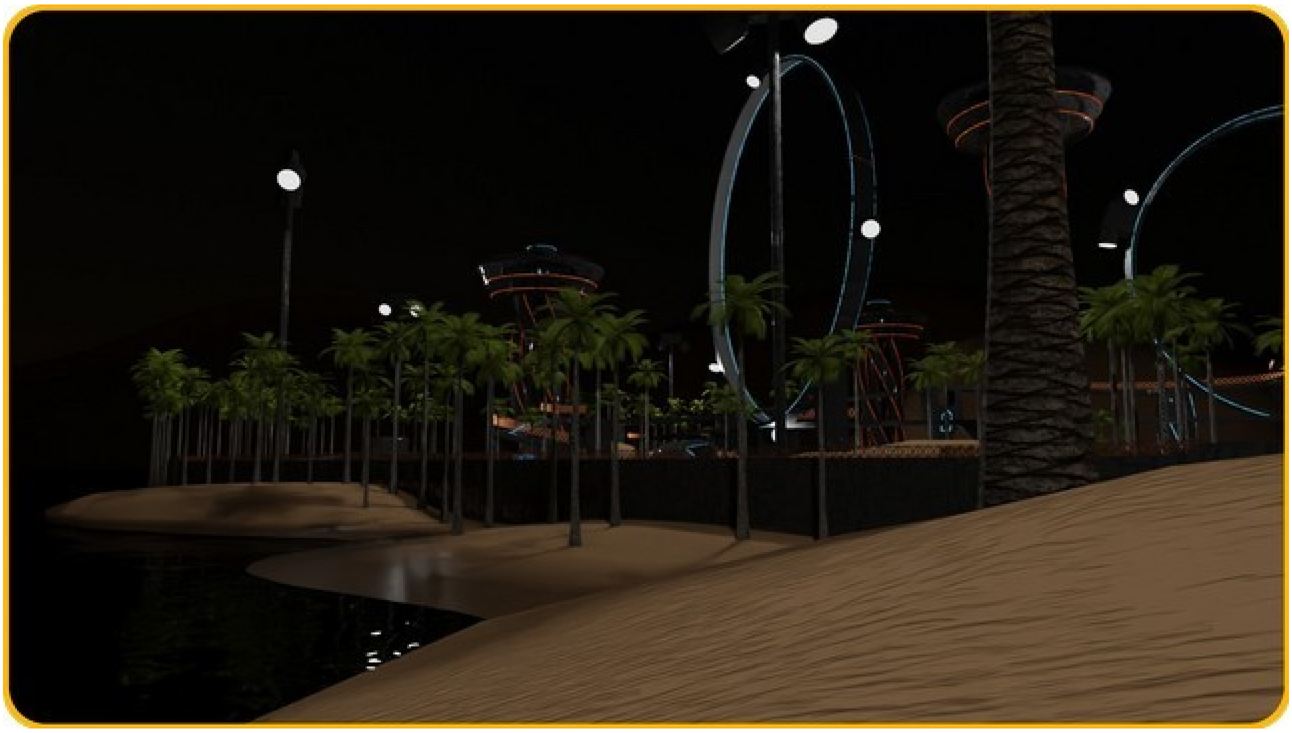


This is a material of sand used in my little movie project Moria's Race. As you can see it's far more complex than the previous example.

This time we have features like procedural bump texture, that displaces the light direction, distorting the normals. Making it look wavy like the sand.

Under a specific altitude in the scene, the material will become darker and more glossy. Simulating wetness from the water.

All of those things are made using various inputs and various function nodes that manipulate the data in some way. To create a desired material.

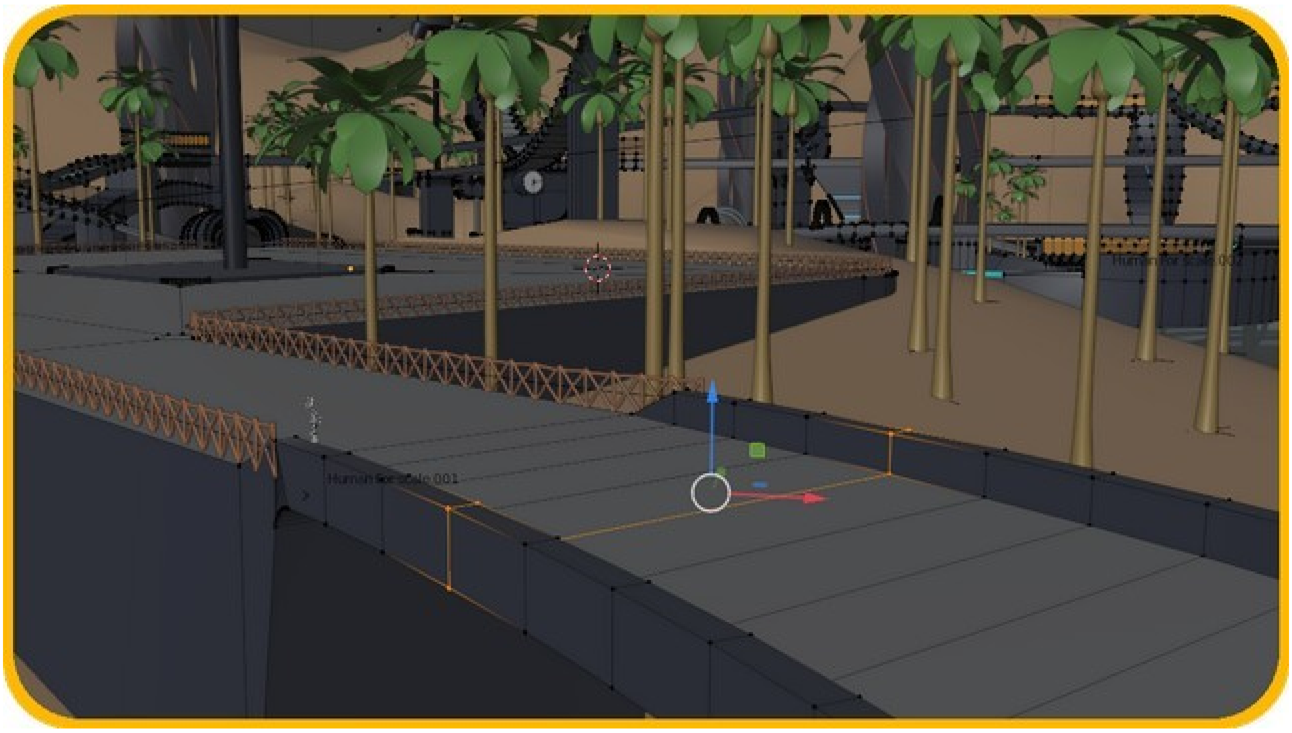


This is how this material looks in action. As you can see there is a texture when you are close to it. That blurs into a solid color as go further away. The part the touches the water is dark and shinny. Looks like it's wet.

Similar things would be done to all objects in the shot like this. The palm trees use image textures for their look. The glow things use Emission Shaders that cast light on other objects.

If you have imagination. Get Blender and start creating.

Lately Blender Developers decided that the natural way forward is to introduce a node editor to basically everything. Calling this huge project Everything Nodes. The focus at the moment is on the Geometry Nodes.



This is polygonal modeling. It's when you put points into 3D space. And connect them with lines called Edges. Between those edges you can define a face. Or a polygon. Those polygons are essentially straight sheets. But with enough of them you can create an appearance of smoothness. Also there are algorithms that may help to make things look smooth without too many polygons.

With Geometry Nodes it's something else. It's all about manipulating geometry is some way using the node editor. And if you have a lot of imagination, and you are willing to experiment, you can make very cool stuff with it.

They are already planning the Nodes for simulations. Which will be it's own huge project. Then slowly but surely you could do everything in Blender using Nodes.

I want to look at... **What If Everything Would Be Nodes? Even if it's beyond Blender.**

Writing text

I use Emacs currently to type this article. I like it. I love the man who made it originally. I love the joke religion around it. But **what if** there was a Node Based text editor?

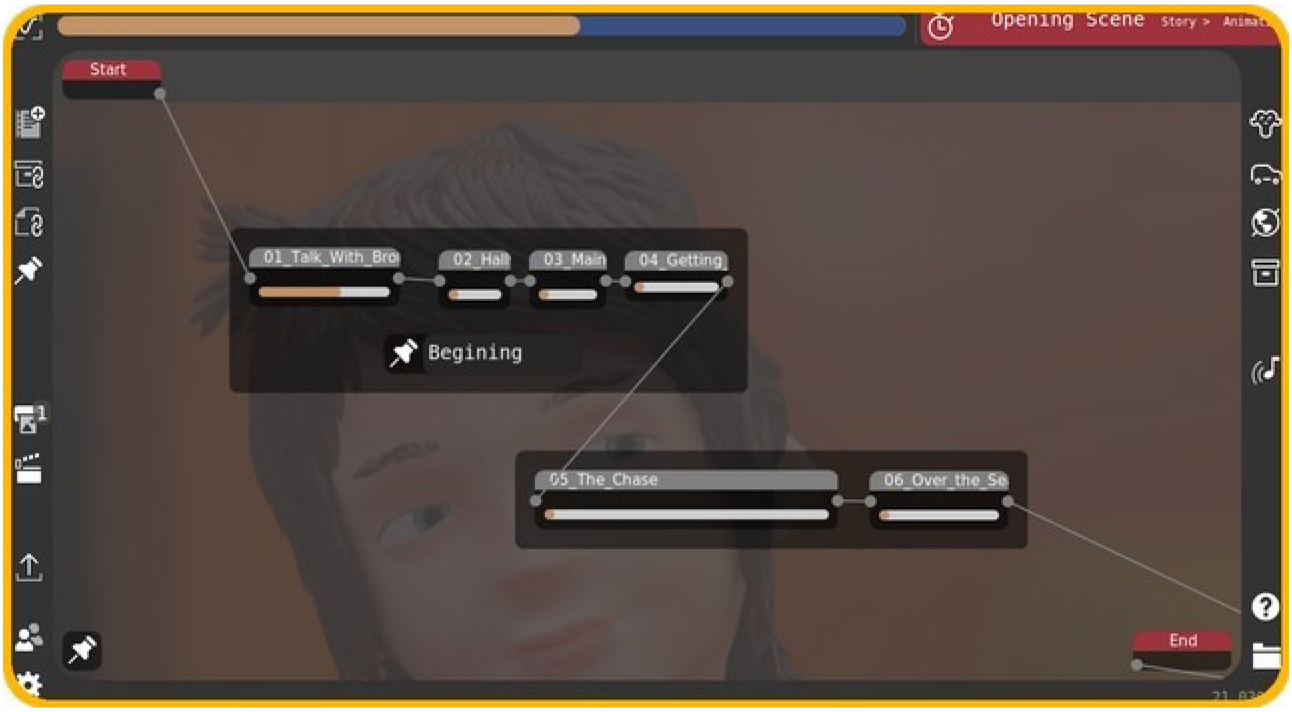
You may say that it makes no sense. But think about it. This article has a tiny opening at the beginning. And a conclusion chapter. From one to another I push you through various ideas and discussions.

It could be implemented as a node editor. Where each chapter is a node. The input node will be the first chapter. And the output node will be the last chapter.

Some screen writers use a technique where they write scenes on cards. And then shuffle them around to see what flows better. They might take out a scene. Or add one. Depending on their imagination. Or what feels right to the guts.

It would sound like an advertisement. But there is such an editor. And I made it. It's called VCStudio. Don't worry it's Free Software. But don't be too hipped. It's in a very early alpha. And probably will irritate you a bit if you are not me.

There is an issues page. If you will want to try it out. Let me know about any annoyances. I will try to fix them.



This is what I call Story Editor. A Node Editor. But where you plot a story.

It's not a functional node editor. It's not for making a shader. Or compositing an image. It's more to keep track of what scene goes where. And to be able to store scenes that you want to remove. Just by disconnecting them from the chain.

The typing is still typing. You double click on one of the scenes. And you get into a normal text editor for this scene.

Of course the program does more then this. It's an asset manager. And a scheduling system. Basically taking a lesson from Blender that has everything you need to make a movie. VCStudio is everything you need to organize a movie. Or with enough imagination, any project what so ever.

Enough praising my own ego. Let's talk business. This is not really a Node Editor to write texts. It's a node editor to organize chapters. Would it be even practical to write text with a node editor?

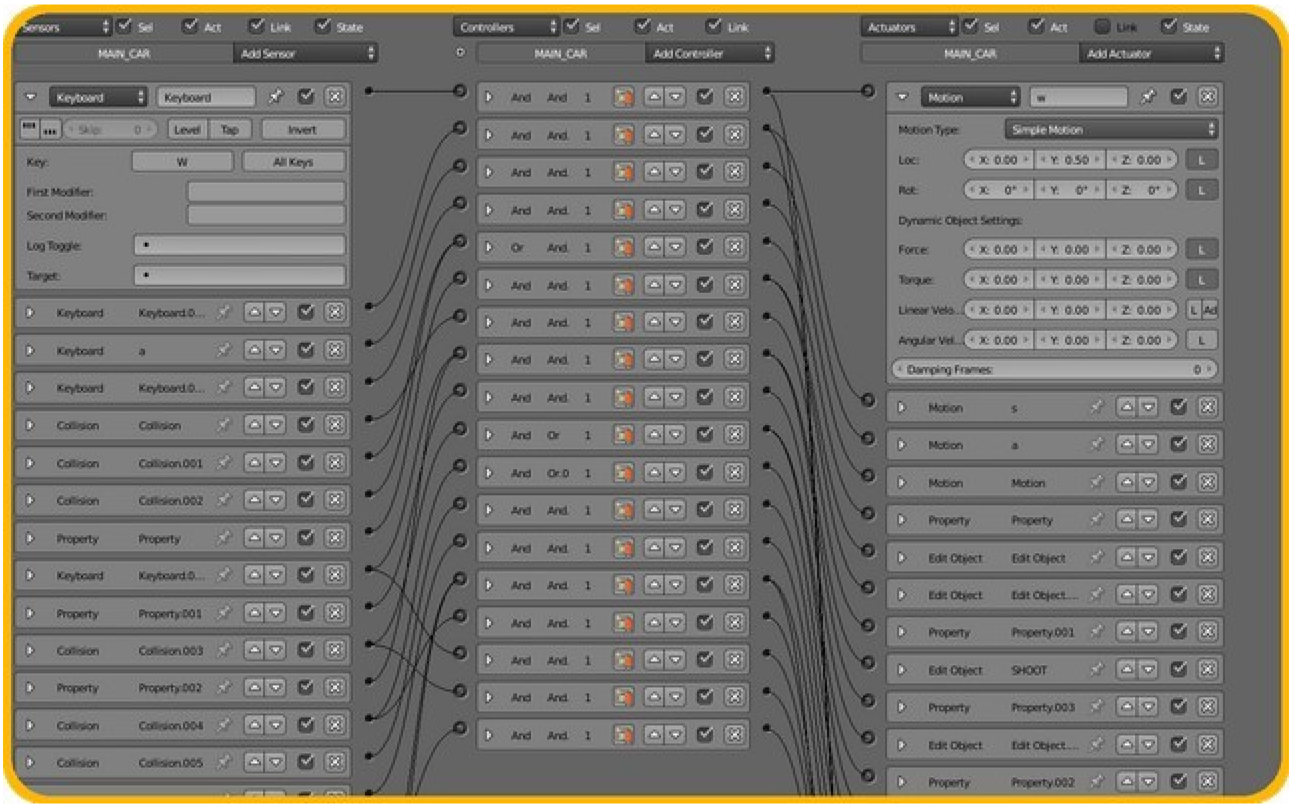
Would it be connecting words? Or making meaning graphs? That some algorithm will translate into an essay. I don't know. Perhaps the comment section will help you out.

Building Software

If you really think about the Node Editor of Blender. It's basically an interface to build tiny algorithms. Such an interface could be used to build actual programs. In fact it's already being used for this purpose.

A lot of game engines use Nodes or something similar to make the logic of the game. The Developers of the game engines figured that people that make games are more on the creative side. And learning coding to make a simple thing react to something that the player does, is not very productive.

I can't tell you who was the first to implement it. Probably Unreal Engine 4. Or Perhaps Blender Game Engine. Since even tho it's not a full node editor. It's very resembling one.



This is Blender Game Engine's Logic tree from my game J.U.M.P Limited. Stop promoting yourself, you bloody bastard...

It looks like a Node Editor. But it's not quite. You have sensors on one side that activate things on the other side. It's clever in it's own right. Bummer that it's no longer in the official fork of Blender.

If you still want to use it. There is a fork called UPBGE that keeps the Game Engine while implementing all the good new stuff. Like the EEVEE engine that takes the Cycles nodes, but makes it all run in real time. Here is a power of Free Software for you.

I don't want to talk about the editor of Unreal Engine 4 since, well... I don't know much about it. And it's a non-free, proprietary software. Even tho they uploaded the source code. The license is too restrictive. And not everybody is authorised to even look at the code.

So instead I gonna link you a different game engine that uses real nodes for logic. And it's Armory Engine.

Music and Sound

If I gonna brag about everything I do in this article. How can we not talk about the Music. While I make films. I need to compose songs for them. So I learned this. Out of necessity.

Some of the things I do quite often while recording various things. Is setting up JACK and various effects in Rakarrack. And if you seen the interfaces that come with these programs. It's quite similar to node editors.

I think Rakarrack can be implemented as a node editor. Since it's a pedal simulator. And if you seen real effect pedals.

This is basically node setups, but in hardware.

For melodies I use Rosegarden, if I'm not playing something for real. It's not a node editor. It's a note editor.

Conclusion

Not everything makes sense as a node editor. But some thing do and welcome. Sometimes simple interfaces are too limited. And a node editor can help breaking those limitations. And unleashing the power of imagination.

Happy Hacking!

VPN doesn't work !

VPN – Virtual Private Network. It’s when connection to the internet is routed through a separate node, with which the target website will not see your IP address, but an IP address of that node.

`lbry://@blenderdumbass:f/VPN-doesnt-work:d`

One of the things that Evil Tube's "monetization" did to authors of videos. Since Evil Tube cuts so much and pays so little anyway. A lot of authors on that platform need to get money some other way. Usually in a form of ads built into the video it self.

They do sync their videos here on Odysee. And even if not. There are ways to watch those videos while keeping your personal freedom. So I do see some of them sometimes.

Odysee is so much more advanced in "monetization". Bryan Lunduke said that his channel on Odysee made enough money on it's own. While YouTube was paying so little that he needed to advertise various companies like System76.

But most channels enabled the sync long time ago and completely forgot about the existence of their channel here. They act as if Odysee or LBRY never existed. Keeping sponsor segments. Saying to Like and Subscribe. Ignoring the obvious Support button in the center under each publication.

If the videos make thousands of LBCs and nobody withdrawn those LBCs for a long time, you can be sure they forgot about Odysee. They could potentially even forgot the password. A notable example of this is Veritasium.

It's a big channel on Evil Tube. But it's also one of the largest channels on Odysee. But still they have videos dating October 9th with a thousand LBC just begging to be taken. But nobody seems to care.

They could move their entire existence to Odysee and forget about the sponsor segments. But since they never cared about the platform to begin with, when you watch their videos, you feel like you are not supposed to see it. Since they do it as if you, or Odysee never existed.

They make a lot of sponsor segments. Some are good. Like Kiwico. Some are questionable like Raycon earbuds. But some are outright cringe like Nord VPN or Express VPN.

Privacy is not that simple

It's a fair point to advertise that a VPN system can be used to make a server think you are in a different location. And with it allowing you access to publications otherwise inaccessible. This is an argument I see a lot. And it's a good argument.

But more often, you hear them begin the sponsor segment from words like:

> There are a lot of hackers trying to get your personal information online.

And then talking about the "military grade" encryption used in those VPN services that they promote.

Internet is heavy on encryption anyway. If URL starts with https the connection to that site is encrypted. Crackers (Hackers that specialize on breaking security) know that. They will not even attempt most of the time to see what the computer is sending over the network. Since it's no longer viable. Too much is encrypted with or without a VPN.

But people's accounts and passwords are still being cracked regardless of that. And that's because they attack thing out your control. Like for example. Servers.

There is no cloud. Only other people's computers.

There are various ways to crack a badly designed website. Using techniques like SQL Injection a cracker can make the server execute code that it was not supposed to execute. Revealing potential sensitive data.

If you are using a VPN to upload the data to this web site. Well this doesn't help you. Since the web site is cracked. Not the connection.

During such a crack, a cracker can be lucky to get email addresses and passwords of accounts on the site. Sometimes they will be stored in hashes. But not always. And even if it's hashes. It's just a matter of time until a cracker knows the passwords anyway.

A hash is just a way of turning one information into a different information. But in such a way that it will be hard to reverse. When you type your password, it's turned into a hash and compared to the hash on the server. And grants you access if the hash is the same.

If a cracker has hash of your password. He has something to check the password against. And since he is not randomly tries to log into your account. He has as many tries as he would like to have. With some automating software and few clever tricks, like trying often used password combinations first. He can get to passwords of people in a very short period of time.

And then if a person is stupid and uses the same password everywhere. It's easy for a cracker to gain access to anything from that point. No VPN will help you against it.

But, you say. If I'm careful, isn't VPN saves my IP address?

TOR is better

TOR has a lot of Reputation problems these days. From association with criminal activity and Dark Web. To attempts of "redeeming" them selves by Screaming at Richard Stallman. Or hiring only minorities.

It's understandable that if you give people so much freedom, some times people are not very responsible with it. And then everybody blame you for giving them the freedom to begin with. Maybe an idea for an article for another time.

Anyway. Tor is good in terms of a VPN solution. I rarely use the Tor browser. Sometimes to watch Odysee streams. For some unknown reason it works better in the Tor Browser.

For my regular browsing I use GNU IceCat with Tor enabled. But in IceCat it's an optional thing.

If read an article somewhere. Why would they care about where I am? Some sites tho, are very bad at this. And if they see that you are connected over Tor. They don't allow you in.

Tor and VPN are similar in concept. Connect via a node somewhere in the world. Encrypt the data transmitted, just in case. And the web sites server thinks you are a person in Peru, while you are actually from Ukraine.

Tor works similarly. It's about rerouting the connection through some other computers. But instead of 1, it's at least 3. Sometimes it's even 6. Depending what kind of a web site you are accessing.

One more thing that has to be addressed is that with VPN it's usually companies own computers. Meaning they control each node. And if the company wants to track you, they totally can do that.

With Tor each node is setup by a volunteer. Meaning that you have extra security against the Tor it self. And since it's 3 nodes. You have extra security against any of the volunteers.

Even Tor is not enough

Some people connect to Tor or a VPN and then login to their Facebook. Facebook knows who you are and where you are already, anyway. So Tor will not help you with it.

In order for Facebook not to know who you are, you should not have a Facebook account. Whether you use Tor or not, doesn't matter much.

But there are various other ways to figure out who you are and where you are. Even if you are anonymous.

Chatting with people on the Tor network. Even anonymously. Without telling your name. It's something not advised if you want to stay private.

People who want to find you, will use dirty tricks like asking you broad information about yourself. That you may not realize will help them to know exactly who you are and where you are.

Or for example they can analyze the way you type. The mistakes you make frequently. And patterns of words that are yours specific. And then search for those in other places. Using a normal search engine they might find you on some forum, with your real name in plain text. Or at some web site which is easy to crack.

In order to stay Private on Tor. You need to turn on the strongest protection, turn off any java-script. And just observe. Do not interact with anything. After you see what you saw. Forget, burn, erase. So if you will be asked later. You would not remember what you saw.

Conclusion

Free Software is the only hope for security. It's not guarantees security. But without the 4 Freedoms you don't even have a chance of security. There is no security in proprietary software against it's proprietor. So Tor, being Free Software, is better then any proprietary VPN.

VPN works if you want to see a video that is not allowed in your country. But to be secure. You need more then that. You need a good amount of paranoia and cleverness to even start going towards security.

But if I want to do things online? Like publish this text of mine? I'm not private anymore. This is why Freedom is important. I can publish. Or can decide not to publish it. I can reveal my information if I want you. **If I want to**. Not because some dis-service requires it.

Happy Hacking!

Movies Under GNU GPL License?

GNU GPL is a very strong copyleft license which requires publishing the "preferred form of the work to make modifications to it". In software it's easy, you publish the source code. But what if it's something else?

lbry://@blenderdumbass:f/movies-under-GNU-GPL:1

What I do quite often in these articles of mine, is I talk about Free Software. Many of you are familiar with this term already. But for who are not:

Free Software is **not** about Price. It's not about being gratis. It's about Freedom. Most people call software like this "Open Source". But having the source code publicly available is just a half of the picture.

This is why me and a lot of other people prefer the term Free Software. Because people who use software, should be able to control it. Users should be **free** to decide on what the software is doing.

Software is Free when it has the 4 essential freedoms. To run when ever, for what ever purpose. To study and modify it, for which you need the source code. To give or sell copies. And to give or sell copies that you modified.

Sometimes in order to avoid the ambiguous term "Free" we borrow a word from a different language that doesn't have that issue. So example the word "Libre". And we call Free Software. Software Libre.

People ask Richard Stallman frequently about what if Libre could be something else rather then just software. I want to take a look on one such concept.

What if there were Movies Libre?

The GNU GPL (CopyLeft)

Before we dive into the Movie speculation zone. First I want to talk about the GNU GPL license. And remove some ambiguity about it.

I guess, since you are reading this, you have heard about the GNU GPL. But I'm not sure what level you are in understanding it.

With the copyright system. Simply giving people the source code is not enough. You need to give them also a permission to do something with this source code.

You can personally tell people "I allow you to use it for this and that." But a better idea would be to give them a document. Something they can show in a court of law. That states how they can use it. And on what conditions.

This document is called License. The copyright holder always can do what ever he wants with the software. The license is there only for those who are not the copyright holder. The copyright holder is choosing the words in a license. And can if he wants to, to license a given peace of software under multiple licenses.

Some Software Libre licenses are very permissive. Like saying "Do with it what ever you want, I don't care.". But with this comes an inherent flaw.

If we want software to be libre. We should not allow those who want to take it and run with it, to modify the software slightly. Enough for the copyright law to be applicable. And then stripping away all freedoms from their copy.

Basically taking Free Software and turning it into Proprietary.

For this a CopyLeft maneuver was invented.

Think about the software license as software source code. And the ability to turn your fork into proprietary is a bug. How do we fix it? We specify that you **can not** turn it into proprietary.

If you use the code of the software, modify it. And decide to release it. (You may use it for your personal needs, who cares.) If you release it, you should do that under the same terms as in this license.

This prevents Microsoft from taking Emacs and making a proprietary NotePadCs from it. If they want to do this. Their NotePadCs should be free software too.

GNU GPL is advised to be used in all software. Apart from a few strategic ones. Where a more permissive license could be used. Like with the OGG codec. Allowing Microsoft to built it into their proprietary media players, so people would be able to listen to OGG files.

Open Movies and Creative Commons

In 2006 somebody already beat me to the idea of Movie Libre. He chose a different name. Open Movie. I guess I, personally, have no objections against this name. But I know a person that would have.

This Open Movie idea came to Ton Roseendaal, the chairman of Blender.

They were doing a test for Blender. Trying to find out whether people can work with the software they made. And in the same time. Trying to improve it to the point of where it could be used.

The movie that they made is called Elephants Dream. Just to remind you. It's 2006. Made using only Free Software.

To raise funds for the film they sold DVD disks with files used to make the movie. All under a Creative Commons Attribution license.

This reminded Ton the way that Free Software operates. You have the Software. And the source code. So he called this kind of movie "Open Movie".

Problems with Open Movies compared to Free Software

When you get a copy of Free Software. You either get it with the source code. Or the source code is easy to get. No additional fees for the source code are required.

This is already one crazy difference between one and another.

Some Free Software sell the binary. Not the source code. It's always available. Either gratis on the side. Or in the package with the binaries.

Here with the Open Movies. You get the movie Gratis and you pay for the sources. I understand the reasons of why they did it. But it's not a very good analogy to Free Software.

Another thing is that they use CC-BY (permissive license) and not something like CC-BY-SA (copyleft license). Making it possible for a mere Attribution to fork the movie and make a proprietary movie from it.

Can we even make a movie under the GNU GPL?

First question is to answer. Whether anything but software can use this license?

And the answer is surprisingly - yes. In the text of the GNU GPL v3 it says quote:

> "The Program" refers to any copyrightable work licensed under this License.

Meaning. That anything what so ever, be it software, hardware designs, painting, movie, or anything else, could use the GNU GPL license.

Can we make the source code available?

> The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

So we have a clear definition of the words "source code". How do we apply it to movies?

I think in order to understand it we need to look at the process of movie making.

The Process Of Movie Making

The first kind of source work that you can get with a film. That might or might not be needed to modify the movie, is the screenplay.

Most films already have their screenplays publicly available. Some probably don't. Maybe since this is a source file. It would be required to be available to those who has the copy of the film.

I think even tho it's arguable. Since the movie could be modified without it. It's still good to have the screen play covered under the GNU GPL. Since it will make the same revolution that happened in software, also happen in films.

If a big, commercial film uses text from a movie under the GNU GPL. That big movie should release it's sources too.

Then there are props used during the shooting of the film. The actors. The cameras. And other things.

We can think of the actors as of programmers contributing to the movie. Since making them available would be an utter nightmare.

The cameras and computers used to make the film are just equipment. Similar to computers used to write Free Software.

But what about the props? What about the items in the film? The suits, the guns, the cars? Everything that they used, that you can see in the final picture... What about these peaces?

These things are not copy-able. Or at least not easily copy-able. Maybe a manufacturer can produce dozens of such things. But making sure that each person who has a copy get a copy of all props is more then unreasonable.

So I think we do something else. If a prop could be found, rented or bough the usual way. Meaning it's an ordinary object. That you can just get in a store somewhere. An object that wasn't designed specifically for a movie. Then it's not needed to provide it.

If it was designed for a movie. So the makers of the movie have the designs. If those designs are easy to copy. Meaning digital files. Or drawings, paintings. Not something like sculptures. Then these designs should be included with the sources of the film.

Then there is coverage. And other footage. All of these items are copy-able unless shot on physical film.

If it's digital media. Then it's an easy - yes. Just give it to us. But what if it's strips of film?

I think a reasonable thing would be to provide a digital copy of the images on the film. But then there is a question of, in what resolution. Since if the resolution is too low. There is not much modification could be done to the film.

I think that it should in a resolution not less then the average resolution for digital screening at the time of the release of the film.

I don't want to base it on the digital version of the film it self. Since it's not a good factor. And they might intentionally not release the film on digital. Or show it in a very poor quality. Thus taking our ability to reasonably modify it.

And then there is editing. And the files used to edit the film. And the various Visual Effect elements and 3D models.

Those should be all included since they are all easy to copy. But it's probably make you think about one possible problem with this.

Use of Proprietary Software in production

If you receive files from a Movie Libre like this. But the files require proprietary software to work with them. In my opinion is not good enough.

Similar problem happens in the world of Games today. Since they make the games in Game Engine Editors. And not via text documents.

Is it possible to call a Game, Game Libre if in order to modify it you have to use non-libre software?

I think the answer is - no. No matter what license comes with the game. But this is another territory of discussion.

Conclusion

Making Movies Libre is possible. It's just hard. With animation is a bit simpler. That's why we have Open Movies today. But to make a true Movie Libre is going to be a tiny bit harder.

Like for example. Where do I upload all this stuff? Oh wait. There is Odysee now. How cold I forget about it?

Happy Hacking!

Reputation of TOR

Compared to things like VPN Tor is protecting the end user way stronger. But since their protection is so good, various nasty people use Tor to cover up themselves as they do various nasty things.

lbry://@blenderdumbass:f/Reputation-Of-Tor:2

Tor is similar to VPN. But that insures security also from the Tor network it self by giving multiple parties control over Tor nodes. Anybody can start a Tor node. You connect through multiple of them at any time. Meaning that even if any one node owner would want to do something to you. You are protected by another 2.

This level of security made it possible, if you know what you doing, to become so anonymous, that a lot of illegal activity is happening over Tor too.

Selling of drugs, exchanging illegal publications, security cracking services and many more are hosted in such a way that only via Tor you can access those.

Probably you heard the term "Dark Web" used to talk about these sites. Tor has "enabled" those sites to exists and thus facing consequences. Tor's **reputation** is at stake.

Why do we need Tor?

I like the idea of GNU Net that wants to redesign the protocols that we use daily to "surf the web". To insure way more security, privacy and so on. Making the internet a better place to be.

One more idea / protocol that might work to replace the current internet infrastructure is LBRY. But it lacks few things that GNU Net solves.

One of the things GNU Net wants to do is to make every internet connection go through Tor by default. Making it impossible to track people's locations when they are using the web.

Also they gonna have encryption layers all over the place. Even before you send the signal from the computer to the router. Meaning ISPs would not be able to know nothing about what you do on the Internet. All they will know is how much you use it and how frequently.

This is very essential. Since privacy is important. Go click on that link, if you don't believe me.

People suffer in unfair conditions in countries with no Freedom. And this same oppression is trying to become reality even in "Democratic" countries. So being able to access information without anyone knowing it. Being able to share information without getting caught. It's very important.

This is why Tor exists. And this is it's goal. Making sure people can be private. And not fear the thought police.

But...

While doing something good. Tor "enabled" illegal activity to take place. Using Tor's software people are able to make web sites that sell illegal substances of illegal publications.

About illegal publications. Well this is the whole point. Censorship is bad regardless what is being shown. But freedom to avoid any publication also should exist.

I get to this conclusion in an article titled Should "Content" be Free?. And following on it I also already proposed a system of how to deal with things you want to avoid.

About illegal substances. I don't like drugs. But it seems like making them illegal just gonna make matters worse. Opening a market for those who want to sell it illegally. And since there is a risk of law enforcement. The price goes up and the quality goes down. Then people waste all their savings just to get 1 shot. And later die. Since it was unfiltered crap.

On the other hand making it legal makes it possible to regulate it. Control the quality. Regulate the price. Regulate who gets it and how much.

Tor's illegal drugs sites go half way. Since there is less risk of law enforcement. The risk is still there. But less. People who sell drugs can think more about the quality and charge less.

Even tho Tor enabled selling drugs. It's made it so you don't waste your last savings. You still have money. And you don't die from it. Since it's most likely good crap. Or at the very least, better then buying it on the street.

And again. I'm even against drinking alcohol. I don't like drugs since it's even worse. But allowing it will make it less likely to be dangerous.

And yet. It's true. People are sometimes criminals. And there is way more crime on Tor compared to censored, controlled and spied on, normal web.

Regime

Imagine a world without crimes. But it comes at a cost of total surveillance. And immediate punishment for a even a slight hint of thoughts that are defined as "wrong".

This kind of Regime is what the normal internet is. You cannot express an opinion against the so called "norm" without being cancelled, censored or something else.

Without a Regime like this. In a country that has a reasonable level of Freedom. In such country there will be criminals. And if you allow privacy. Those criminal will be not easy to catch.

This is why we have detectives. If there was total surveillance. Detectives would not be needed. So how about instead of saying bad things about Tor. Just treat it as a Free country. Where you need a detective to solve a crime. And stop crying about your inability to do it the easy way.

Reputation and Tor's responses

The Tor Project is well aware of the issue of reputation that they have. And they are trying to solve this issue. But perhaps the means that they take, just put them in a further Reputational limbo.

At first Tor Project joined the mob against Richard Stallman. I understand why. Long time ago he said that child pornography should be legal. And since there is plenty of that accessible via Tor. And since people don't like that fact. It's easy to see the managers of the Tor Project go against Richard. Trying to signal "Hey, see, we are against child pornography, calm down".

But as always. It raised eyebrows. How is it that an organization against censorship is pro censorship all of a sudden?

Apart from illegal things Tor gave us an ability to talk freely about any subject. Like for example. An ability to criticise very vocal minorities.

So then I get why the Tor Project now hires people of minorities. And not the White Men. But again a wave of eyebrows-raising hate comes at tor.

What can Tor do?

It feels like Tor is trying to pull an Oscar Schindler on us. He saved a lot of Jewish people from Nazis by pretending to be a Nazi. Since if he would just say out loud that he is saving the Jews. They would execute him on the spot.

But balancing doing good work. While keeping a fair amount of false image. Just so those who want to censor and subjugate will think that you are their friend. I think it's a fair strategy.

How can we end this? How can we make a world in which the Tor Project just can say "Fuck it" and no longer be against all of what it's "enabled"?

Well by using Tor casually. Just like I do. I'm connected to Tor on the GNU IceCat browser. And I post these articles through Tor.

My name is J.Y.Amihud and I live in Ramat Gan, Israel. I am not using Tor to stay anonymous. Quite frankly, I don't give a fuck. I'm using it to make it more mainstream.

Happy Hacking!

Freedom vs Feature

There a big difference in Freedom to have a feature and a Feature to be actually implemented.

lbry://@blenderdumbass:f/Freedom-vs-Feature:1

I'm currently bombed in the Tel Aviv area. Bombs were falling all over the place. Didn't hit this building. I'm staying in what's considered a safe zone. A staircase. And I'm trying to focus on the article. Since focusing on bombs exploding doesn't feel remotely nice now. If I miss out a mistake. That's do to stress. It's 2021 May 11th. J.Y.Amihud from Ramat Gan.

When it comes to arguing with people, one of the most frequent reasons why the argument has started in the first place, was because people are arguing about two different things. While thinking that they are arguing about the same thing.

When this happens, from both sides, each of the arguments makes total sense. But then, the other person is talking about something else entirely, perhaps similar in concept. Maybe even both people call it the same name. But their understanding is different of the subject at hand.

Perspective is one of those things causing arguing. It's true that some objects can have two or more contradictory elements. Like a knife is both good and bad. It can be used for killing and can be used for cutting a salad. The truth in this situation is that a knife is a tool. And what you do with it - matters.

But what I'm talking about is more interesting then a mere perspective idea. And some of you probably thought about it already. The knife example in both cases means the same object. A knife. In what I describe here is when people call two different objects, two different ideas, the same name.

I think it happens more then too often with Free Software and the confusion of **Freedom and Feature**.

Freedom

Freedom is when you have control yourself and things belonging to you.

Feature

Feature is when a given function is implemented in some tool or software.

The Freedom Zeroth

The Freedom Zeroth of the Free Software says:

> The freedom to run the program as you wish, for any purpose.

For **any** purpose. Any what so ever. So you can use the program for something it wasn't designed for.

Let's take a GNU Image Manipulation Program or simply GIMP. It's a image editor. It's used to edit images. You have a selection of tools, filters, plugins and other stuff to edit images. And it does what it was built for, quite nicely.

The Freedom Zeroth covers any type of image manipulation in GIMP. Since GIMP is Free Software. But the freedom zeroth states that I can do what ever. Any purpose. So I have to be able to edit audio with GIMP. Or surf the web with GIMP. Can I do that?

It's yes and no. Since GIMP obviously doesn't have those features. It can't surf the web. Unless it will be implemented in the future. I mean Emacs exists. But we are talking about GIMP, now, in 2021. And it **can not** surf the web.

So does it mean that GIMP is no longer Free Software, since it denied us the ability to surf the web? No it doesn't. Why?

The Freedom First

The Freedom First of the Free Software says:

> The freedom to study how the program works, and change it so it does your computing as you wish. Access to the source code is a precondition for this.

This means. That who ever, can do what ever, with the program. Change it. Add features. Remove features. Do what ever.

Earlier today I had an argument about Free Software with one streamer on Odysee. I said that what I value, is being able to control my computer. And Free Software with the 4 freedoms gives me this ability.

His counter argument said that I can control Windows too. I can change it to do what I want it to do. And to look how I want it to look. And I get that it's possible to some degree.

A lot of software, Free or Proprietary, have settings and API and other means to modify their looks and feel and function. Almost all software have settings.

And arguably it's a modification zone. You modify the behaviour of the Software from with in the settings.

Some software have more settings than other. And even tho I don't use Windows. I believe it has a large amount of very in depth settings. Basic and Advanced. And Super Advanced. But...

Free Software has one more layer of modification. It has the source code. Meaning modification of anything at all. Not only the things that the developer was kind enough to include into settings.

Settings are features. Source code is a blueprint. With any rational Image Manipulation Program. Be it GIMP, Krita, Photoshop or anything else. No matter how proprietary it is. No matter how advanced the settings are. You can't set it to become a web browser. There is no such feature.

But with Free Software there is a source code that you can edit. And you can add this feature if you so desire. Or course, not always you may know, how to add the feature. But due to Freedoms 2 and 3. That say that you are free to give or sell copies of the software. Both exact and modified. You can hire or just ask somebody at the very least, to add any feature you want, or remove any feature you don't want.

And thus, you can surf the web in GIMP.

FSF approved distros

A lot of people claim that the Free Software Foundations approved distros, such as Trisquel, Parabola and Guix are less Freedom respecting. Because they "took the freedom" to install non-free software.

If you install Trisquel and go to it's software manager. And inside the software manager you type "Discord". You will not find it there.

Well this is because it doesn't have Discord repositories added to the software manager. So Trisquel doesn't even know that such a thing as Discord even exists.

This doesn't mean that you can't install Discord on Trisquel. You can. But it will require a tiny bit more effort then expected from a casual user. While software like GNU Jami will just be there in the software manager. And be install-able easily.

The maintainers decided to exclude software that are proprietary from their default settings. Making there a lack of feature to install proprietary software.

There is a similar system to Trisquel. As it comes as 100% Free Software by default. It's called Debian. But Debian is not an FSF approved operating system since it's easy to install proprietary software on it.

From the words of the GNU project: *(this page)*

> Debian's Social Contract states the goal of making Debian entirely free software, and Debian conscientiously keeps nonfree software out of the official Debian system. However, Debian also maintains a repository of nonfree software. According to the project, this software is "not part of the Debian system," but the repository is hosted on many of the project's main servers, and people can readily find these nonfree packages by browsing Debian's online package database and its wiki.

Statements like that makes it seem like the GNU project or the Free Software Foundation are hypocrites. How is it that they are pro Freedom if they are clearly taking the "Freedom" of installing what ever you want, from you?

But they didn't take the Freedom. The took the feature. Saying that a lack of feature such as "installing non-free software" is a lack of freedom. Is similar to saying that Signal or Tor is non-free because they don't have the feature of tracking and sending data to Google and Facebook.

"They took my freedom to be spied on."

If you have Software that has a malicious feature. Like spying of you. You will want to remove it. And if it's a Free Software. You will remove it if you care enough to do so.

For the Free Software Foundation and a lot of people such as myself. Being able easily, by mistake, to install non-free, proprietary software is a malicious feature. It's a bug. And it has to be patched and removed.

If you want to avoid proprietary software. Then FSF approved distros is what you need. It will not guarantee that there will not be proprietary software. Since you can add the feature to install it.

Conclusion

When you debate with people. But you are arguing about two different things. It can be uneasy to say the least. Hopefully this article will help you recognize such situations. So you could catch it, clarify the confusion and get to a satisfactory conclusion.

Happy Hacking!

By this time. It's 22:00 Jerusalem time. The bombing is seems to be stopped. I heard that one of the bombs hit a bus. But I heard so many explosions that I'm scared too even look outside and see how the city looks. I don't know if I'll get up tomorrow morning. This is insane. Anyway. Thank you for reading. If I post tomorrow another article. I'm still alive.

Bombing!

So of my thoughts about living inside an actual war-zone. And a few of the raw emotions I felt while this was happening.

`lbry://@blenderdumbass:f/bombing:1`

Yesterday's night was explosive. If you have access to the news, you probably know about the bombing happening in Israel. More then a thousand bombs were sent all over the place.

I don't like political topics like this. And I don't really want to take any side in this issue. Or argue who is right and who is wrong. What I want to do in this article is to give my experience. So you would have a chance to live through the bombing like the one happened yesterday. But in safe place. From the comfort of where ever you are.

As I type, it is currently 5:08 PM (17:08) Jerusalem time. It's 12th of May 2021. One day after the night of the bombing. Just to give you the context. It's now already 5:09 PM for me until I got to this part.

I heard that maybe there will be another bombing at around 6 PM today, so in less then an hour. It's all rumoured at this point. And I don't read the news.

The Siren

Yesterday at around 9 PM (21:00). Maybe little earlier. I was in the middle of typing my yesterday's article.. I was listening to the soundtrack of Avatar while typing. I finished the article. And I had to move to the next step. Which is proof-reading. Reading it myself to spot mistakes.

For this I usually put a different album. Soundtrack from The Book Thief by John Williams. So I did just that. Also I'm in a voice call with my Girlfriend. And she is doing her things. We usually just sit like that and listen to each other.

I get through the first 2 paragraphs and in the call... In the sound I receive from her... I hear a siren. A bomb emergency siren. Few seconds pass and I hear the same kind of siren on my end too.

This is happening usually for 2 reasons. Either a bomb is incoming. And it's warning people to run to a bomb shelter or take cover. Or it's something like a memorial day when you have a similar siren. But it's for a symbolic minute of silence instead.

I live in the center area. It's not very easy to shoot rockets that far into the country. In cities near the edges you may hear an emergency siren more often. Since it's a more frequent danger there. So it was really weird to hear a siren that has no holiday context all of a sudden.

A minute of so into the siren 2 very loud, booms were heard. Something just exploded. And it was not too far away. So I figured. That's it. The bomb just fell. So the siren will be stopped. And we are done with it.

But the siren doesn't stop. It continues. Another few booms. It feels like there is a lot more bombs now. Compared to something normal. And since they explode not to far away. It's important to take a cover.

The Cover

Since bombings are a usual thing in Israel. We have a lot of bomb shelters all around the place. But not everybody can get to one. So the standard protocol is to get to a staircase in a situation like this.

Due to their construction, staircases are usually a lot stronger then the rest of the building. Also it's usually in a center of the building. Meaning even if a building it hit. It will hit the outer wall. Not the staircase.

I'm currently waiting to get back to the staircase. Now this laptop is charging. For the sake of context. It's 5:37 PM (17:37). Less then half an hour till it's assumed the next bombing will begin.

So let's come back to yesterday. We put something quickly on. No shoes. Just a shirt and something. I grab this laptop. And we go to the stair case. There are a lot of people already there. Some of them are kids. And they are crying. Terrified to hell. Adults are way calmer.

I hear bombs exploding one after the other. In a long, randomized sequence of explosions. I'm scared myself. People are yelling curse words at the situation. We sit. And the only thing we can do is wait.

Five or so minutes later the siren stops. People from the staircase come back to do their business. Those that were shopping in the supermarket on the first floor come back to the shopping. Like nothing had happened.

And the explosion sounds stop too. Finally. The nightmare is over. So we go back to our apartment.

And as soon as I plug the computer into the charger. Another siren starts.

This happened multiple times. For between half an hour and 40 minutes since the first siren. People were coming in and out of the staircase. Then, when it's all ended, I sat in the staircase alone and tried to focus on finishing the article. This is when I added the first and the last paragraphs. The ones in italic text that talk briefly about the bombing. Then I made a quick image in GIMP for a thumbnail while still at the stair case. And I uploaded the article to Odysee.

I went back home and decided that I need to sleep. Since it seemed like the bombs are finished. And nobody canceled the job tomorrow morning.

3 AM During The Night

Before we gonna talk about it. I'm 10 minutes before 6. And I just changed my location to the staircase. Waiting the siren any second. Hopefully it's all just a warning. And there will be nothing.

Needless to say sleeping wasn't the most pleasant. I was waking up from any noise imaginable. But since there would be no sirens along side the noises. I would just pass it on to something simpler. Like a boom sound could be attributed to somebody just dropping something one floor above. So I kept sleeping.

Near 3 AM I heard a sound I could not explain this way. It was 2 more explosions. This time it was for certain that they were in fact - explosions. But there was no siren. So I was trying to convince myself that I will find an explanation for it later.

Few seconds later another siren began. And I had to wake up and run back to the staircase. My Girlfriend woke up too. She entered the call again. And we sat in fear for another half an hour.

This is when on Odysee I saw a notification that one channel I subscribed to just started streaming. So I joined the stream. And started typing my panic into his live chat. This was very embarrassing.

The Morning

It's 6 PM right now. Nothing started yet. Will see.

In the morning I woke up casually. Like if it was a normal morning. I prepared myself and went to work.

I work at a store. Even if some places would be closed. Stores still should work. Since people should be able to get food. So by that logic. I didn't stay home.

I went out. And walked towards the store where I work. It was strangely calm. There was not a single reminder that something even took place the night before. No ruins. All the buildings stood as they were. At least where I was walking.

I came to the store. And it was casual. I just started working as usual. People were coming and buying as usual. It felt like maybe it was just one big nightmare. And I dreamt all of it.

But then this granny comes to me and tells something about the bombing. Something about how unimpressive this was compared to some bombings he had lived through.

And I understand it finally. Nobody gives a shit because they developed some kind of immune system against such attacks. It's just a regular reality to them. Like the cold whether in Antarctica. Or the lack of water in Africa.

People here are just adopted to constantly being bombed at. So nobody given a damn about what happened during the night. They took the covers. Yes. Standard precautions.

Waiting For Another Bombing

This is 6:07 PM (18:07) right now as I type this text. Nothing still happened. I'm waiting here till... Well. I don't know. Till something happens.

6:18 PM (18:18). We decided to go back into the house. Nothing is going on so far. Me and my brother were the only people at the staircase now. The other people seems not to care at all.

6:30 PM (18:30). Nothing is going on yet. Will see. Maybe it was 7 and not 6. I don't know for sure. I don't follow the news myself. Still a lot of fear in the air.

6:45 PM (18:45). Just learned that it's actually 7. I was mistaken all this time. Prepared to go and take the cover again.

6:55 PM (18:55), five minutes before 7. Waiting for the bloody siren in the staircase again. Nobody apart from my brother is here. Probably they won't care until the siren starts. Feels unusually calm.

7:01 PM (19:01). Nothing yet happened. Silence. Suspicious silence.

7:16 PM (19:16). Nothing yet. Feels like it's going to be one of those unresolved tension moments. Where nothing actually happens.

7:20 PM (19:20). Siren! No explosions yet. My GF has went offline. Which is fucking scary.

7:32 PM (19:32). No explosions. My GF is back online. Everything is seems to be fine. The people came into the staircase only during the siren. As soon as it was over they wen back to their business. I'm still scared.

7:36 PM (19:36). I'm the only one at the staircase. Everybody, even my bother, went back to their business. Maybe I will too. Okay. Going back.

8:30 PM (20:30). From my GF side I can hear booms but no siren. I hope it's just somebody messing around. But I'm very scared. **Happy Hacking!**

Is There Any Bad Action Film?

I hear a lot of people say thing they like about action films. I know a lot of techniques in actions films that I very appreciate. But are there any action films without a single redeemable quality?

lbry://@blenderdumbass:f/Is-There-Any-Bad-Action-Film:f

Stuff that happened to me lately, made me think about Michael Bay a lot. And since he is primarily known to do action movies. And since I like Bay's movies. I want to ask a weird question.

Is there any BAD Action Film?

Good Action Films

Before we can talk about bad action films. We need to define a good action film. Most people will define a good action film as a good film with action in it.

This definition will be true for movies like Drive by Nicolas Winding Refn. It's a good film. With not a lot of action. But there is some. So you can technically call it an action film. Even tho it's mostly drama.

What I would define as a good action film. Is that film where the action was made with passion. Where the action is well directed. Let's go over a couple of examples of a very well directed action.

To find a well directed action. Or just simply a well direct film. You may look no further then to the work of Steven Spielberg. For example, his latest film

Ready Player One and the race sequence in the beginning.

Yes, this scene is CGI. Which meant that Steven had more flexibility. But what he did with it was truly awesome. The shots are just the right kind of amazing. The motion of the camera is perfectly juggling elements with in the frame to reveal just the right thing in just the right moment.

The race is insane, but since it's a game where the main character knows how to play. The visuals, even tho seem chaotic, actually very focused and precise. Even tho this is an action scene, you clearly understand what's going on.

And despite all of that. Spielberg is pulling off, making the scene also very exciting. Juggling with surprises, tension builds, tension releases, great camera angles and great camera movements. Awesome zooms. Awesome uses of slow motions. Awesome, awesome, awesome...

Comparing it to something like the chase scene in Knives Out by Rian Johnson. Which is an amazing film. And just a right kind of chase scene for that film. But it was just a simple coverage of what's going on. Clear. But not very exciting from the directorial perspective.

In Knives Out this is the joke. Rian Johnson can direct very good action. The reason the chase scene is so basic, it's because the driver is not a professional driver. It's a nurse. So it makes sense that it would be slow and boring.

Action Not In an Action Film

Knives Out is a detective comedy. It has an action scene in the middle. It's not an action film. Lately a lot of movies use this kind of idea. They are mostly cheap to produce. Since it's mostly people talking. But here and there you will see a scene of total action.

Drive is one such film. It starts with an action scene. It has action scenes with in the film. And they are directed beautifully. But the core of the film is drama. It's a slow paced movie where the main character is looking at the distance while listening to melancholic music.

One very amazing use of this technique comes also from Nicolas Winding Refn (the director of Drive) in his "long movie" Too Old To Die Young. When the main character is being chased by two bad guys in an electric car.

They enter a highway and the bad guys put music on the radio. You hear Barry Manilow's Mandy (a slow Romantic song) playing over a scene of main characters faces inter-cut with shots of the highway. They are tired and nearly fall asleep. And then the morning comes. And they are still driving.

This is not a typical chase scene where they take corners, break through things and drive crazily to make it exciting. It's a straight forward, one guy drives, the other guys drive after him, scene. But using a little bit of imagination Nicolas made it be one of the most Romantic chase scenes ever. And one of the greatest comedy moments of his career.

Few more notable examples of action scenes in otherwise non-action films will be the drama films from Steven Spielberg. He is a master at crafting both good drama and good action. So no wonder good action bits exist in some of his drama films.

For example in his film Bridge Of Spies, which is a story about a very good lawyer. And the main set pieces are, him talking cleverly. There is an action scene of a plain crash. For a few minutes a serious drama film becomes an explosive action movie.

Bayhem

When I talk about well directed action, a lot of people are quick to differentiate Steven Spielberg's work from somebody like Michael Bay. Claiming that Michael Bay makes badly directed action.

I don't think that it's the case. Michael Bay has story problems. Character problems. But his action is very good. Sometimes he might do a few questionable decisions. But more often then not he is very good at action.

Bayhem is a term describing a specific style of movie making. A style very hard to replicate. This is the style of Michael Bay.

It's super hot people standing epicly. It's low angle shots. It's stupid humor moments. It's a lot of American Military. Nice cars and a huge number of explosions. But it's not easy to do right. As evident by films trying to replicate Bayhem.

One critic said about his film 6 Underground that Michael Bay has perfected the art of audience manipulation. He knows just the right string to pull in just a right way to make you feel just right for the movie to be always entertaining. Even tho more likely, nothing of substance is shown.

Michael Bay tried to make a serious movie while making Perl Harbor. He went slower. He went with much more care. But then realized that his Bayhem impulses are stronger than any other artistic vision he might've had. So this is why Perl Harbor is the way it is.

The movie is not good. The only good parts are the bombing scene and some other action. This furthers my point. While Michael Bay's films as films are not very good. His action is just the right kind of action for my taste. Bayhem. I love it.

Mission Impossible 2

One more example when it comes to bad action films is usually Mission Impossible 2 by John Woo.

This is a sequel to Mission Impossible by Brian De Palma. The first film is a slow burning, tension building spy drama.

The second film is basically Bayhem. Which if you look at them side by side, make no sense at all. But from some perspective it is it's genius.

John Woo is a great action director from Hong Kong. A place where they know how to make a good action scene. His specialty is Gun-Fu. A type of martial art style of using guns in an epic way. And he is also very well known for using white doves all the time.

His films like The Killer and Hard Boiled are regarded as classics of Hong Kong action cinema. Epic, explosive Bayhem with lots and lots of sparkles and slow motion shots. The camera always moves in a ballet, turning all this craziness into an elegant opera-like experience.

Imagining this director doing anything less than what he did with Mission Impossible 2 would be wrong. He is not Bryan De Palma. He is fucking John Woo. Deal with it.

In my opinion, it's true... This doesn't feel like a sequel to the first film. But the film is brilliant in his own right. The action is beyond amazing. This is not a bad action film.

Samurai Cop

There is a type of movie that's controversial. It's when a movie is so bad. It becomes good. A good example of it is The Room by Tommy Wiseau. It's so bad that it's one of those rear films playing in theaters for more then a decade. It's so bad that an Oscar nominated film The Disaster Artist by James Franco was made about the making of this film.

The Room is famous and infamous. But unfortunately for this article. It's not an action film. Samurai Cop is.

Samurai Cop is one those low budget films trying to be cool. But having no knowledge or money to do anything real. They try to put some "good" action set pieces. There is a car chase. There are explosions. There are shootings.

But the movie executes all of it so badly that it's just amazing. You can't keep yourself from smiling. Everything is just the right amount of cringe.

Interesting how Tommy Wiseau recognized his movie as a bad movie classic. And produced / starred in Samurai Cop 2: Deadly Vengeance. Which was shot the same way as The Room. Shooting both Digital and Film in the same time. True visionary.

Conclusion

I think since I make my own movies I know the pain of making one myself. And it contributes a lot to my inability to hate films. It's either brilliant, or so bad that it's good.

The only 2 action films that I think are bad. Are those that I made. Since I always notice my mistakes. And I always wish, I was able to notice them before.

Happy Hacking!

My Thoughts on the Gaza Bombing Situation

I couldn’t just leave the politics alone. Could I?

`lbry://@blenderdumbass:f/My-Thoughts-on-Gaza-Situation:a`

This article going to be political. Note that I'm not a political official. And I have no direct influence on anything happening today in Israel. Just 5 minutes ago 2 rockets from Gaza exploded just above my house. I'm glad that we have the Iron Dome.

For Those Who Don't Know

This week or so, was and is, a tense week for the middle east area. Especially the Israel part of it. Since few days ago on 11th May 2021 we were shot at by Gaza with more then a thousand rockets. And it keeps happening. Only much less.

I have written an article about my experience with the bombing. But I avoided political discussion about this issue. So I think I might give my views on it here.

Disclaimer

I'm not a politician. And probably not very smart. And probably not very educated about some things. But I do have thoughts. And those thoughts I gonna present in this article.

Please be free to yell and me and disagree with me. But I hope you can give me a genuine, constructive criticism. And not just yelling at me about how much of a dumbass I am. Since I know that I am a dumbass. It's in the name of the channel.

As I type, another siren is occurring right now. And one very loud explosion just happened near by. So I might be too emotional about these issues. And might not think some things through.

But from the other side. I might have more insight since I live here.

Middle East Situation

For a very long time, Israel was a Jewish country. But since Jewish people left it. And Arabs were living here instead. Rightfully so. I mean this is a land and any person may live on the land. It's my opinion.

I think due to Hitler and the Holocaust, something had to be made. A Jewish country had to be reinstated. Because living without a Jewish law and Jewish army was too dangerous. So the Israel area, naturally was the choice to reinstate the country.

Arabs that lived here didn't like it. Since we are suddenly demanding them to move away. I understand that it was a decision that made sense at the time. But I think they didn't think much through it, back then.

Democracy

In my personal opinion every human on earth. No matter how evil. I include in it the guy who pressed the "send" button on the rockets. All of them should have Freedom.

Software Freedom is on of those things that I can talk more about. Since I understand software more. But Freedom in general is essential.

With Free Software it's easy to make Freedom possible. If somebody didn't like a feature. They can fork it. Modify it for them selves at the very least. So nobody has to agree with anything in the software code.

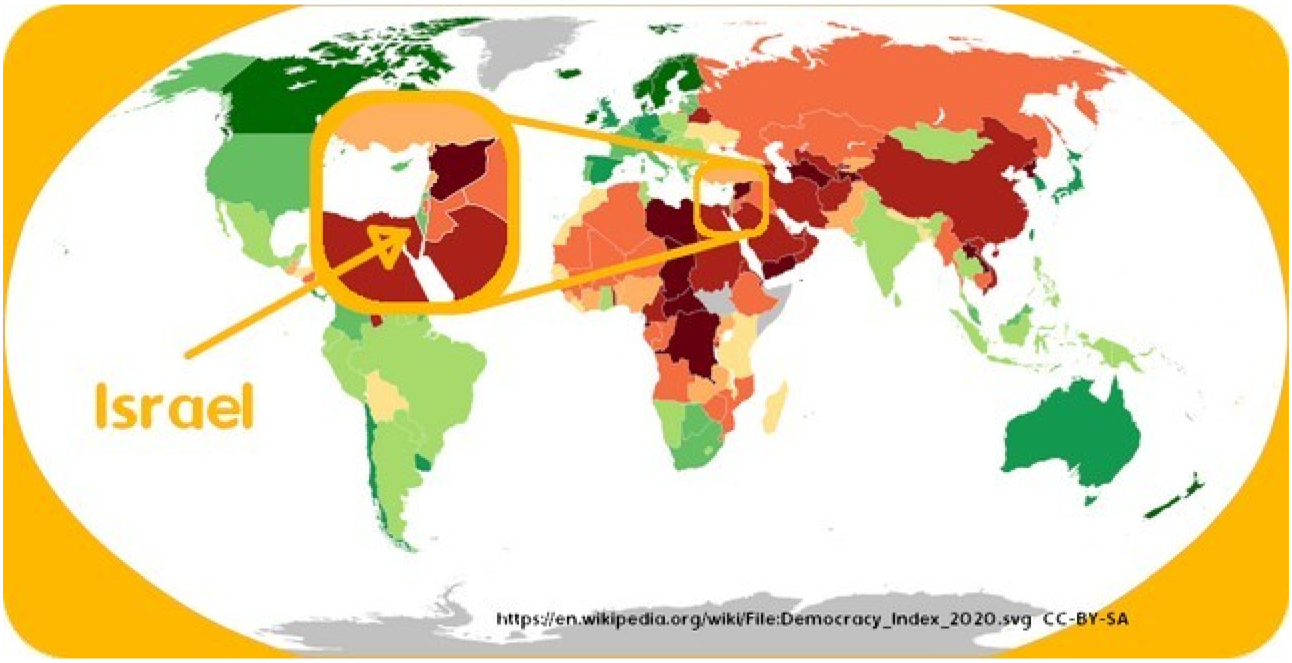
For law it's not that simple. If everybody could fork it. If everybody could make their own version of law, that they like and that suits their needs. Living in a community of people would be very terrible.

For example. I agree with that killing should be illegal. That other person disagrees with me. So he forks the law for himself. And it's legal for him now to kill me. How should we judge it? This is not going to work.

The best attempt to give people freedom over the law is called Democracy. Which is, in very basic terms, when the largest group of people vote on the law. And the minority has to accept it.

I know it's voting on a Representative who will maybe choose the law you want. But you got the point. With Democracy you have some Freedom with what law is. It's just not really your personal freedom. But a Freedom of majority or people in that country plus the most voted representative's personal believes.

Islamic Countries vs Democracy



https://en.wikipedia.org/wiki/File:Democracy_Index_2020.svg CC-BY-SA

There is a whole discussion about whether Democracy could even exist in the Islamic Countries. There is a whole article on wikipedia about this whole issue.

But the fact of the matter is. Only 3 Islamic counties have Democracy. Indonesia, Malaysia and Tunisia. Some have Hybrid Regime. Which is when they are trying to be Democratic. But still are very Authoritarian. And most of them are Authoritarian Regime.

If you see the map on the top. There is one green pixel in the middle east, which is Israel. Small Democratic country surrounded by huge red zones.

It is questionable to say the least that Israel exists at all. Israel took the place. Moved a lot of people. Made bad decisions in terms of disrespecting those who were there before.

But I think currently. At the current stage of today. Israel is this little pixel of hope. In otherwise hopeless region. It's the only place on the map where people can disagree with the government. And where people can influence the decisions of the government.

If Freedom is important, which it is. Israel is important for the whole region. Unless Islamic Countries just become Democracies all of a sudden.

Rock in the Shoe

You walk down the street. You pass a little unpaved area. And a tiny rock falls into your shoe. Making it very uncomfortable.

This little green pixel of Israel pushing onto the middle east feels like that. It feels like a little tiny rock that fell into your shoe.

But a good amount of well spoken people made their decisions clear enough. Israel is not there to harm people. It's there to protect people. Mainly Jewish people. So Israel is very careful not to strike anybody around. Since Israel is trying to be a **soft** rock that fell into your shoe.

My Girlfriend's father hated me from the beginning. And he hates me still. At first he wanted to argue with me using reason. Now, he just wants me to die. Since there is no more reason left on his side.

I'm not there to hurt him. And I'm not there to hurt her. He just simply doesn't like me. I'm a little rock in his shoe. And since he is loosing the arguments by speech. He is trying to win me physically.

I think this is similar to what's happening with Israel and the bombings. They are not very friendly. And there is this peace of land that promotes friendliness. AKA a Democratic country. They just simply do not like friendliness. So they argue us out.

We win the arguments. So they bombs us. Since there is nothing that they can argue about anymore.

Would you kill the rock in your shoe if the rock gave you enough arguments to stay there?

Conclusion

I think in order to fix this. We need to get more people into talking. We need to get more people who knows how to talk. How to explain things in such a way that a completely opposite ideology will agree with us.

And this I mean from both sides. Islamic countries need more well spoken individuals who can persuade people. Since then they will have more confidence in that they can do it without bombing everyone.

If you just simply bomb people. Those people will agree with you less. This is not going to help. Don't form hate groups. Lynch mobs. Don't try to argue with violence. Because an act of violence is not an argument.

Happy Hacking!

Comedy Rock

Sometimes you want your funny to be also rockin'.

lbry://@blenderdumbass:f/comedy-rock:e

I've talked about software a lot on this channel. And I talked about movies sometimes. I even talked about politics and personal experiences. But I don't remember talking much about music.

There is one genre of music that I enjoy a lot. And it's comedy-rock. It's when they play a rock song. But it is not serious. It's all jokes. And it's amazing.

Rock

Rock music is a very broad style. From one side you have styles like pop-punk-rock which can be confused with regular pop music. It's pop, but using live instruments like electric guitars and acoustic drums. Sometimes it can get a few heavier moments. But usually it's normie-friendly.

From the other side you can have styles like suicidal-depressive-black-metal which sounds like your radio accidentally picked up a broadcast from hell, and it's an opera of souls screaming in eternal pain.

Rock is wild to say the least. The world of Rock derivative genres seems similar to GNU / Linux Distros. All similar in idea, but different in execution.

Rock is when you use guitars with effect pedals and real acoustic drums. Sometimes you can add to it things. The most famous addition is vocals. But it's not necessary to be counted as rock since there are a lot of Rock instrumentals.

Some genres of Rock like Symphonic Metal add to it a whole orchestra of instruments. And it's sounds like a heavy opera.

There is enough diversity is just rock, to not listen to anything but rock.

Murderdolls

When I was 15 or so and was to trying to find the most edgy music to listen to. I typed into a music search application the word "Murder", and what I got was amazing. I discovered a band named "Murderdolls".

Officially it's a Glam Metal, Horror Punk and Heavy Metal band. But I think it deserves to be called a Comedy Rock band since it's very hilarious.

Murderdolls are not your typical band. They have only 2 albums, released 8 years apart from each other. No current members. And all the member doing more serious music elsewhere. Murderdolls is just a side project for all the band members.

It explains why it feels like Murderdolls is a kind of parody on the Heavy Rock genre. While being very heavy at the same time. It's one of the heaviest bands out there. And yet. It's all only for fun.

They have a very sick sense of humor. The kind you would find in a Lars Von Trier film. The lyrics of the song "She Was a Teenage Zombie" by Murderdolls illustrates it very well.

> I don't know what I'm feeling or what I should say. I had my life changed the day I stumbled on her grave. Not a necrophiliac, I guess I was bored. I love to spend my time with a fresh embalmed copse. Of course...

For some reason it sounds like it's both very dark and in the same time silly as hell and obviously meant to be funny. It's like horror films by Sam Raimi. Everything is so obvious and intense that you can't stop laughing.

Tenacious D

You are probably familiar with Jack Black. A Hollywood actor. The Jablinski Games. The guy who voiced the Kung Fu Panda.

And some of you maybe seen his movies like School Of Rock and The Pick Of Destiny where Jack shows his skills as both singer and a musician.

He and Kyle Gass actually have a band called Tenacious D. The band shown in the film The Pick Of Destiny. Those two played themselves in the film.

There was one thing that was changed obviously. Jack black is not from a Catholic family. He is Jewish. But similar restrictions on Rock Music are in both religions. I know that from experience.

What Tenacious D pulls off is quite extraordinary. Apart from just being a good band. I mean their music is actually rocking good. They also pull off some amazing humor. Sometimes even layered in various ways.

For example. Let's take a look on their album Rize of the Fenix.

https://upload.wikimedia.org/wikipedia/en/c/cd/Rize_of_the_Fenix.PNG

The album is about Rize from failure. The album is referencing the poor performance of their film The Pick Of Destiny. And the Phoenix is a bird that burns and then rises back from it's ashes. Signaling an epic return of the band.

But then you may notice the intentional misspelling of the word "Phoenix" as "Fenix". Then you may look at the cover art depiction of the Phoenix is question. The Fenix is clearly a Penis. And it is the Tenacious Dick that they are talking about. Tenacious to get laid.

Their music is all about becoming better. Becoming more famous, because they are two fat dudes. And the only thing that will make them laid is being famous.

You can hear their struggle with this "getting laid" theme very clearly in their lyrics. For example here is a part from "39" by Tenacious D.

> Shes 39, but she still looks young. Not very young but a... lot of fun.

By the way. This song was apparently never written. And Jack just came up with the words during recording. Because if you listen to the entire thing. Nothing of it looks like it could be designed. It had to be done instinctively, at random.

This theme of getting laid even passes on various characters in the Tenacious D music universe. For example the infamous Roadie from "Roadie" by Tenacious D.

> Then a beautiful girl come to me. She... says "Hey, can I suck a your dick?" I say "Yeeis", I am in love. Then she quickly say "I sucked you dick. Now give me that backstage pass. I do not want you, Roadie. I want KG's chode."

This is now making a statement that Kyle Gass, KG, is instantly hot enough since he is a Rock Star. And just some Roadie is not up there with Hollywood Jack and the Rage Kage. Making both statements reinforce the Tenacious D running joke about getting laid.

But it's not the only joke that they make. For example their song "Tribute" is about that feeling when you have an amazing idea while high. And then can't remember it when you are sober. Only can remember that you had the idea. Not the idea it self.

The song is basically a story about how they wrote the best song in the world. But they can't remember it.

> This is not the greatest song in the world, no... this is just a tribute. Couldn't remember the greatest song in the world, no... This is a Tribute, oh...

Conclusion

I like Comedy Rock Music. This article was just my fandom, recommendation, I guess. Sometimes there is just no theme there. Just a simple observation of amazingness of something.

Happy Hacking!

Why You Keep Using Proprietary Garbage?

There a bunch of tricks that proprietary software companies deploy to hold you inside their shackles. Knowing about a problem is the first step in solving it.

`lbry://@blenderdumbass:f/why-you-keep-using-proprietary-garbage:6`

There are few evil tactics used by proprietary software companies and free software companies that makes it very hard to switch software. You probably know the feeling I'm talking about.

You ditch YouTube, de-google your phone. Install GNU / Linux on your computer. Use Brave Browser. But you can't stop using WhatsApp or Discord. Since most of your friends are on these platforms. And you don't want to loose them.

Some people can't stop posting on YouTube when having a good following on Odysee. It's because a lot of people are still going to find them on YouTube. And it's going to be "*not enough*" to just use Odysee.

On the other hand. There is a this good hardware. Librem 5 phone. Or the GNU / Linux laptops and desktops from System 76. But you never get one. Since you rather get a cheaper windows computer requiring proprietary drivers.

Those tactics are either Network Effect, Bandwagon Effect or Economies of Scale.

Let's take a look on each one of them and discuss what should be done with them.

Network Effect

Network Effect is prevalent largely in messaging and social network applications. It's when not only one person is using a given platform. But it's when multiple people using it collectively. And the platform is built to take advantage of this fact.

For example the infamous Facebook. It's designed so the Facebook developers wouldn't need to publish things them selves. The users are both publish things. And both see those publications.

Meaning the more users you get. The more value the platform gets. The less chance a given user will just abandon it for something less evil.

Similar to let's say WhatsApp, the more contacts you have in the app. The more people are talking to you through this specific app. The less chance you have to get away from it.

If you were talking to just one or two people on WhatsApp. And suddenly they decide to update their privacy policy something a bit more evil. You could easily just ask the 2 people to go to a different messenger. And then quickly move.

But since you most likely have 10s or even sometimes 100s of contacts. You will accept what ever it is their are asking you to accept. Because it just doesn't worth, to stop using it. And persuading hundred people to move with you, seems like a huge endeavour. So you don't. You just go along with what ever it is WhatsApp does.

Similar thing is with Discord. A lot of people will just not move to something else. They don't like a particular aspect of the alternative Free Software. Or they are not going to move because of their own Network Effect. So you stuck using proprietary garbage apps like this.

Bandwagon Effect

Bandwagon Effect is similar in concept to Network Effect. But puts the pressure at a different point. Bandwagon effect is why garbage like Tik Tok is so popular.

In Tik Tok you have private messaging. But I'm sure that nobody actually uses it. Beause Tik Tok is a short video platform.

They took the YouTube algorithm approach to searching videos. But they took out the feature of choosing a video. So you end up watching only what's recommended next. But since every video is a couple of seconds long. You don't mind doing it.

This is why Tik Tok had an initial user base. It's finding your tastes. And then reinforcing them on you. Making it a little nasty, dopamine inducing, drug.

But Tik Tok success now, is due to a different phenomenon. It's because so many other people use it. You feel like you have to do it too.

There was a study made by physiologists some time ago. I don't really care to look it up. They put a group of people into an elevator. But all facing the rear wall, not the entrance. And they recorded the reactions of those normal people who enter the elevator. Those people who don't know that there is any study even happening.

Those poor people, at first, stand facing the entrance. The normal way. Then they hesitate. And eventually. Few seconds after they enter. All of them turn the same direction as the rest of the people in the elevator.

This is the Bandwagon Effect. If enough people do something, no matter how stupid. You will be pressured into doing this as well. This is why it's hard to get away from a lot of proprietary software. Since usually it's the software used by the most people.

Economies Of Scale

Librem 5 is more expensive then a spying android phone because not a lot of people buy Librem 5. If something is popular. It's usually sold in larger quantities. And with larger quantities you can have lower prices.

When a hardware manufacturer makes hardware. He sells you it with the cost of it's manufacturing plus the cost of his interest. What he want to gain.

I mean, the manufacturer not only wants to get back the money he put into the manufacturing. He also wants to buy some cheddar. Pay the bills. So he expects to get a percentage from it.

Let's say he manufactures a phone for 20$ and sells it for 30$. From each phone he will get 10$ to spend to his own needs. But if you buy 10 phones. He can charge only 25$ per each. Because he will still get more money as the result. But you will have more incentive to get the 10 phones instead of just one. This is called a discount.

Another factor is that Phone Manufacturers do not produce all of the details of the phones. The chips are usually bought from a different company. The screens are bought from another company. This makes the phone cheaper. But also reinforces the economy of scale.

Since now it's not only the manufacturer that will make the phone cheaper. But also the manufacturers of it's parts.

Basically the more Librem 5 you buy. The more parts for new phones the company buys. The less money they pay for those parts. The less production cost there is for each Librem 5. And combine it with the general discount. The phone will become affordable.

But since nobody buys it. It's too expensive to buy. And it keeps people from buying it. A bad feedback look.

But What Can You Do?

True friendship is hard. My dad used to tell me, when I was a kid. That true friends are those who will keep you in, hide you, while you are running from the police. Those who will refuse are not true friend.

I have a different definition of a true friend. It's those people who want to talk me so much, that they will install an "unknown" Free Software application just for it. I have those friends.

When I was 15 I was F'd. I used Facebook. I had a lot of "friends" on Facebook. Only about 2 or 3 of them are still talking to me. Because they were good enough friends to make a Rocket.Chat account. Or to use Tox or Jami.

To break the Network Effect of proprietary communication software. You have to not use them under any circumstances. Unless it's a live and death situation.

Using Windows for a few minutes is not that bad. Nobody going to suffer from that apart from you. Using Skype on the other hand. This will make another person suffer. The one you are talking to using Skype. And this is not okay.

For the economies of scale. We have to organize and buy those phones. It's not going to be easy. But it has to be done. Or perhaps a different idea could be used.

Happy Hacking!

Triangulation

I know this book is a bit random. But so is life. Actually I just know how to make you excited about Triangulation.

`lbry://@blenderdumbass:f/triangulation:8`

A lot of people around me are doing things related to 3D graphics. Either visual effect, animation, modeling or game design. All of them need a three dimensional representation of an object that they want to portray in their art. And I want to explain some things related to it.

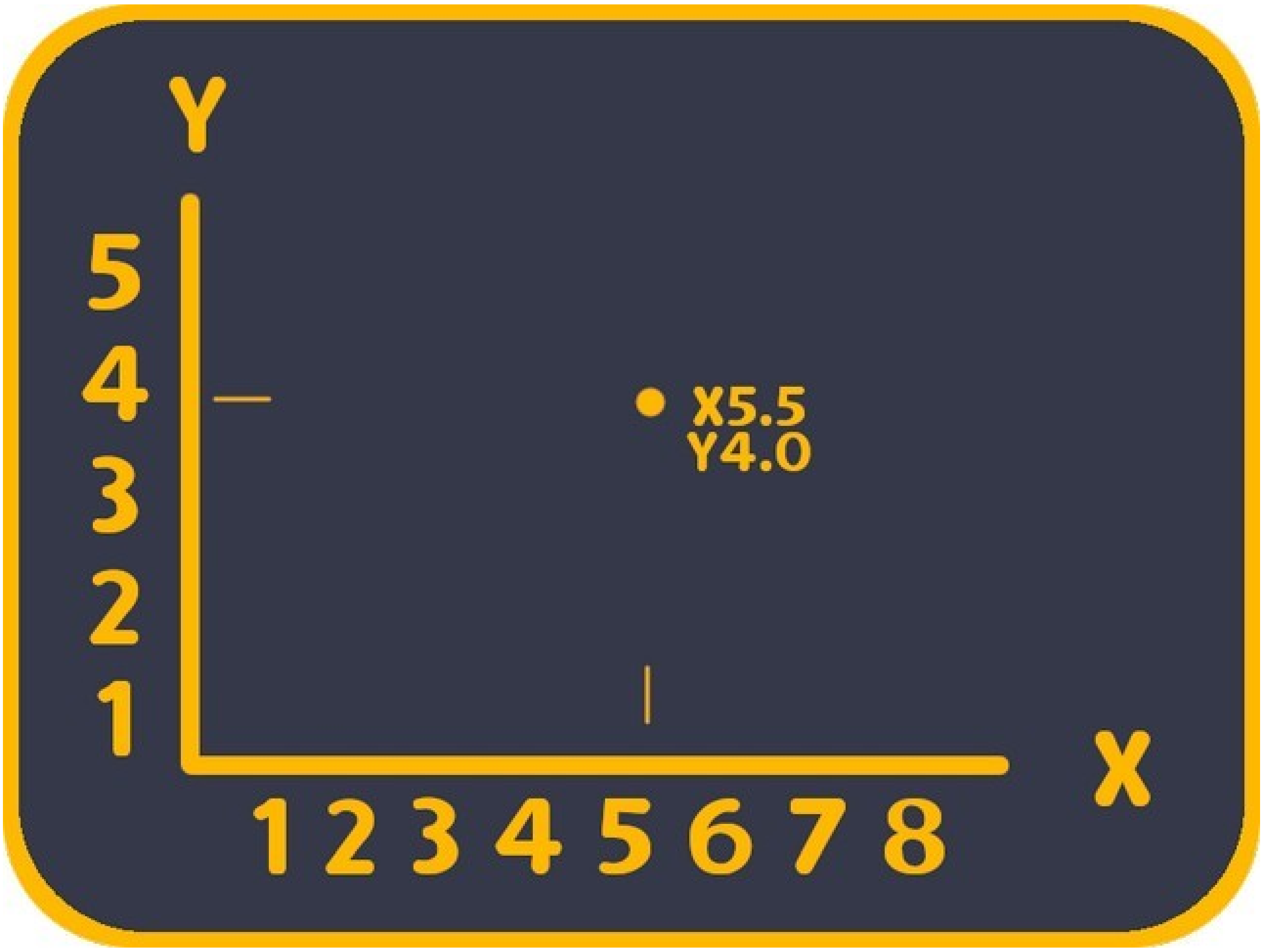
Triangulation

This article is about triangulation. But not about the term that means to find a center point of a triangle. This is a different triangulation. I touch on it briefly in my other article.

This triangulation is used to render 3D images. Since sometimes a polygon isn't what you need. Having more than 3 dots may cause problems.

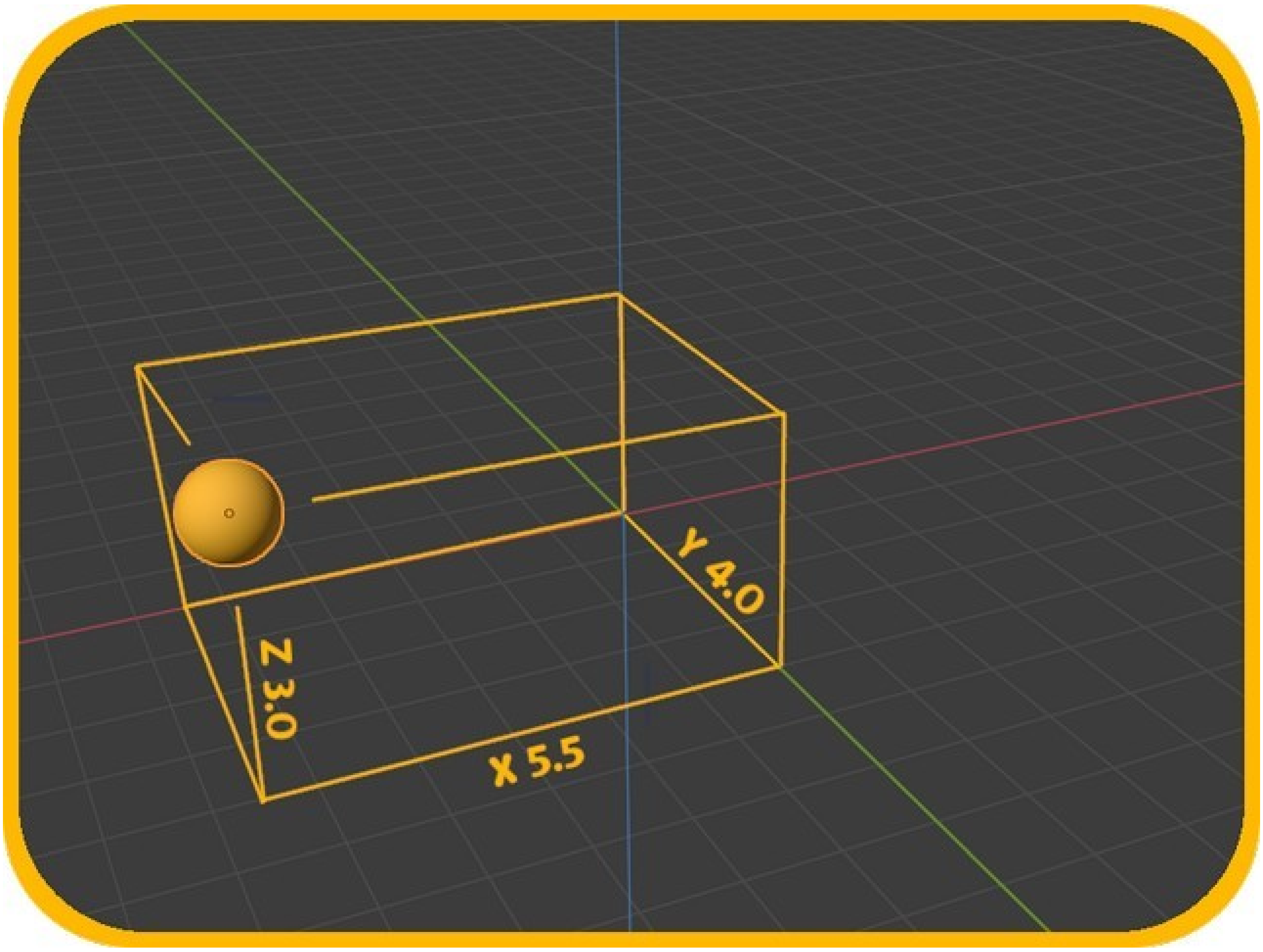
Dot In Space

Have you ever seen a graph? The kind they teach you in school. One direction is the X axis. And the other direction is the Y axis.



On the picture above you can see a dot. A dot in a position of 5.5 on X axis and 4.0 on the Y axis. If you combine both numbers into one. Like (5.5 , 4.0) as one little object like this. It's called a vector.

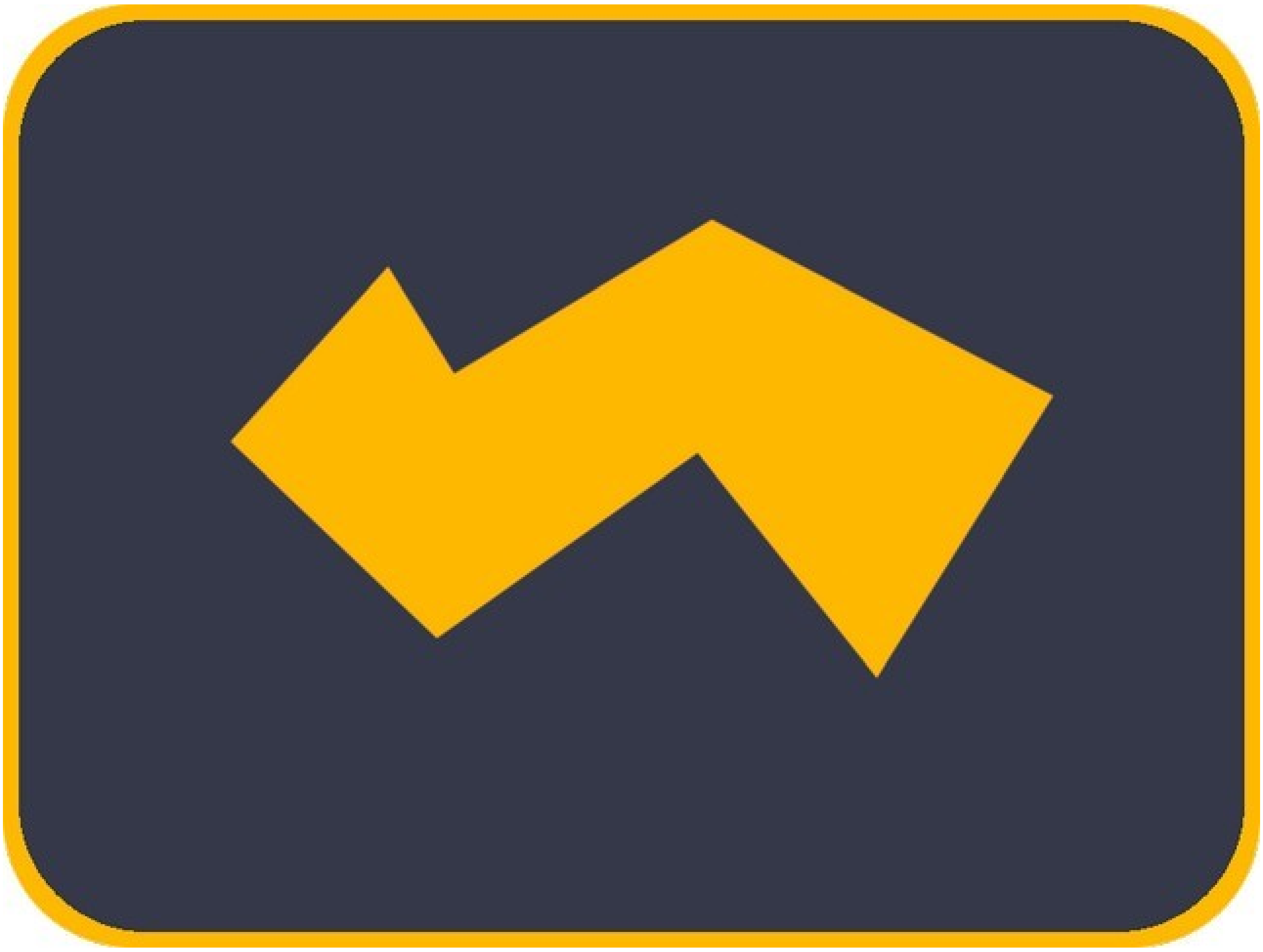
Vector is not 2 numbers. Is not 3 numbers either. It's any number of numbers. For example (5.5 , 4.0 , 3.0) is a vector containing 3 axis. It can be used to put dots in a three dimensional space like so.



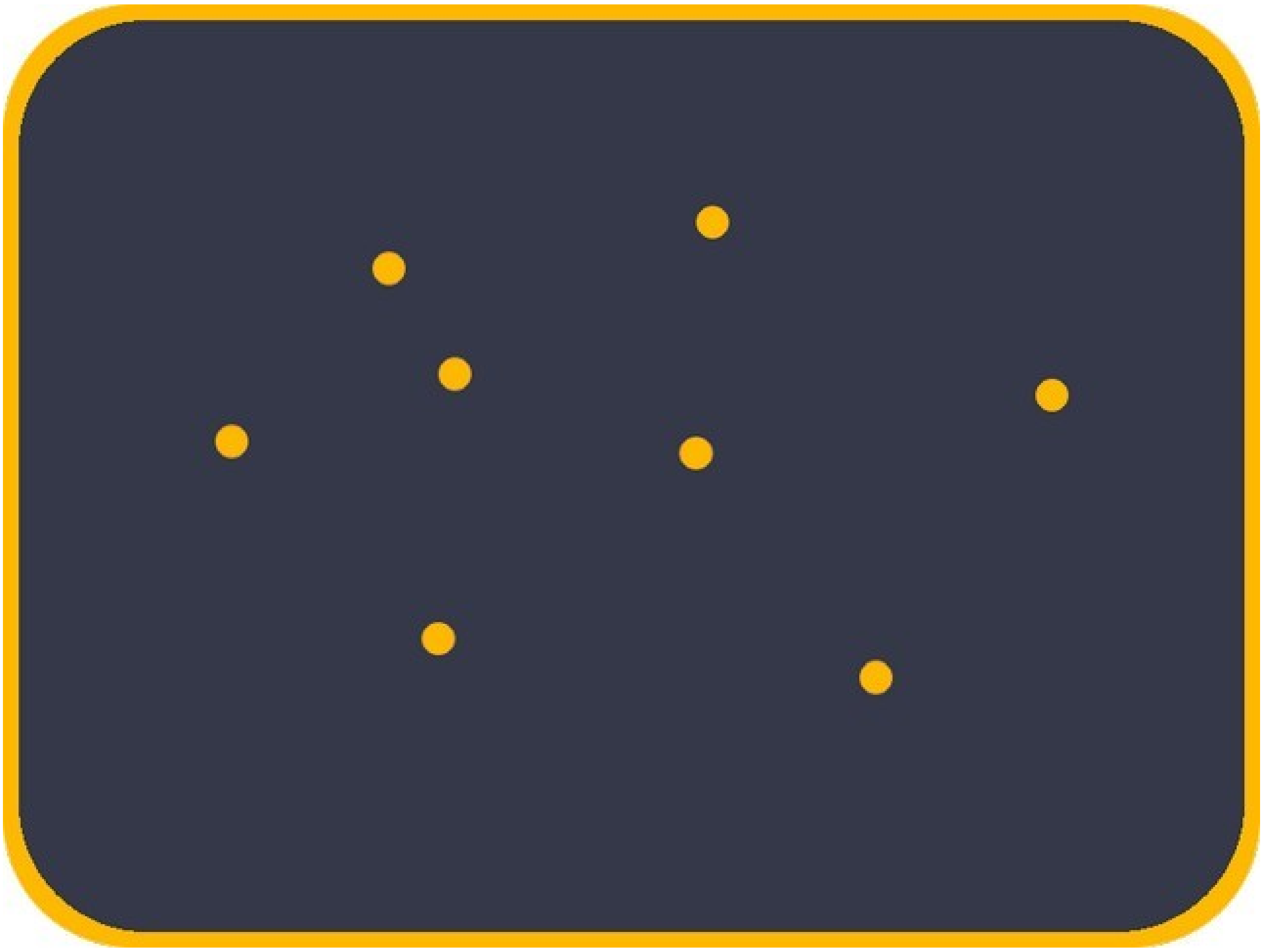
You can go even further and make vectors with 4 dimensions and 5 dimensions. But that's a topic for another day.

In order to render a shape, one dot is not enough. So you need a number of them.

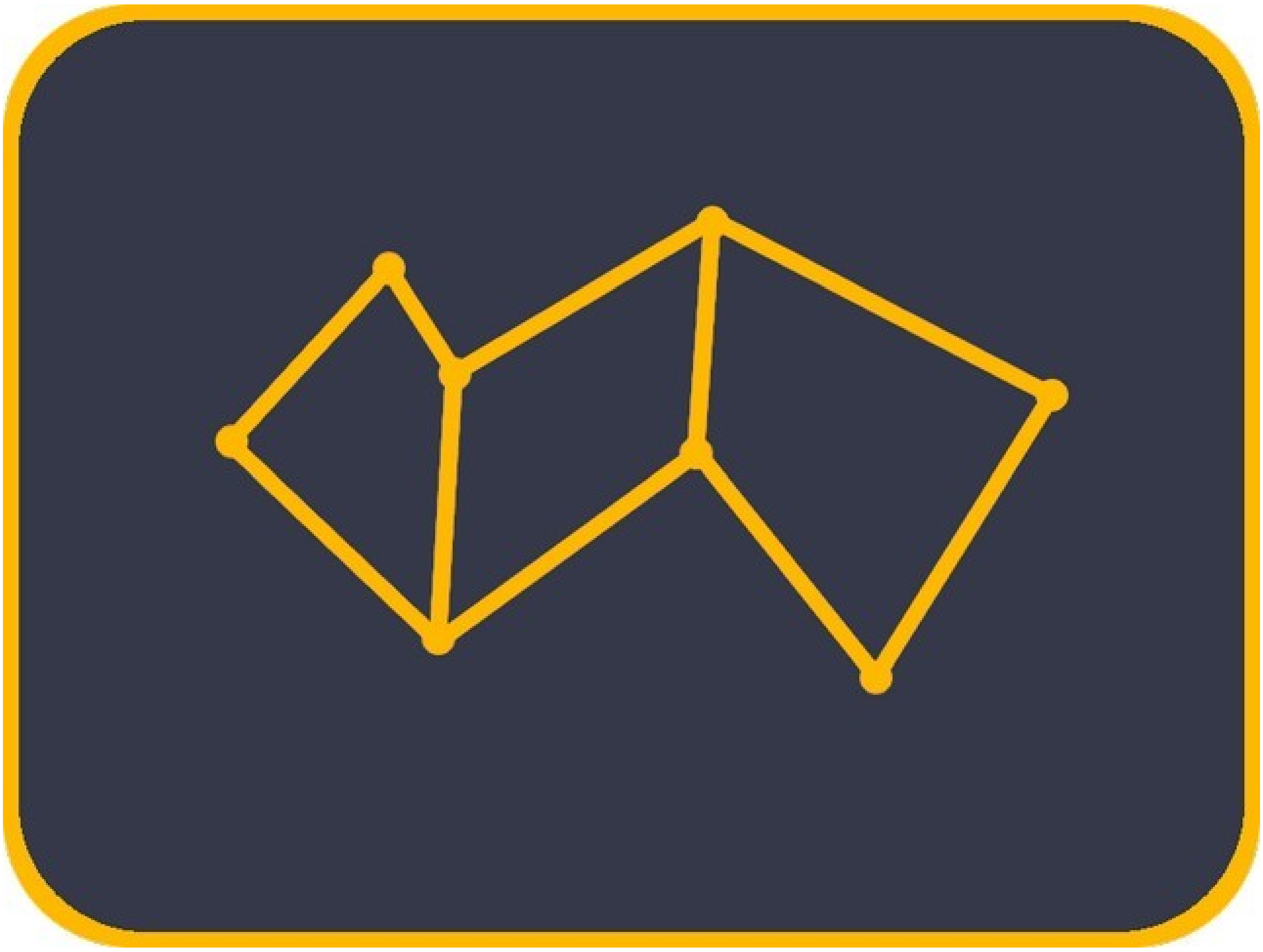
2D Shape



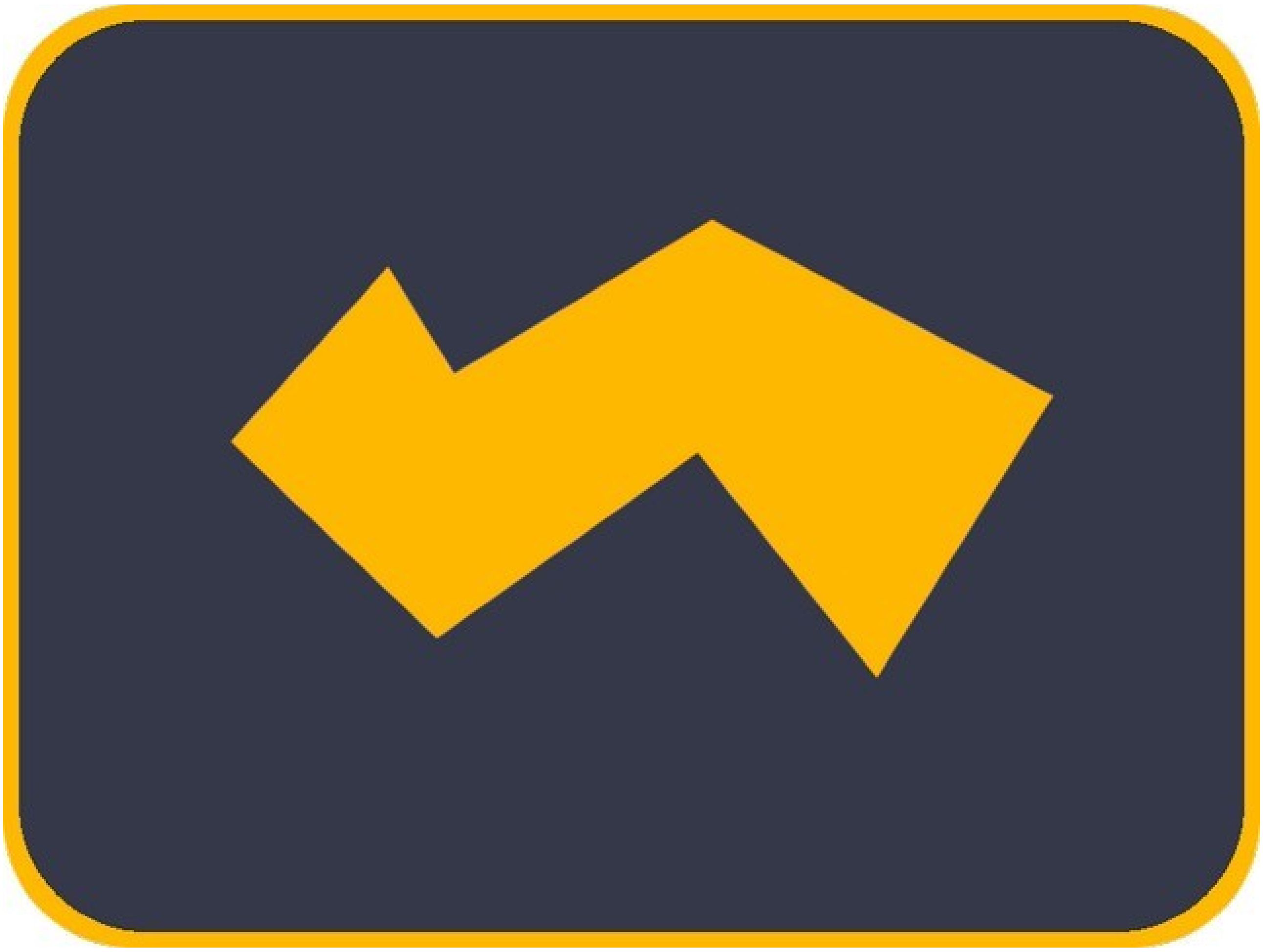
Let's say you want to make a shape from above using 2D polygons. First step will be to define all dots, all corners of the shape.



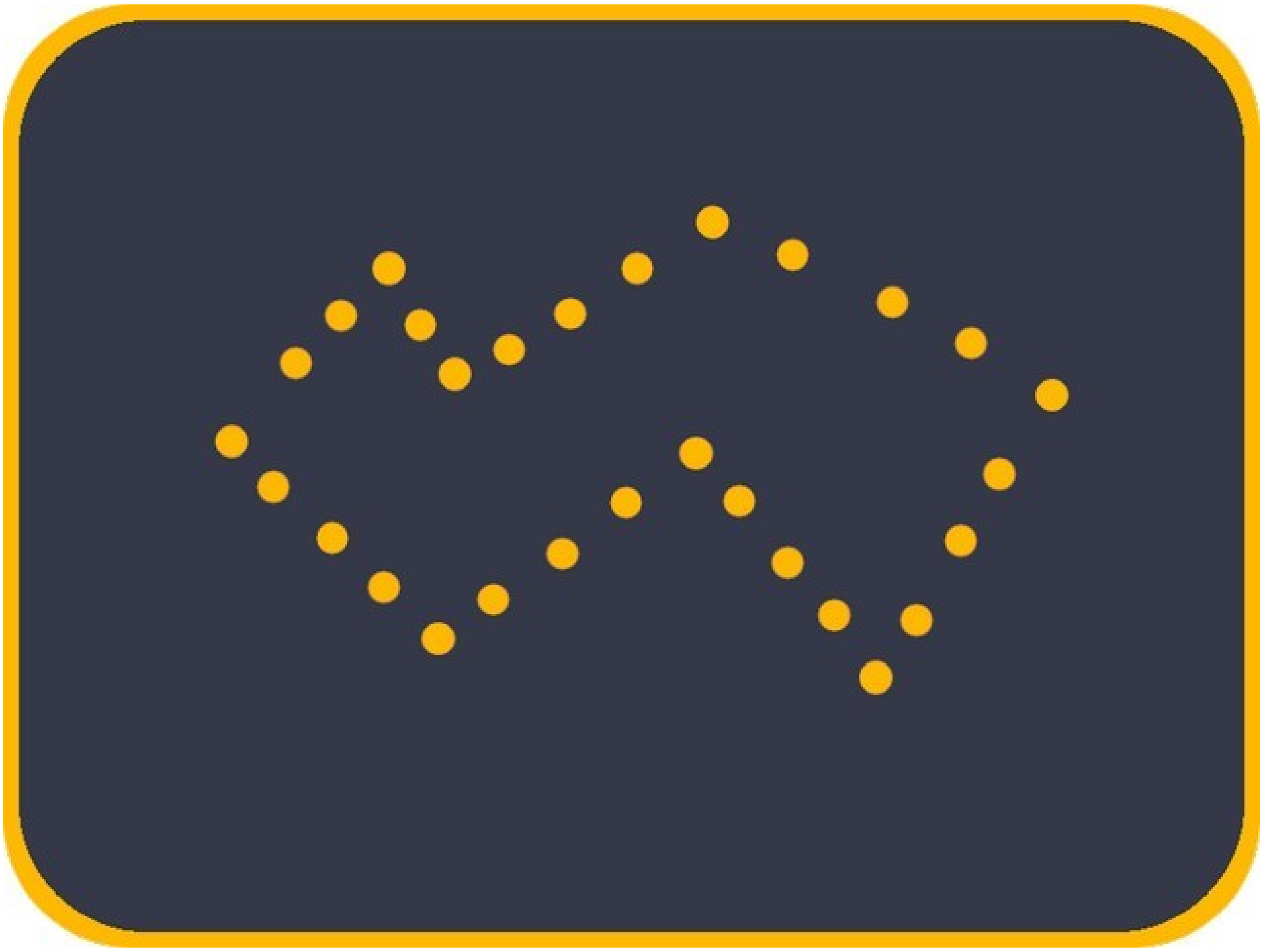
This dots are usually called vertices. Each of them has a vector to represent their location in the space. To draw this shape, I found that 8 vertices is enough.



Next step would be to connect the vertices with edges. Lines that go from one vertex to another. Between those lines we tell the computer to fill color. Or in other words. Make a polygon.

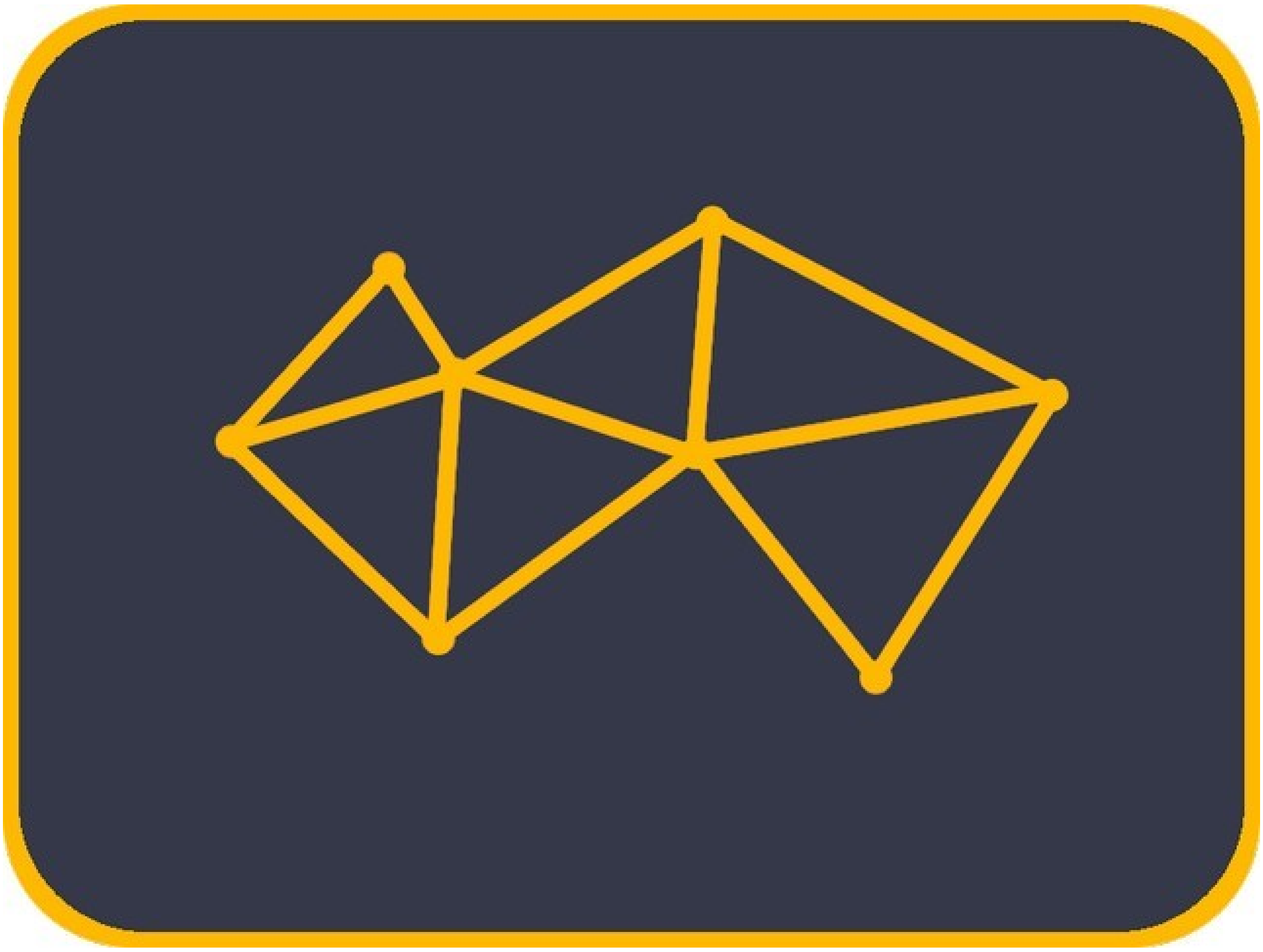


And so we make our shape. This doesn't mean that you have to have exactly 8 vertices though. For example you can build this shape with 29 vertices. Like this.



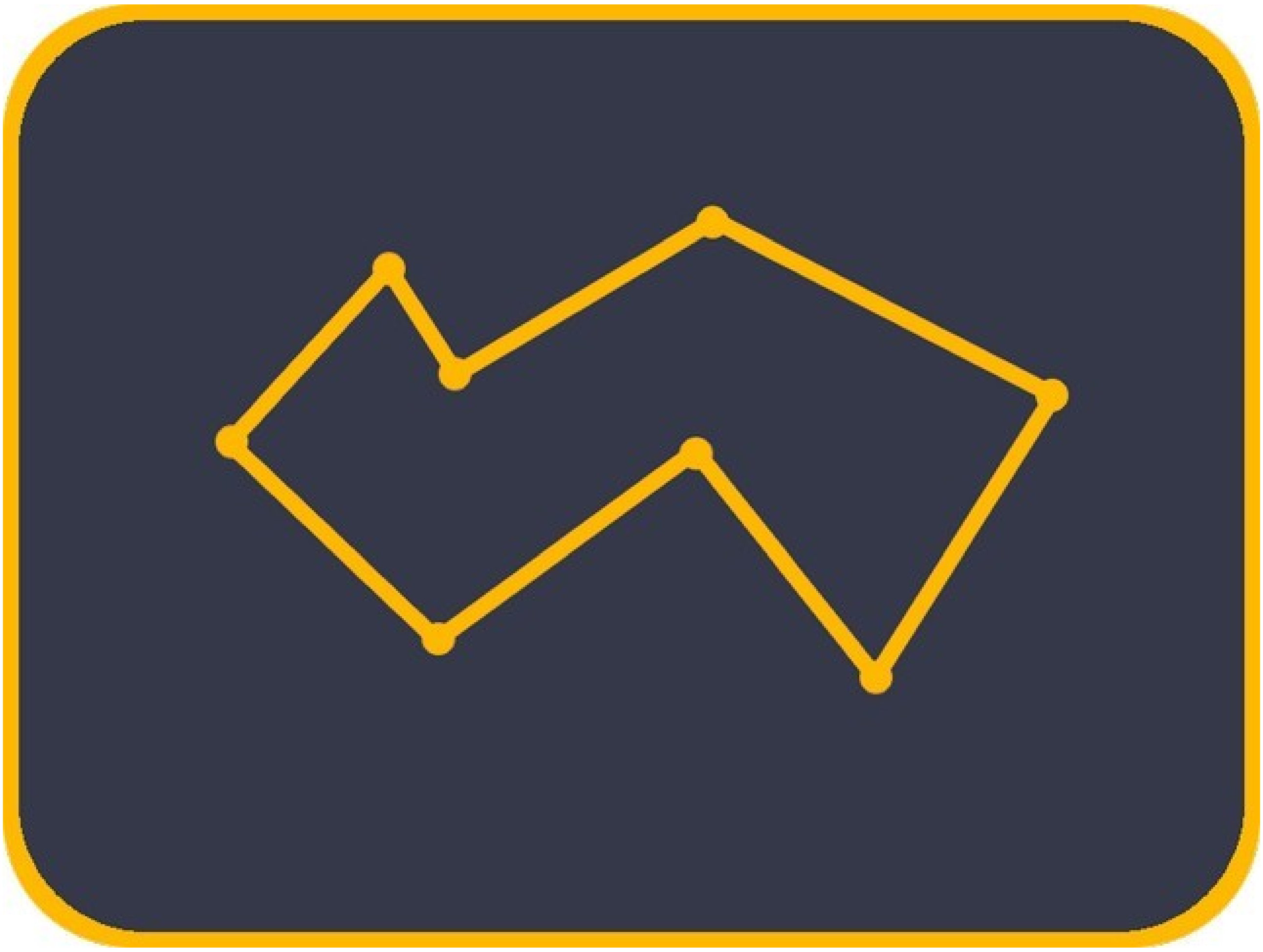
But it will be definitely more resource heavy. Instead of only 8 vertices, your computer will need to calculate all 29 of them each time it draws a frame.

With edges it's also a matter of how much you want to interconnect everything. You can either do more.

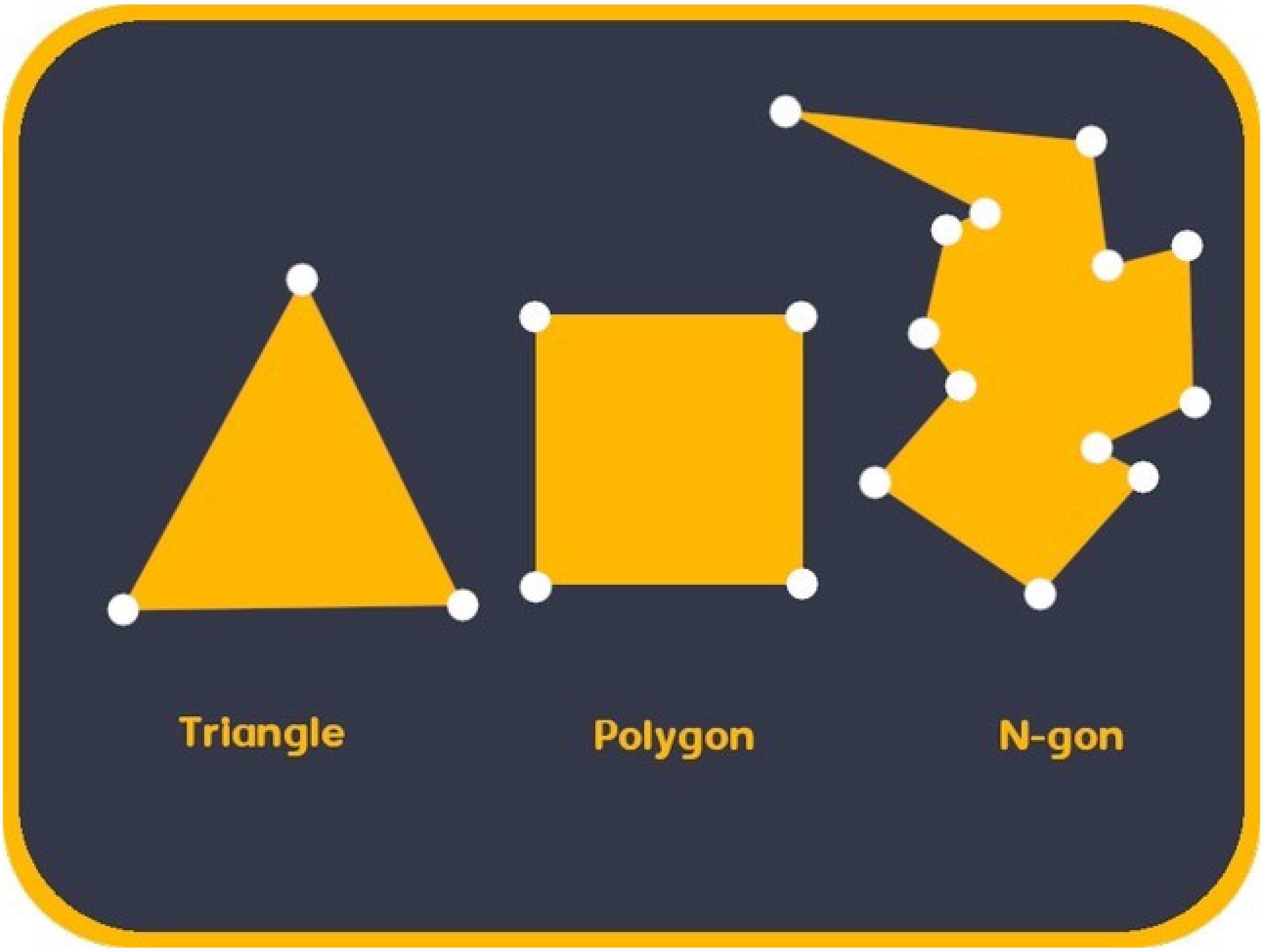


In the previous example we had only 3 polygons. Since I've connected point not that densely. Here we have 6 polygons. Since there are more edges dividing each square polygon into two triangle polygons.

Each polygon now is one forth less complex. Since instead of 4 vertices and 4 edges, now each one has only 3 vertices and 3 edges. But we doubled their mount. So this is not very good in terms of optimization.



From the other side you can make a one very complex polygon. This way there is only one of them. But this one, has 8 vertices and 8 edges. Making it twice as complex than any of the previous 3. And if we actually make a calculation. This seems like the most optimized option so far.

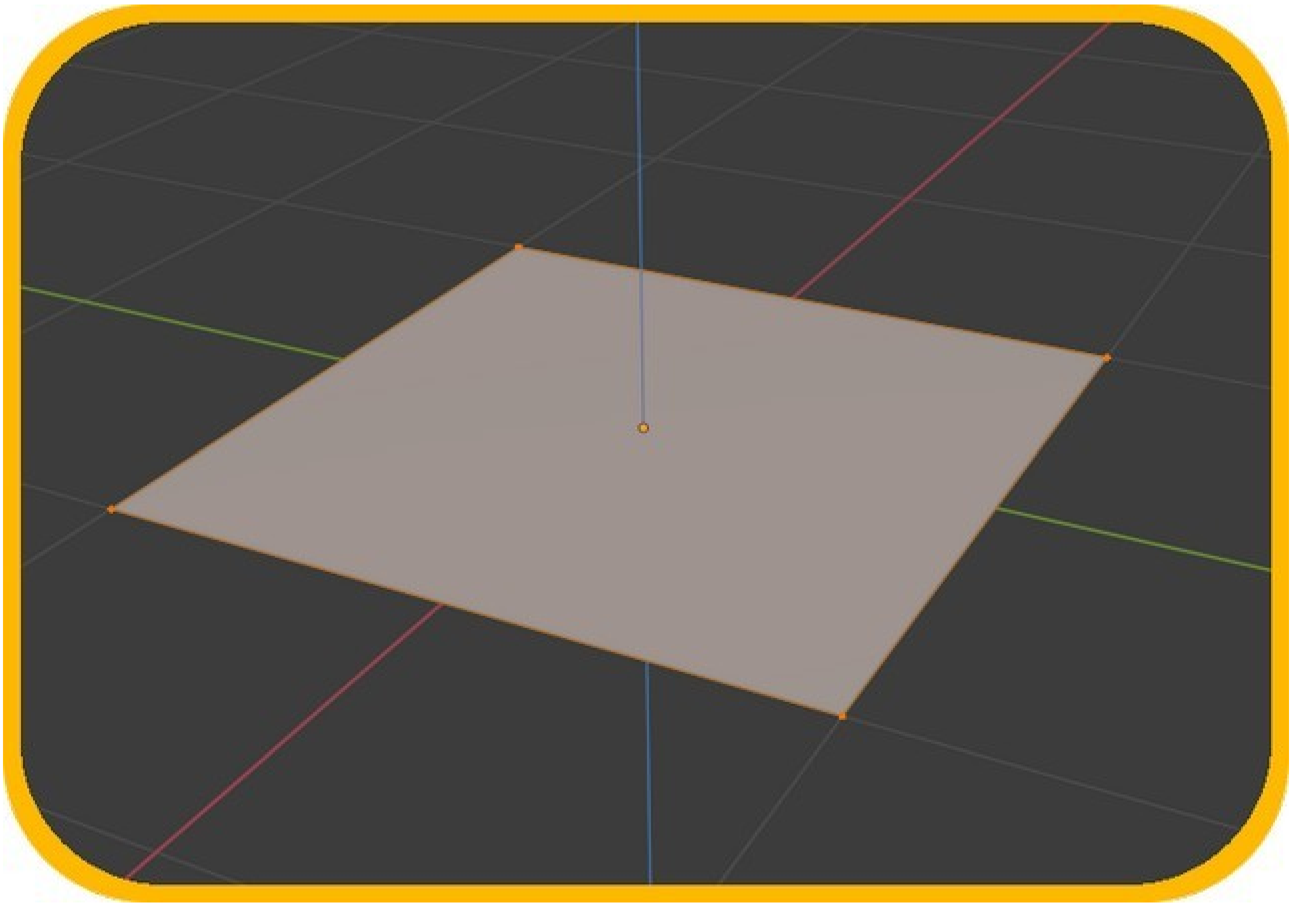


These are the 3 types of polygons:

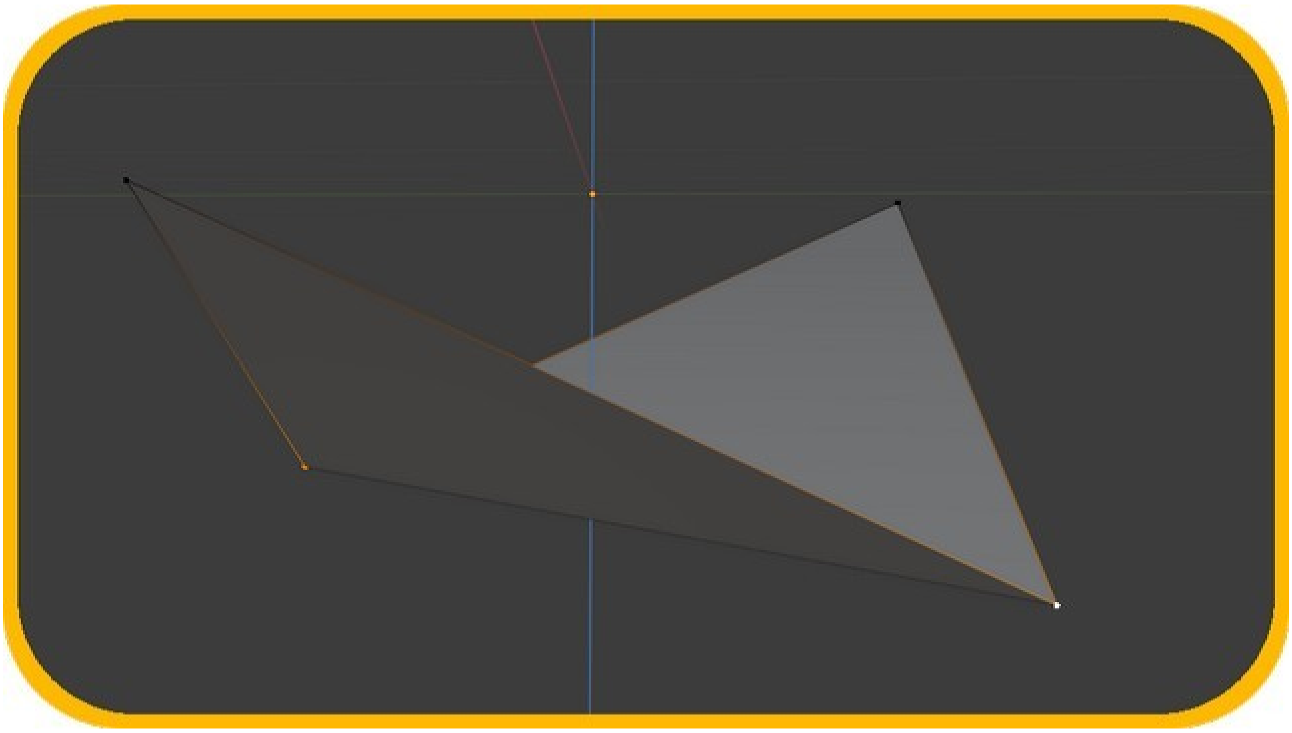
- **Triangle**. When it has 3 vertices.
- **Polygon**. With 4 vertices.
- **N-gon**. When it's any number larger than 4.

Problems with this in 3D

In 3D space you can have all 3 types of polygons. But there is a problem. Since we introduce another dimension, it might break the geometry. Let me demonstrate.



This is a normal 3D polygon. 4 vertices, 4 edges. A perfect square. But what's going to happen if I take 2 vertices. The ones that are diagonally opposite to each other. And move them either up or down. What will happen to the shape of this polygon?



This is what we get. It's should be one polygon. But what you actually see is two triangles. Similar thing will be with N-gons too.

Rendering algorithms are usually pretty dumb. They need the simplest types of shapes. And if you introduce a curvature, like with this polygon. It might not be able to render anything.

So a triangulation is always performed. Sometimes it's dynamic. Sometimes 2 separate algorithms are used. One for the rendering and the other for the collision in the game. And you sink through the world without understanding where is the bug.

Conclusion

Sometimes there is an obscure thing like this one. It's not obvious to the regular person. But significantly impacts how things will look. Or how optimized something will be.

I like digging deep into details like this. I don't like standing on the surface of things. And seeing only the top of the iceberg. I need the source code. I need the easter eggs. I need the nerdy stuff.

I'm glad that I know some already. But the quest never ends.

Happy Hacking!

Figuring Stuff Out

Sometimes you might have a problem that you need to solve but don’t know how yet. Sometimes you may choose a project that you don’t know yet how to make. This chapter will let you know how to Figure Stuff Out.

`lbry://@blenderdumbass:f/Figuring-Stuff-Out:1`

My current migraines made me realize that I lost a big part of my character. Currently as I type this article I have a strong headache. I took the pills. I gonna feel better shortly. So don't worry about me.

When you have migraines it's painful for you to figure stuff out. Hard for you to concentrate on a task. Things like programming or 3D modeling becomes hard. Even writing this article is hard. Since I need to concentrate on making meaningful sentences.

When I had my old Blender Dumbass channel I had a hidden philosophy on it. A philosophy I would like to come back to on this channel as well.

I would start recording a video without knowing much about the thing I want to make a tutorial about. I may try it ones before if it's really hard. But most of the times I would just go straight to it without knowing much.

And the video, apart from being a super-cut of hilariousity, would also be a showcase of how to figure out stuff. I would sit there and do things until it works. I would usually start knowing nothing and finish with a working example.

Here is an example of such a video. It's when I tried modeling in K3D. A different Free Software modeling tool. I knew Blender so I had a head start. But I still had to figure stuff out. Here it is:

lbry://@BlenderDumbass:c/blender-3d-vs-k-3d:7

Compare it to other channels

I want to plug a buddy of mine. Tyler Zane Kelley. Or @OfficialZaney and his video where he plays my game.

lbry://@OfficialZaney:8/jump-limited-playing-a-game-made-in:b

I love this video. But I think it misses one crucial thing. He doesn't beat the game. He is not dedicated enough to go through the entire thing. So even though it's interesting in the beginning. It might make it less interesting to watch the whole channel after you watch it. Since you don't expect him to beat anything anymore.

To be honest. Now he is streaming game creation. And he is doing the whole thing during a stream. So you have more opportunities to see his brain actually come up with solutions. So **Follow Him** regardless of the example that I just gave you.

Figuring Out is Interesting for Audience

A lot of very good films are built on the concept of showing the audience, a person or a group of people tackle hard situation and go through them. Win the bad guys. Figure out the case. Win the court battle. Escape the prison.

Let's a take a very dark film, Schindler's List. It's about a guy who tries to save Jewish people from Nazis. But in a time when expressing that you like Jewish people may get you in trouble. And if they figure out that you are saving them, you will be executed.

A hard situation indeed. But he manages to shuffle people, lie just enough, pretend just enough, and be in the right time in the right places to actually succeed to save a lot of the Jews. This is what's interesting to watch in this film. The process.

People always learn. And showing them an example of success, an example of **how to get** successful, is what makes people engaged. This is why tutorials, even though boring, make people engaged. This is why challenges and rage games are so popular. People want to see **how** you do a thing.

Figuring Out and Free Software

I'm a person that likes figuring stuff out. Since for a couple of years I was doing it frequently on my channel. And I was doing a few other projects along side it.

I have software projects like VCStudio where I developed a whole graphical engine just for it. Because using Gtk was too easy.

I like Free Software mainly because I can dig into it and figure out how to do with it things that other people can't. Or that other people rely on proprietary software to do the same things.

I run into the same kind of argument with people. They need a specific function in a proprietary software. Something that they find had to do in Free Software. Because in Free Software it might take a few extra steps.

Like for example I learned that Unity has a special object for terrains. In Blender Game Engine I had to use simple mesh objects. For some people it's just more convenient to use a thing specially developed for the task and never think about it.

For me. This feels like an assault on my abilities. What do you mean there is a function that makes it easy? What skill will **I** show, using it? This is stupid...

Of course I can be very much a hypocrite in this. Since if I'm using Blender and not K3D. And If a free software application does something cool, I'm more then happy to showcase this. But then again.

I refuse to use EEVEE (the new real time rendering engine of Blender) in anything real. Since it's not making me wait rendering. I will say how it's about the quality of the image. But I just like to suffer a bit. I guess...

I'm Not Even Human

There is this movie that I did long time ago:

lbry://@VCS:7/Imnotevenhumanshortfilm:3

I know the animation is a bit shmackadush. And the scenes are not very well rendered. But let me get you through what I had to go through to make this film.

In the beginning I just wanted to make the characters. Make the locations and render them. I found a problem on my computer. Blender liked to crash exactly during the time of me saving the files. So the files would be 0 bites. Meaning I lost everything.

I figured out a painful solution to it. When ever I save. I save 2 copies. So if one of them crash. I have the other one to restore things from.

Then I had a problem with organization. Since I had a lot of characters and a lot of shots. And a lot of everything. I had to keep track of all of it. Know whether I do everything in time. And so on.

At first I would print out papers with a checklist for each character. So I would know the process. And so I could just do one thing after another. And thus be able to be more organizer.

Then I made a little program that handles it for me. It grew into today's VCStudio. But back then it was built on Python 2. Actually I have an archived python 2 version of the software here.

So this program took a large portion of the work from me. And I could care more about the directing part and less about the folder structure and checklists.

Then I hit a rendering issue. If I start rendering it would give me first 2 or 3 frames and then crash. So I had to restart rendering every 2 or 3 frames.

I tried rending on the background. Using the blender -b commands in the terminal. It gave me a frame or two above. Meaning it would crash after 4 or 5 frames instead of 2 or 3. But still it wasn't enough.

So then I made a script that would launch all frames separately. It would open blender. Render a given frame. Close blender. And open blender again. Render the next frame and so on.

This worked way better. But later I realized that some frames are still missing. It would crash on random frames. So they will never be rendered. And I had to fill the gaps manually later. Not good.

So then I modified the script. It would know the first frame. And the last frame. And it would check in the folder if any of the frames missing. And it would launch rendering for the missing ones in order.

So if blender crashes during a render. This frame would not be saved. And thus it would start rendering this same frame again. This worked so well that this feature is implemented in the current version of VCStudio too.

Conclusion

A lot of people when encountering problems like this would just give up. A lot of people when having a migraine would not type an article. And if they will, they wont survive till the end of it. The pill made it easier. But now I have a slight feeling of euphoria. So pardon me if I left tons of grammar mistakes.

The idea that I was trying to make with this article. Is that if you are having less tools. It's not an excuse for not trying something. Because you may figure out a way to make it work. You just have to have a right set of mind.

For those of you who want to make videos. Please. Show us, the audience, the process of Figuring Stuff Out.

Happy Hacking!

4D Polygons

One dimension is a simple line. Two dimensions is a square. Three dimensions is a cube. But what does a four dimensional hyper-cube looks like?

`lbry://@blenderdumbass:f/4D-Polygons:f`

In my article about Triangulation I touched briefly upon the idea of putting points using more then 3 axis. More then three dimensions. Doing 4D polygons. Or 5D polygons. Or N-D polygons (polygons with any number of dimensions). And rightfully so, people were totally confused.

From the point of math this makes total sense. If you can make an array of 2 numbers. You can make and array of 3 numbers. Or any number of numbers. So theoretically a 4D polygon is totally possible. A 4D triangle will be a polygon with 3 vertices. 3 points. Each of which have 4 numbers to specify their location. One for X, back and forth. One for Y, left to right. One for Z, up and down. And an extra one. But in what direction is this extra one goes?

You can't see 3D

Even though we are living in three dimensions. Each of our eyes only sees two dimensions. This is perceived as three dimensions since our brains are very clever at figuring things out. A camera will produce only a flat image of something. While playing a game or watching a movie, you will see what's appears to be 3D on a 2D screen. A flat surface.

I'm not saying that there is no 3D world inside the game. Or that a picture the camera took wasn't 3D before it was taken. I'm just saying that the medium is more limited then the reality. But since humans are smart, we can fill the blanks in our heads and can accurately guess a 3D shape of an object from a flat image.

It helps to have more then one point of view. Either rotating view, shifting view. Moving through the scene. Or simply, as with human vision, just having 2 eyes that are slightly apart from one another. It's similar to having 2 ears that makes you hear from which direction a given sound comes. Having two eyes can make it easier for you to see a shape of an object and judge it's distance.

In 3D movies they have 2 images shown simultaneously. Each one is made for only one of your eyes. And the 3D glasses is a filter, so only the right image comes to the corresponding eye. It tricks your brain to think as if you see 3D. But there is no 3D. Just two 2D images.

1D vision in 2D world

As we humans have flat vision of 3D object, so characters in 2D games probably see only 1D images. A strip of information. One line. If you had a game character in a 2D world. While you, the creator of the game, has a third dimension. You could raise or lower an object on the Z axis. The one that the game character doesn't even know exists. You can make the game character perceive that something had appeared and then vanished. But in reality, you just moved the object through the third axis.

If there is a 4th special dimension, we as 3D humans who see 2D images, will not be able to grasp what's going on. Things would appear, vanish, deform in weird ways. Similar surreal stuff would the 2D game character experience if you just rotate the object. For him it will be weirdly deforming.

Tesseract

A Tesseract which sounds scary, just refers to a model that mathematically describes a 4D cube. In 2D there is a square. Where all edges are equal and perpendicular. In 3D you have similar concept in a Cube. Where all polygons are squares that equal and perpendicular. With the Tesseract, since it's 4D. It's a bit more wild. I want you to imagine an object constructed out of cubes. But so they are all equal and perpendicular.

If you click on the link I gave there in the beginning of this chapter. And read about the Tesseract on Wikipedia. You will see an image. Like if it's just a basic 3D cube with a smaller 3D cube inside it. This is not a Tesseract. This is a 2D image of a 3D representation of a 4D Tesseract. I know mind blowing.

As the 2D character in the game sees 1D representation of his 2D world. As we see 2D representations of the 3D world. You can make a 3D representation of a 4D object. But there is a catch. When ever you take a photo, or look somewhere. You will see it from one perspective. Moving around the object will alter the 2D view of it.

To draw a 3D representation of Tesseract you need to place a 4D camera with a 4D perspective. And then render it into a 3D model. Moving the camera in the 4D space will distort the 3D representation. Rotating the Tesseract on the 4th axis will look like it's going through himself in the 3D representation. Which doesn't look like rotating. But the same is true with a cube on a turntable, shot from the side. It doesn't really look like rotating if you really think about it.

Text as a Metaphor for 4D

Text is a 1 dimensional medium. Every article like this is just a long string of characters. Some are weird, special characters like \n. This is a visual representation of a new line, used in programming. So even though you see text spreading on a 2D plane. It's actually just a 1D object.

Articles, Books, Text Adventure Games can be very expressive. They can hold information about a whole world of things. You can express a complex system. A world with a clear geometry. A story. An emotion. Everything through just text. Through just a one dimensional object.

When I do 3D modelling, I do it on a flat screen. But with enough tools to navigate around. Enough tools to understand the shape of an object without this shape being here, near me. Text is the tool to navigate through the world of story. Or through a complex map of emotions. Or though an academic knowledge. Math is text. Text with more words. Math is a tool that could be used to navigate beyond out world. For example into worlds with more dimensions.

Having a good set of tools, like a 3D representation of a 4D object, helps to make things easier to understand. It's like when people go through Text Adventure games and draw maps to understand the game's world better. The map is a 2D representation of a who knows how much D world inside a virtual existence. In which you can navigate only using text.

4D vertex is a point with 4 coordinates. Each for a different axis. One of which is imaginary. One of which does not exist in our 3D world. But you can write it. (0.1, 5.3, -9.4, 7.0). This is an interface of communication with the imaginary. 4 numbers as coordinates for a single point in space. From left to right move it 0.1. From front to back more it 5.3. From Top to Bottom move it -9.4. And also move it 7.0 on a 4th dimension. A dimension that requires imagination to exist.

5D vertex is a point with 5 coordinates. 6D vertex is a point with 6 coordinates. 7D vertex is a point with 7 coordinates. Math doesn't break. I can still express it with text. What breaks is your brain. Think about it. A 7D Tesseract. A 2D representation or a 3D representation of a 4D representation of a 5D representation of a 6D representation of a 7D object. It'll fry every-bodies br**y**an just to attempt at visualizing how it looks. But there is a picture that will match the description. It could be done.

Conclusion

Things like extra dimensions are a good exercise in thinking outside the box. Outside the norm. A good exercise in making your brain more flexible. If you would suddenly discover a 4th dimension. And how to navigate through it. In a short period of time, your brain will adapt and it will be the new norm. Scientists already claim that there are more then 3D in this world. Perhaps one extra dimension is time. And we are stretchy noodles. Since we exist and keep existing. Time passes and I'm still here, still typing. I'm a stretchy noodle for sure.

What a weird experience would be to rotate. And suddenly one spacial dimension becomes time. And time becomes one special dimension. The world would be unrecognizable. That's for sure.

Don't keep your self following the norm. It's boring. Use your freedom. Explore things. Challenge yourself. Stop looking at the world like it's the only way it can be. Ask yourself "What if... ?". Edit the source code.

Happy Hacking!

Zombies

Zombies – Humans with a terrible disease, usually very contagious, that is so scary to those who are not yet Zombies that killing Zombies is morally justified.

`lbry://@blenderdumbass:f/Zombies:8`

There are a lot of terrible things people do all the time in this rotten world of ours. And some of them could not be justified. Murder, murder of children. This stuff is real. This stuff is happening. And those people who do that are rightfully dismissed and ridiculed. Murderers are more often then not face consequences for their murders.

But it seems like human nature is to murder. Since humans battle themselves all the time to find reasons for death to occur. There is not a lot of moral issue when it comes to murdering somebody that's brain-dead and wants to attack you. Even if the Zombie in question is a child. Pulling the trigger seems to be socially acceptable.

What are Zombies?

Zombies are people. Usually either people with a disease what makes them hungry for blood. And makes them look all sick. In some stories Zombies are already dead. And some magic thing, or a scientific thing made them alive again. Meaning that it's like a human who was unconscious, but now back to consciousness with some neurological dysfunctions.

If it's okay to kill Zombies. It means that it's okay to kill sick people. People with a disease. Even if they are children. But is it?

Is it okay to kill sick people?

I think it's very self evident that the answer is - no. If it would be okay to kill sick people, people would go into hospitals and kill the patients. Because all of them are sick.

Or is it with Zombies that they are blood thirsty? They want to kill you and eat you. What about this kind of sick? Is it okay to kill a child that wants to murder you?

Couple of years ago I was shocked to watch one film. Spoilers ahead. It was The Good Son with Macaulay Culkin. He plays a psychopath child. His character is constantly terrorising other people. Makes a huge car accident. Tries to kill his own little sister. And tries to kill his own mother.

In the end of the film his own mother kill him brutally. Something I didn't expect. I thought that what would happen is that he will be caught and put into a special place where they treat children like this. But murdering him? This is insanity. Also the movie framed it as if it was okay. To which I totally disagree.

There are Lions and Tigers on this earth that are much more dangerous than Macaulay Culkin's character. But we are not just going and murdering them all. We found a way to coexist. We have lions and tigers in a zoo. Where people can observe them in peace.

Deranged, sick people. They are just sick. It's a disease. There are special people that can treat it. Or at least there are means that could be taken so this person will not hurt anyone. Killing a person is too much. Especially a child.

Shaun Of The Dead

One of the Zombie movies that I don't cringe while watching is Shaun Of The Dead by Edgar Wright. It's a comedy. A parody, if you will, on the Zombie genre. But with enough actual horror and tension. Since the director is legendary.

Spoilers. In the end of the film, best friend of our main character becomes a zombie. And they find a way to deal with it. Not by curing the Zombie infection. By cleverly utilizing zombies. His friends is sitting with a huge chain on his neck. And playing video-games.

A zombie is turned into somewhat of a human pet. Something that humans would do in a real life with a person who's mental ability is not very good. If you had a deranged brother, with an unusual level of aggression. You would probably restrict his freedom of movement just enough for him not to hurt other people. And let him live.

They could kill the dude in the film. But hey. It's still his buddy. And he still wants to play video-games with him. Why kill him if he is just sick?

Conclusion

I think people are always trying to find ways to get out their aggression. Murder is illegal. Murder is looked upon as terrible even if you only think of doing so. So you need scapegoats. Zombies, Rapists, Murderers, Super-Villains. Those who, if you kill, you will be a hero. And that will totally justify your desire to kill them.

I don't know if such a desire is good though. Since than you will justify other things. People who are not Zombies but acting like they are would be justified to be murdered. People who have a dysfunction in the wiring of their brain, could be justified to be murdered.

You could blame it on laziness. Not wanting to deal with a hard issue properly. This is why killing is preferred to arguing and curing. But I think it's deeper then that. Humans are violent in their nature. They are just looking for any justification possible to do violence.

Happy Hacking!

How To Make Your Followers Care?

Followers – People who follow. Does it makes sense then to tailor your publications to make the followers happy? Or a true follower will follow anything you put out?

lbry://@blenderdumbass:f/How-To-Make-Your-Followers-Care:c

I think I know how to make sure that the Followers of a given channel will be dedicated. Will comment very deep comments. Will engage, support and follow you everywhere you go.

In short: I know how to make your followers care.

Many Followers

For platforms like Evil Tube (YouTube) that are based on showing ads to people it's very important to have a huge following. People who will click. People who will watch the ad. Not necessarily those who will be engaged with the video it self.

In the early days of Evil Tube this was the basis for it's algorithm. And the authors were optimizing their thumbnails, tags and so on to be clickable. Not very much caring about the video it self to be watchable.

But we are not on YouTube anymore.

Even though they tried fixing it by introducing various changes into their algorithm. The "YouTube Creator" scene is still largely based on the idea of having lots of subscribers. But since the algorithm now takes more things into account, they now ask you to like, dislike, comment and so on.

This is fundamentally wrong. Since this will produce brainless number of non-caring people. And in such, with 100 Billion Subscribers only about 1 or 2 million people watch the video. Here is the PewDiePie problem for you.

The largest part of subscribers doesn't care.

Trying to be popular

One of the problems I think with "Trying to be popular", is that authors are doing things in order to grow an arbitrary number. Not in order to grow an amount of actual people who care.

There are a couple of strategies on making your channel / profile / account, popular. For example. There is the obvious. Trying to include everyone. Trying to cover relevant topics. Trying to fulfil everyone.

This may give your publication a good view count. Since if people care about something. They have a chance to find and click on your publication. They may leave a comment. And if you are persuading enough, they will Follow you.

But those people cared only about that one topic. And here we have a problem. Now you either need to base the entire channel around that one topic. Or... Either you risk loosing the follower. Or you gonna have a dead follower. A follower that doesn't care.

One more way of growing your following is by literal cheating. Some authors buy from subscriber firms to add tons of fake followers to a given channel. This boosts the channel up. Then the algorithm picks up the boost and boosts it even more. Making a large portion of the subscribers fake, non caring ones. And the rest are only maybe caring.

The answer is not to think about the followers

When you make a publication based on what you, personally want to see. Not trying to make a given amount of people happy. Then you have 2 things happening.

1. Your channel will be growing way slower. Some people will un-follow.

2. Those who will follow you and stay, will most likely really care.

   If you don't afraid of people leaving your channel. If you don't afraid of scaring them away. Then those who will stay are your true fans. People who care deeply about your stuff.

   Think about this as a filter net. You pass the followers through it. And those who pass, are the purest, most amazing motherfuckers imaginable. The ones who don't. Well. Fuck those assholes.

   Now think about every publication being a filter like this. Those who will survive through all of it. Those who will pass and pass and pass. Those people are the most valuable followers you can ever have.

   I know. There is a downside to it. It's way harder to grow a following like this. Since a lot of people will not follow you. Or will un-follow you after a short period of time. But over time. This will pay off significantly.

Pay offs

Think about being able to pull off any kind of ridiculous shit. Say what ever you want. Ask for what ever amount of donations. Be as uncomfortable as possible. But still having a dedicated group of followers that love each and every seconds of it.

Or course I'm making it a bit too strong here. I don't say that you have to be uncomfortable to make it happen. But you shouldn't be afraid to be uncomfortable. Since this is the filter you need.

You will be able to make so much with those few followers, that those channel with millions and bazillions of fakers, will look like a total joke in comparison.

Conclusion

If you have an idea that people don't like. Fuck those people.

Happy Hacking!

Promoting Malware

A lot of people are openly promoting all kinds of software online without a second thought about what kind of software is this that there are promoting.

`lbry://@blenderdumbass:f/promoting-malware:c`

We all know about the malware and other injustices of non-free, proprietary software. No ability to change things. Constant spying. Addictive design, so you will never escape the clause of the proprietor.

Just to illustrate some bad things with proprietary software, let's give you a few examples.

2017-12

> HP's proprietary operating system includes a proprietary keyboard driver with a key logger in it.

2017-02

> DRM-restricted files can be used to identify people browsing through Tor. The vulnerability exists only if you use Windows.

2020-03

> The Apple iOS version of Zoom is sending users' data to Facebook even if the user doesn't have a Facebook account. According to the article, Zoom and Facebook don't even mention this surveillance on their privacy

policy page, making this an obvious violation of people's privacy even in their own terms.

2019-09

Keeping track of who downloads a proprietary program is a form of surveillance. There is a proprietary program for adjusting a certain telescopic rifle sight. A US prosecutor has demanded the list of all the 10,000 or more people who have installed it.

With a free program there would not be a list of who has installed it.

2020-03

Roblox (among many other games) created anti-features which sucker children into utilizing third-party payment services without authorization.

2019-07

Resourceful children figured out how to empty their parents' bank account buying packs of

> special players for an Electronic Arts soccer game.
>
> The random element of these packs (also called “loot boxes”) makes the game strongly addictive, but the fact that players are pressured to spend more in order to get ahead of their competitors further qualifies it as predatory. Note that Belgium made these loot boxes illegal in 2018.
>
> The only good reason to have a copy of such a proprietary game is to study it for free software development.

More examples of such misbehaviour from proprietary software developers you can find here.

With Free Software. Having the freedom to change software to fit your needs. Even if somebody would implement some of these anti-features. Either you or somebody else could simply make a fork without the issue.

Proprietary software will stay the way it is. Since editing it, is most likely illegal. Not even saying how hard it would be to edit it without the source code.

So the developers can force any anti-feature upon you.

People realize this. This is why you are using Odysee. This is why you are either using or thinking of using GNU / Linux instead of Windows. This is why you ditched Facebook and other nasty dis-services.

But here is the catch. The 2 last examples I gave you are related to proprietary games. Games are a very big nasty. And yet. There is one thing I hear people say. Think about the following sentence very carefully, this is cringe-worthy at best.

I'm going to stop using Windows when all of my favorite proprietary games work on GNU / Linux.

Rephrasing it would sound like this. *I will stop one program from abusing me only if other programs would still be able to abuse me.*

This is like the best facepalm moment in existence.

Gaming Channels

All this talk about proprietary software in the beginning of this article was to get you into this next part. I think I know the root of the problem. And I think I know how to deal with it.

Have you heard any of the following names? Markiplier, Jacksepticeye, Pewdiepie?

Those are famous EvilTube (YouTube) gaming channels. People who based their career on playing video games, recording it. And uploading it to the internet.

Lately with the popularity of Live-streaming, those same channels started also playing games live. Making large sessions with lots of live viewers commenting to them in the live chat.

The difference between a video and a Live-Stream is that in a video everything is already done, edited and the only interaction you can do with is the comment section. With Live-streams. Your comments could be read by the author, during the stream. Making you, the viewer, a part of the experience.

Let's evaluate Gaming Channels

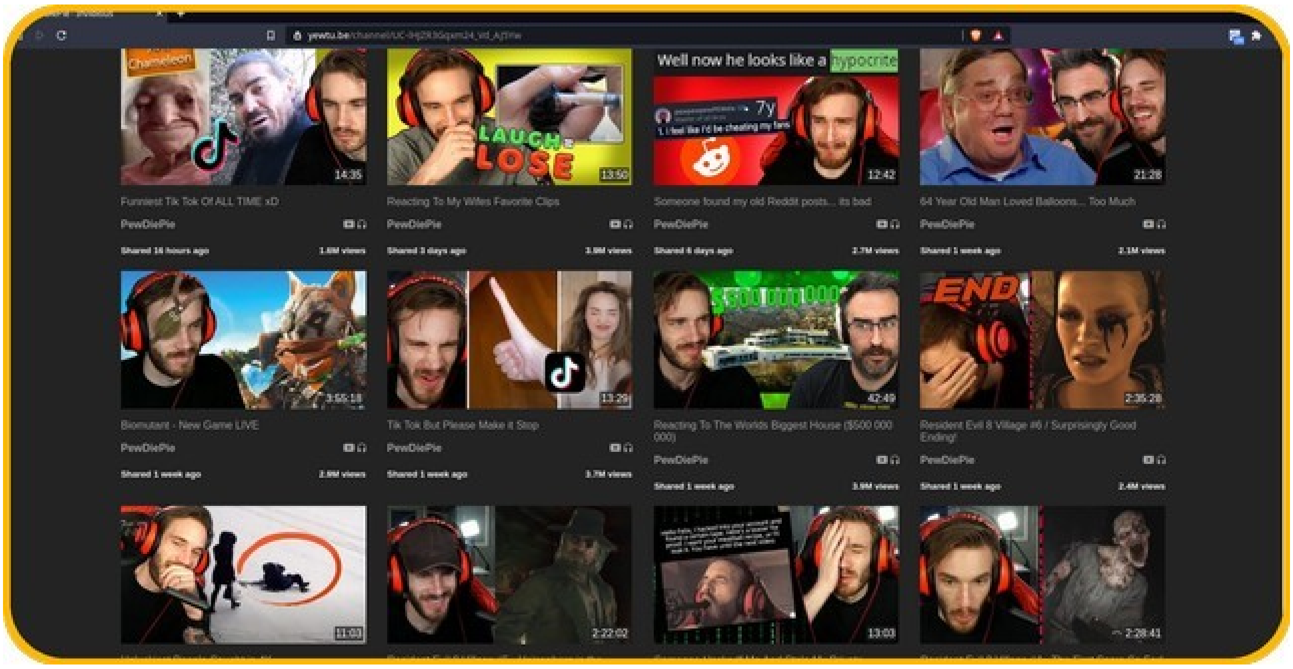
We already know a few things.

1. Proprietary games are as bad, or even worse than other proprietary malware.
2. A lot of people love to watch, other people play games.

But how much of those games, Markiplier, Pewdiepie, Jacksepticeye and others play are proprietary? How much they are contributing to the problem? And do they play anything Free?

Let's evaluate them.

Pewdiepie



Funniest Tik Tok Of ALL TIME xD
PewDiePie
Shared 16 hours ago 1.8M views

Reacting To My Wifes Favorite Clips
PewDiePie
Shared 3 days ago 3.8M views

Someone found my old Reddit posts... its bad
PewDiePie
Shared 6 days ago 2.7M views

64 Year Old Man Loved Balloons... Too Much
PewDiePie
Shared 1 week ago 2.1M views

Biomutant - New Game LIVE
PewDiePie
Shared 1 week ago 2.8M views

Tik Tok But Please Make it Stop
PewDiePie
Shared 1 week ago 3.7M views

Reacting To The Worlds Biggest House ($500 000 000)
PewDiePie
Shared 1 week ago 3.9M views

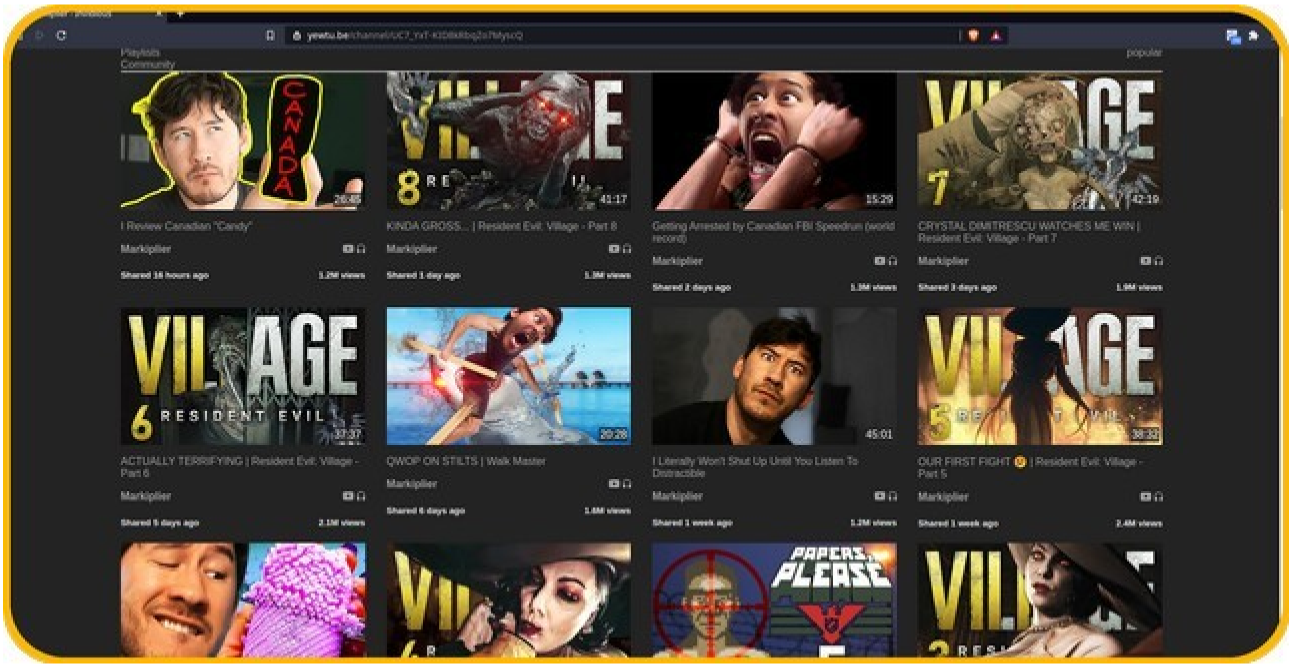
Resident Evil 8 Village #6 / Surprisingly Good Ending!
PewDiePie
Shared 1 week ago 2.4M views

I took this image from his Invidious channel. (it's a way to watch YouTube without going to youtube . com)

Software Promoted	Freedom	Known Issues	Score
TikTok	Proprietary	Heavy Spying	-1
Reddit	Proprietary	Lack Of Privacy	-1
Biomutant	Proprietary	Some versions use DRM	-1
Resident Evil 8	Proprietary	DRM	-1
YouTube	Proprietary	Spying	-1
Windows	Proprietary	Spying	-1

Only from this screenshot Pewds already gets a score of **-6**.

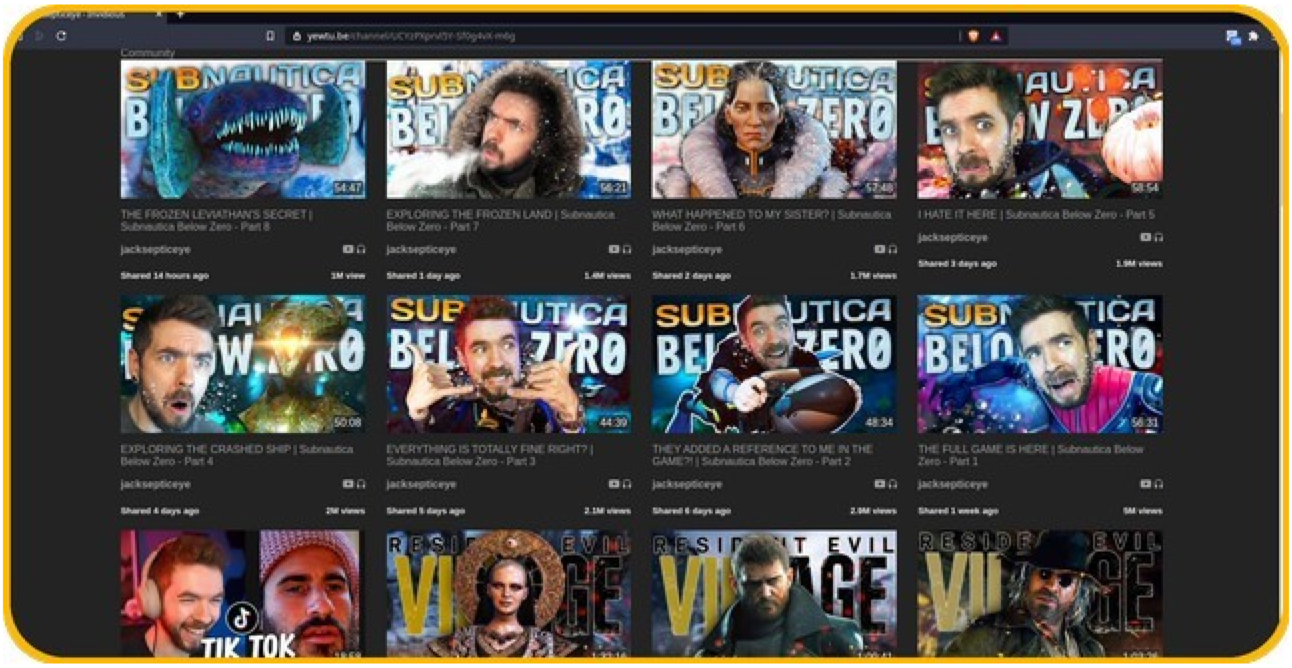
Markiplier


Playlists
Community
popular
I Review Canadian "Candy"
Markiplier
Shared 16 hours ago
1.2M views
KINDA GROSS... | Resident Evil: Village - Part 8
Markiplier
Shared 1 day ago
1.3M views
Getting Arrested by Canadian FBI Speedrun (world record)
Markiplier
Shared 2 days ago
1.3M views
CRYSTAL DIMITRESCU WATCHES ME WIN | Resident Evil: Village - Part 7
Markiplier
Shared 3 days ago
1.9M views
ACTUALLY TERRIFYING | Resident Evil: Village - Part 6
Markiplier
Shared 5 days ago
2.1M views
QWOP ON STILTS | Walk Master
Markiplier
Shared 6 days ago
1.6M views
I Literally Won't Shut Up Until You Listen To Distractible
Markiplier
Shared 1 week ago
1.2M views
OUR FIRST FIGHT | Resident Evil: Village - Part 5
Markiplier
Shared 1 week ago
2.4M views

Software Promoted	Freedom	Known Issues	Score
Resident Evil 8	Proprietary	DRM	-1
QWOP	HTML5 version MIT licensed	Google Analytics	+1
Papers Please	Proprietary	No knows Issues	-1
YouTube	Proprietary	Spying	-1
Windows	Proprietary	Spying	-1

Only from that screenshot Mark already gets a score of **-3**. Which is better than Pewds. But only because QWOP developer added an MIT license mention in the Javascript used in HTML5 version of the game. The game it self seems to be largely proprietary in other platforms. But this is debatable and has to be investigated. No Git repository or source code packs were found by me.

Jacksepticeye


Community
THE FROZEN LEVIATHAN'S SECRET | Subnautica Below Zero - Part 8
jacksepticeye
Shared 14 hours ago
1M view
EXPLORING THE FROZEN LAND | Subnautica Below Zero - Part 7
jacksepticeye
Shared 1 day ago
1.4M views
WHAT HAPPENED TO MY SISTER? | Subnautica Below Zero - Part 6
jacksepticeye
Shared 2 days ago
1.7M views
I HATE IT HERE | Subnautica Below Zero - Part 5
jacksepticeye
Shared 3 days ago
1.9M views
EXPLORING THE CRASHED SHIP | Subnautica Below Zero - Part 4
jacksepticeye
Shared 4 days ago
2M views
EVERYTHING IS TOTALLY FINE RIGHT? | Subnautica Below Zero - Part 3
jacksepticeye
Shared 5 days ago
2.1M views
THEY ADDED A REFERENCE TO ME IN THE GAME?! | Subnautica Below Zero - Part 2
jacksepticeye
Shared 6 days ago
2.0M views
THE FULL GAME IS HERE | Subnautica Below Zero - Part 1
jacksepticeye
Shared 1 week ago
5M views
TIK TOK

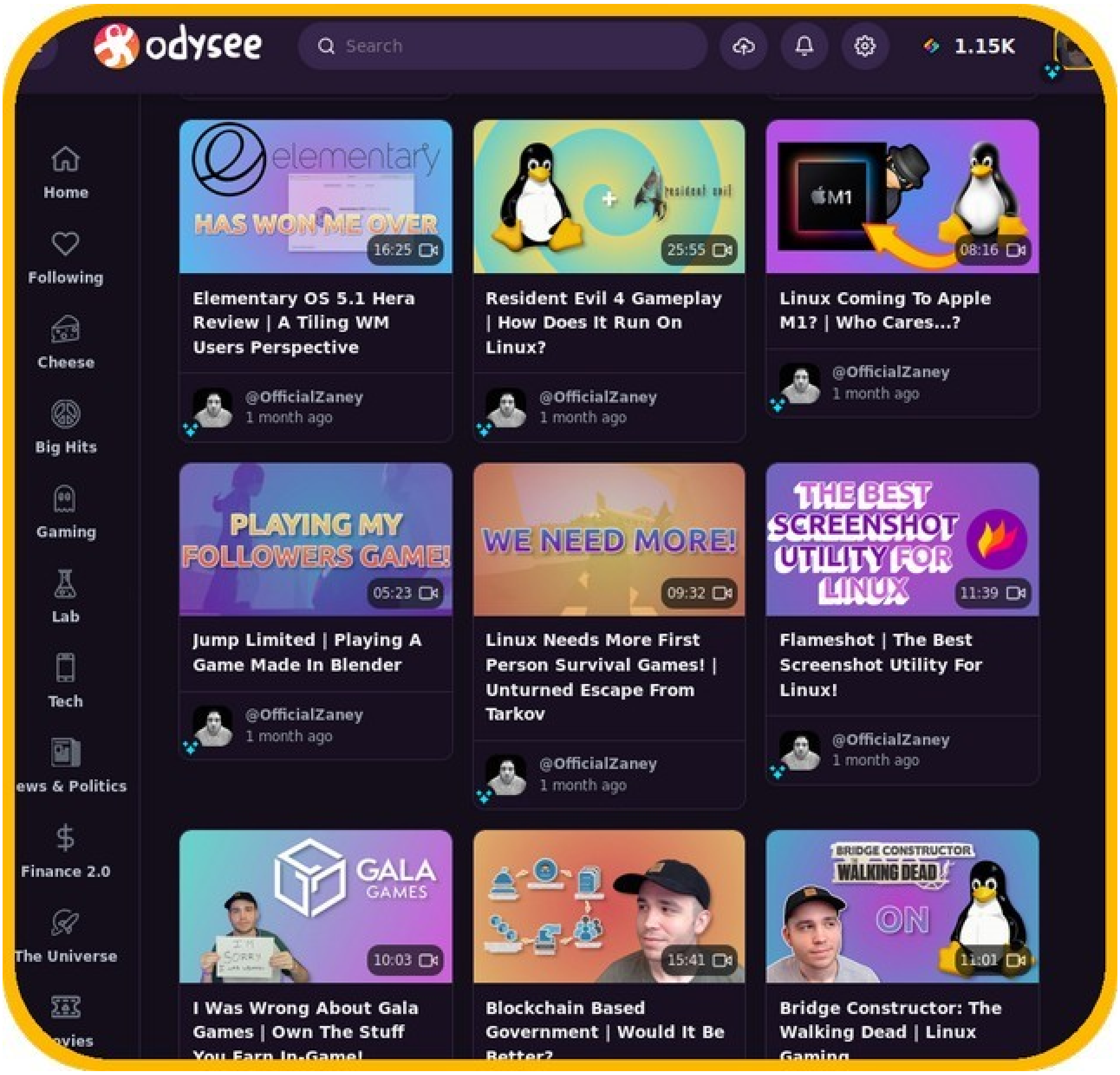
Software Promoted	Freedom	Known Issues	Score
Resident Evil 8	Proprietary	DRM	-1
Subnautica	Proprietary	DRM debated	-1
TikTok	Proprietary	Heavy Spying	-1
YouTube	Proprietary	Spying	-1
Windows	Proprietary	Spying	-1

Jack gets a score of **-5**

Alternatives

For one. Let's look at people who are here. On Odysee. A better platform. Free Software. And let's compare them to the big 3 channels of the Evil Tube.

@OfficialZaney



Elementary OS 5.1 Hera Review | A Tiling WM Users Perspective
@OfficialZaney
1 month ago

Resident Evil 4 Gameplay | How Does It Run On Linux?
@OfficialZaney
1 month ago

Linux Coming To Apple M1? | Who Cares...?
@OfficialZaney
1 month ago

Jump Limited | Playing A Game Made In Blender
@OfficialZaney
1 month ago

Linux Needs More First Person Survival Games! | Unturned Escape From Tarkov
@OfficialZaney
1 month ago

Flameshot | The Best Screenshot Utility For Linux!
@OfficialZaney
1 month ago

I Was Wrong About Gala Games | Own The Stuff You Earn In-Game!

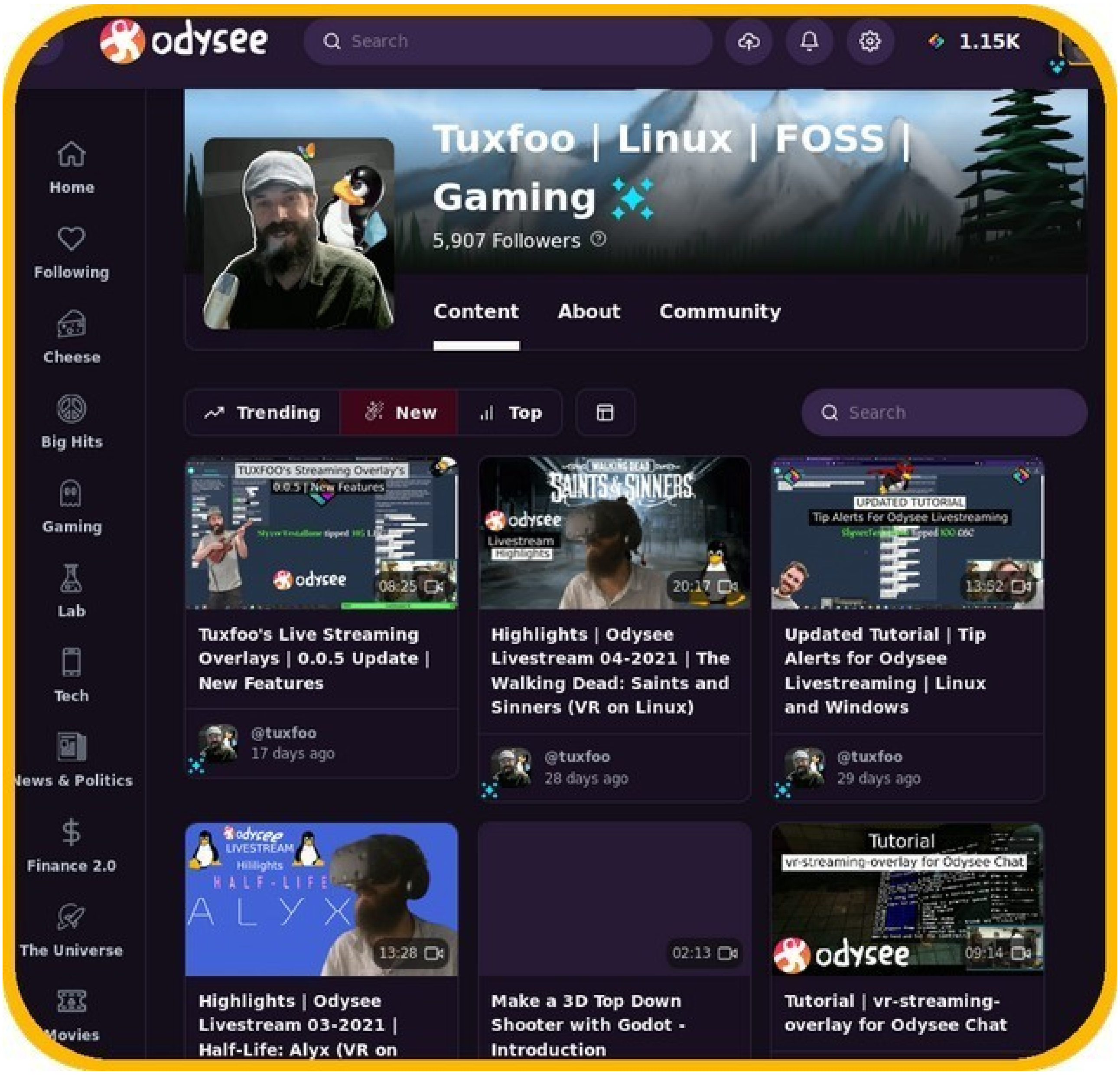
Blockchain Based Government | Would It Be Better?

Bridge Constructor: The Walking Dead | Linux Gaming

Software Promoted	Freedom	Known Issues	Score
GNU / Linux	GPL	Binary Blobs	+1
JUMP Limited	GPL	Needs Blender to Work	+1
Flameshot	GPL	none	+1
Resident Evil 4	Proprietary	DRM	-1
Bridge Constructor: The Walking Dead	Proprietary	no known issues	-1
Odysee	MIT	debatable	+1

From this screenshot alone. Zaney already has a score of **+2**. Which is significantly larger then others.

@tuxfoo



Tuxfoo | Linux | FOSS | Gaming

5,907 Followers

Tuxfoo's Live Streaming Overlays | 0.0.5 Update | New Features — @tuxfoo, 17 days ago, 08:25

Highlights | Odysee Livestream 04-2021 | The Walking Dead: Saints and Sinners (VR on Linux) — @tuxfoo, 28 days ago, 20:17

Updated Tutorial | Tip Alerts for Odysee Livestreaming | Linux and Windows — @tuxfoo, 29 days ago, 13:52

Highlights | Odysee Livestream 03-2021 | Half-Life: Alyx (VR on — 13:28

Make a 3D Top Down Shooter with Godot - Introduction — 02:13

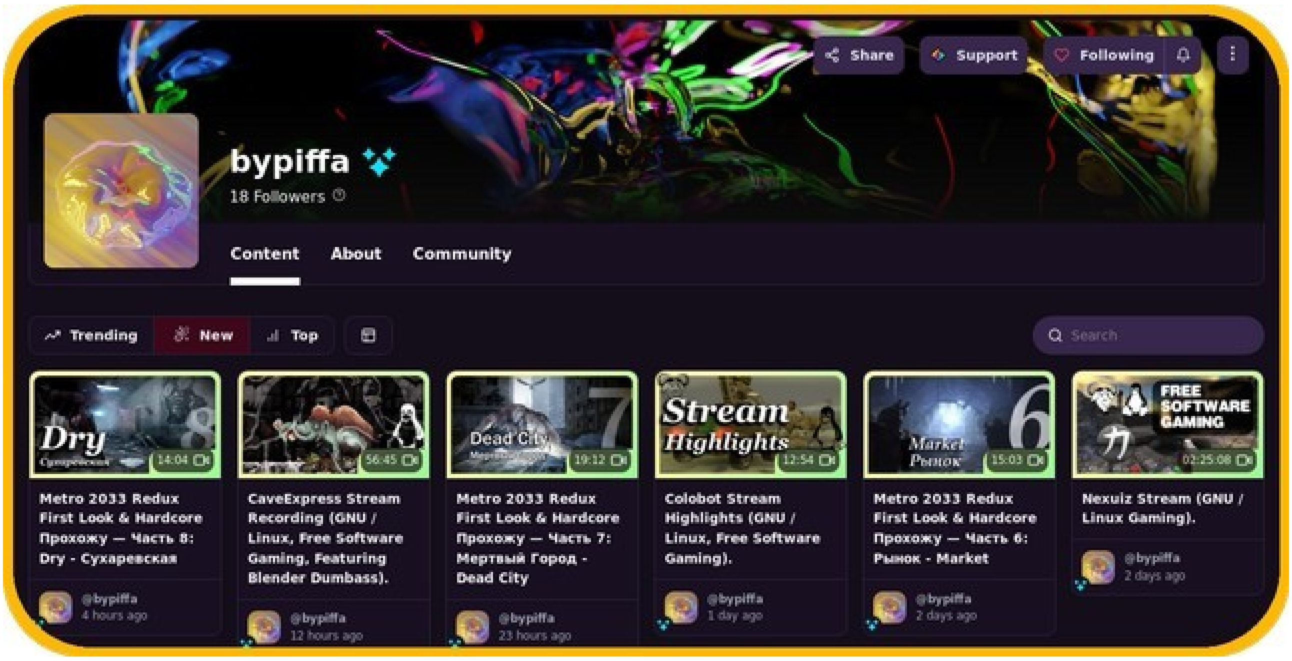
Tutorial | vr-streaming-overlay for Odysee Chat — 09:14

Software Promoted	Freedom	Known Issues	Score
GNU / Linux	GPL	Binary Blobs	+1

Software Promoted	Freedom	Known Issues	Score
Half Life Alyx	Proprietary	DRM	-1
The Walking Dead: Saints & Sinners	Proprietary	DRM	-1
NodeCG Odysee Bundles	Source Available	No license	0
Odysee	MIT	debatable	+1

Only from this screenshot Tuxfoo already gets a score of **0**. The main issue I had, is with his own program for Odysee LiveStream overlays. It's On Github but there is no license by this point in time. Maybe he will add one. I've opened an issue about it on his Github.

@bypiffa



bypiffa

18 Followers

Metro 2033 Redux First Look & Hardcore Прохожу — Часть 8: Dry - Сухаревская
@bypiffa 4 hours ago

CaveExpress Stream Recording (GNU / Linux, Free Software Gaming, Featuring Blender Dumbass).
@bypiffa 12 hours ago

Metro 2033 Redux First Look & Hardcore Прохожу — Часть 7: Мертвый Город - Dead City
@bypiffa 23 hours ago

Colobot Stream Highlights (GNU / Linux, Free Software Gaming).
@bypiffa 1 day ago

Metro 2033 Redux First Look & Hardcore Прохожу — Часть 6: Рынок - Market
@bypiffa 2 days ago

Nexuiz Stream (GNU / Linux Gaming).
@bypiffa 2 days ago

Software Promoted	Freedom	Known Issues	Score
GNU / Linux	GPL	Binary Blobs	+1
Metro 2033 Redux	Proprietary	no know issues	-1
CaveExpress	GPL	boring	+1
Colobot	GPL	none	+1
Nexuiz	GPL	none	+1
Odysee	MIT	debatable	+1

Only from this screenshot bypiffa already get a whopping **+4**. But he has only 18 Followers by that point. Which is :(

Conclusion

Since I'm experimenting with tables on Odysee. Let's make the conclusion as a table.

Channel	Score	Current Followers (by the time of writing)
@bypiffa	+4	18
@OfficialZaney	+2	629
@tuxfoo	0	5.9K
Markiplier	-3	29M
Jacksepticeye	-5	27M
Pewdiepie	-6	110M

Funny that I almost got a correlation here of followers vs freedom. I think it reinforces my other article about that with most channels, most followers are fake, inactive, idiots.

It's not science. I took very vague screenshots that illustrate my points. I've hidden a lot. And I was trying to promote @bypiffa since he is my brother and it's kind of sad that when he streams, the only one who types in the chat is me.

I would recommend you to follow all 3 of the Odysee authors I mentioned. Especially those who lack a good following. I want to see you in the live chat on my brother's stream. And Zaney's stream.

I want you to follow those who want to promote Freedom. Not those who promote what ever is popular.

Happy Hacking!

3D vs 2D Cinema

Is there a winner between Christopher Nolan and James Cameron? Is there a winner between 3D and 2D. And if there is a winner, who is it?

`lbry://@blenderdumbass:f/3D-vs-2D-Cinema:7`

Cinema is coming back. Even though companies are pushing on us their DRM infiltrated, proprietary, streaming services. You can read from many smart people of why they don't use Netflix. But for me. The worst thing about the streaming services is not DRM. Even though it's a huge problem.

Films usually are made for cinema. And other screen factors are considered only as an afterthought. Yes, an industry standard afterthought. But still an afterthought.

There was a video by Tom Cruise & the director Christopher McQuarrie about motion smoothness. They are talking about the release of Mission Impossible 6 for home viewing. And they are asking to turn off motion smoothing on the TV before watching the film.

Christopher:

> Video interpolation or Motion Smoothing is a digital effect on most high definition televisions. And it's intended to reduce motion blur in sporting events and other high definition programming.

Tom:

The unfortunate side effect is that it makes most movies look like they were shot on high speed video rather then film. This is sometimes referred to as the Soap Opera Effect.

Christopher:

Without a side by side compression many people can't quite put their finger on why the movie they are watching look strange.

Tom:

Most HD TVs come with this feature already on by default. And turning it off requires navigating a set of menus with interpolation often referred to by another band name.

Christopher:

If you own a modern High Definition television there is good chance you are not watching movies the way the film makers intended. And

> the ability to do so it's not simple for you to access.

So think about this. Tom Cruise and Christopher McQuarrie are going to all this trouble to explain one feature. Just because it messes up the feel of the movie. They designed the film to look and feel a certain way. And this Motion Smoothness totally breaks their vision.

For most filmmakers it's very important that people will watch the films in Cinema. Not on TV. On of a Phone. In a proper Cinema. Here is a take on this issue from legendary David Lynch. From this 30 seconds video.

> Now if you're playing the movie on a telephone, you will never in a trillion years, experience the film. You'll think you have experienced it. But you'll be cheated. It's a such a sadness, that you think you've seen a film on your fucking telephone. Get real.

James Cameron

James Cameron is a big supporter of 3D. His film Avatar made 3D movies the mainstream it is today.

This makes total sense that Avatar or any other James Cameron movie should be seen in Cinema. It's his own view. He gone even that far as to re-release Titanic and Terminator 2 in cinema again. So people who are just now learning about those movies could experience them properly.

There is an anecdotal story about James Cameron. Told from the perspective of another legendary film maker, Quentin Tarantino. Before Quentin got a chance of directing him self, he went to the premiere of James Cameron's Aliens. And he saw James conducting people to help them find the perfect spot in the cinema. He cares so much about the experience. That even a sit in a cinema matters for him. Not only the fact that you are watching it there.

James always was trying to invent new experience. For James it's about grandiose experimentation. In Aliens he experimented with shooting storyboards using toys. Instead of drawing them. To have motion pre-visualised as well. Today similar things are done on a computer. For example, Sam Raimi's Spiderman 2 was pre-visualized using Blender. A Software Libre program for 3D modeling and animation.

For Titanic James wanted to record real titanic on camera. His brother John David Cameron helped develop the submarine-robot-camera seen in the film.

This was an actual expedition. Actual submersion. Actual, real, titanic. The film Titanic was a clever way to ask for the money to fund this expedition. But since James Cameron cares so much about the experience. He had to make a good movie too.

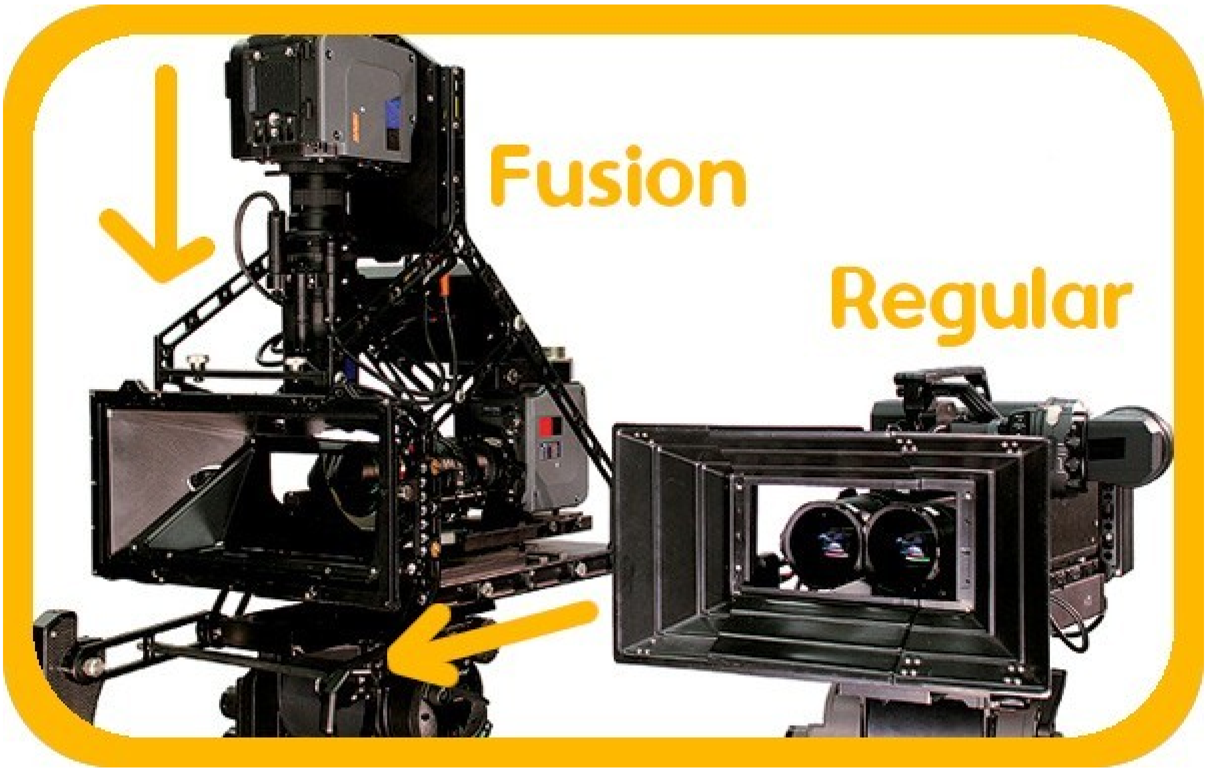
In the film you can already see James Cameron experimenting with 3D. The robot has 2 cameras. And the actor who is controlling the robot is wearing a virtual reality goggles. Actually just before Titanic James released a short film in 3D. It's called T2-3D: Battle Across Time. And it was shown as gimmicky 3D, theater only experience.

The obvious problem with shooting 3D on film is that camera lenses are usually larger then the eyes. For 3D to work, you need 2 separate images shown to the eyes separately. It mimics the way people actually see.

When you look at something. You think you are looking from one side. But actually you are looking from two different sides. Since you have 2 eyes. And they are not in the same exact location. This slight change in perspective helps your brain to figure out the shape and distance of an object. If you close one eye. The picture becomes 2D.

This is fine if you want to shoot aerial shots of landscapes and gimmicky movies. Just put two cameras very close together and record. But for a real, immersive experience. When you want audience to feel like they are in the same place with the characters. You have to put the two cameras at the same distance as the eyes. (For recording with 35mm lenses. More zoom requires more separation.)

For the next decade while developing Avatar, James would also develop a camera rig that was used in Avatar and lately in many other films. It's called the Fusion Rig.



On the right you can see a basic rig. 2 cameras with lenses very close together. On the left is the Fusion rig. The 2 cameras are perpendicular to each other. One looking straight on. And one is from an angle of 90 degrees. Recording a reflection. In the middle there is the most important peace. A special mirror that reflects exactly 50% of the light into one camera. And passes through the other 50% of the light for the second camera.

This Fusion rig allows recording 3D with two cameras closer together than what their size allows. And allows easy regulation of the distance between the eyes. Regulating how strong the effect of 3D will be.

Newer rigs also account for a view plane. It's an idea of when you focus on a point in 3D space your eyes rotate a little to meet at a certain place. Accounting for this reduces the pain from watching 3D movies even further.

Avatar spent a lot of time developing and testing the 3D system. Just for one movie length of experience. But other filmmakers took it wrongly. They thought that 3D is what's important. Not quality of experience. But simply the gimmick of seeing images three dimensional.

This made the 3D boom of the last decade. Studios pushing film makers to shoot 3D because Avatar was so successful. And since they don't have the experience or the budget. Most of those movies have a very bad 3D.

For example a very good film by a very good director that I saw unfortunately in 3D is Valerian: The City Of Thousand Planets. The CGI 3D looks amazing. But all of the live action shots are converted. Meaning they were shot in 2D. And later people on computers make the picture for the second eye by displacing the first image a little bit.

This makes this very drastic change of quality of 3D when you move from a live action shot to a CGI shot. Also I think Luc Besson didn't care much about 3D. Since in one shot the 3D effect made a huge ship look like a tiny toy. In 2D version it looks huge. Which brings me to ...

Christopher Nolan

Christopher Nolan is another master of cinema experience. But in the same time he is a very against 3D. As well as he is very against shooting on digital.

All Christopher Nolan films (by this moment) were 2D only and shot on film. But with all that said. The cinema experience of his films is so much more epic than watching the same films later at home.

I was sceptical of him making a good war film. Since I had no ability to go to cinema in 2014 to watch Interstellar. I was having this idea about Christopher Nolan, that all his films are just smart. How would he do a smart war film?

But then I bought a ticket to Dunkirk and went to see it. In 2D since it was the only option. I was blown away. First. It was a smart war film. But than second. It was extremely epic. And the 2D added a lot to it being so epic.

In Valerian the space ship looked small. Since you had a 3D image. And with it a sense of size. So the huge cinema screen basically canceled out. With Christopher Nolan movies everything looks epic. Since everything is huge. The eye of the character, when watching in cinema, is bigger than you.

After Dunkirk the only 3D film I watched in 3D was Alita: Battle Angle. Since it was produced by James Cameron and they knew what they were doing. All the other ones I saw, I purposefully chose to see them in 2D. So everything would look HUGE and EPIC. And it was worth it.

Conclusion

Experience matters. If the film was made to have the 3D experience as the key. When they spent decades perfecting it. Then the 3D of that film worth your time. For the other films. 2D makes everything look EPIC. Since things are HUGE.

Happy Hacking!

How I Made A Movie Using Only Free Software?

Do you remember a little promo page dedicated to Moria's Race. It's not my first movie project done with exclusively Free Software.

lbry://@blenderdumbass:f/How-I-Made-A-Movie-Using-Only-Free-Software:9

Free Software is epic. It gives people Freedoms they deserve. But there is a problem with it. People in the "industry" tent to avoid it for multiple reasons.

Old Artist Who Liked To Pay

When I was a kid I used to visit one very talented artist. He was an old man. With a big, grey, bushy beard. And he had like 20 children. From whom half are already married and have their own children. While the youngest one is about 7 years old.

His daily job was to produce a religious themed comic book. But secretly he was a Marvel fan. Not the MCU (Marvel Cinematic Universe) fan. He was a fan of actual Marvel Comics. He had a huge collection of comics. He had a dedicated room for all of it.

I went into that room ones. There were boxes, shelves and piles of comic books everywhere. It looked like an old studio of his. With a dedicated painting table, tilted just right, to work the easiest.

This was a little peace of printed heaven for me too. I'm a big sucker for good art myself. I tent not to care much if people assembled their games, for example, without any original art in it. And I love a huge design project, like Avatar.

Needless to say that he didn't work his daily job in that room. He had a new studio space in his main apartment. And his work flow was like this. He would draw the drawing with paper. Scan it. And then finalize and colorize it using Photoshop.

It was weird, because it was one of those people who already had both Blender and Gimp pre-installed on his main computer when I met him. But even though he totally could see how I can use these Free programs to do the same exact things he does. He would avoid them like fire.

He would claim that since this software is gratis, it's somehow not good enough. This wasn't even a discussion of Freedom, or access to source code. It was mainly a discussion of prise.

One time I wanted to prove him wrong by comparing the supported formats in his Photoshop exporter and Gimp exporter. Photoshop, as I remember, had just a few formats. While Gimp had a list bigger than the space of the screen. And also Photoshop had no Gimp's format in it. While Gimp had Photoshop's format in it. Clearly Gimp wins.

He said that Gimp has to include those formats only because Photoshop is mainstream. Photoshop has no obligation to be any good. Because more people use it. And it somehow proves that it's better. The logic was very flawed there.

I met a few other artists. Some of which made movies. And all of them had something nasty to say about Free Software. And all of them had something good to say about the proprietary garbage. Since most likely they payed real money for that garbage. And now it needs to be justified.

Sparkles Or Holiness (Worst Movie Studio)

When I was 15 I was already well trained in Blender. And I was totally convinced that Free Software is the way. I used to try to develop a movie project. Something that didn't work out yet. But I had some nice ideas. I might still develop them into something real.

Sinking in the fire
YA

I had this concept. This is a drawing of mine from when I was 14 or 15. The concept of a boy and a girl with a door between them, with a small window. And from one side there is a flooding. While from the other side there is fire. The scene is both tense, since they need to find a way to survive. And both emotional. Since they see each other.

I called this project **Sinking In The Fire**.

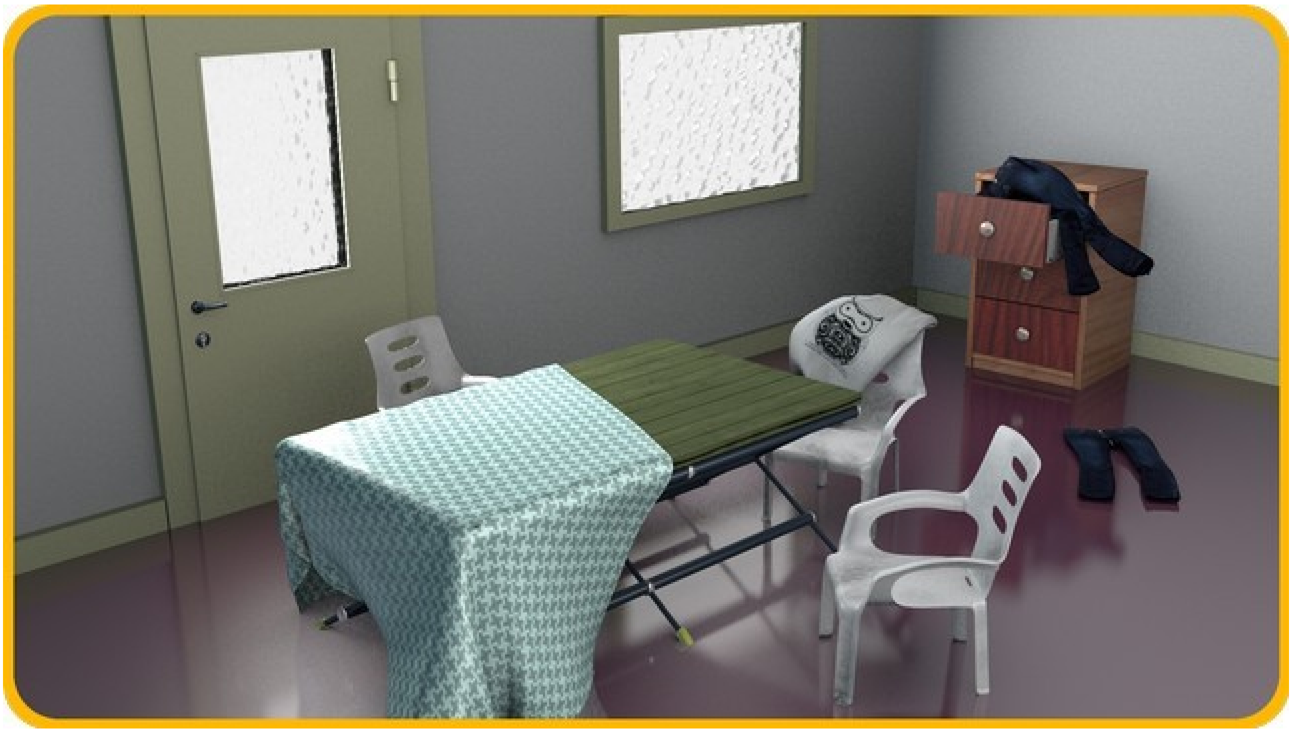
My idea was to do what ever I had no access to, in Blender. And shoot some kids on green screen to populate the CGI backgrounds. I wanted to do this for the majority of the film.

For example I had no home I can use to dress up for the scene. The way that will make sense to the plot thing I developed. So I had to construct a virtual home. And then populate it with real humans via a chromakey (green screen).

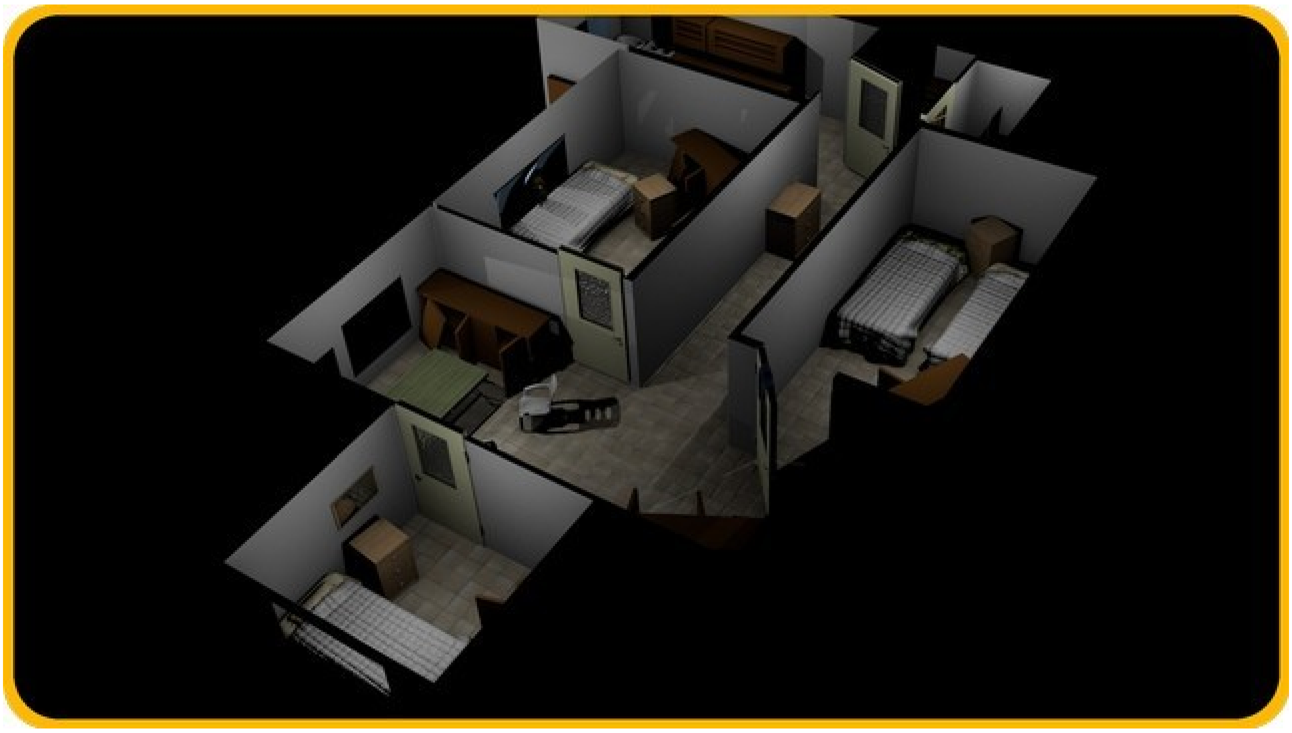
I didn't really need to cast anybody yet. And also this casting thing was and still is too complex to manage. So I would start with building their home in Blender. And later I would figure something out.

This is the plan of their house. I was building the story around this. I know, I could build a story around an actual location I had access to. But, then it means I would not design their house. This is not cool enough.

I would go on and design all kinds of assets. Like these doors. And I would make sure that each door is different. So the whole CGI place looks alive.



This is the assets kind of combined into a test scene. Now that they are finished I can go on and make their house populated.



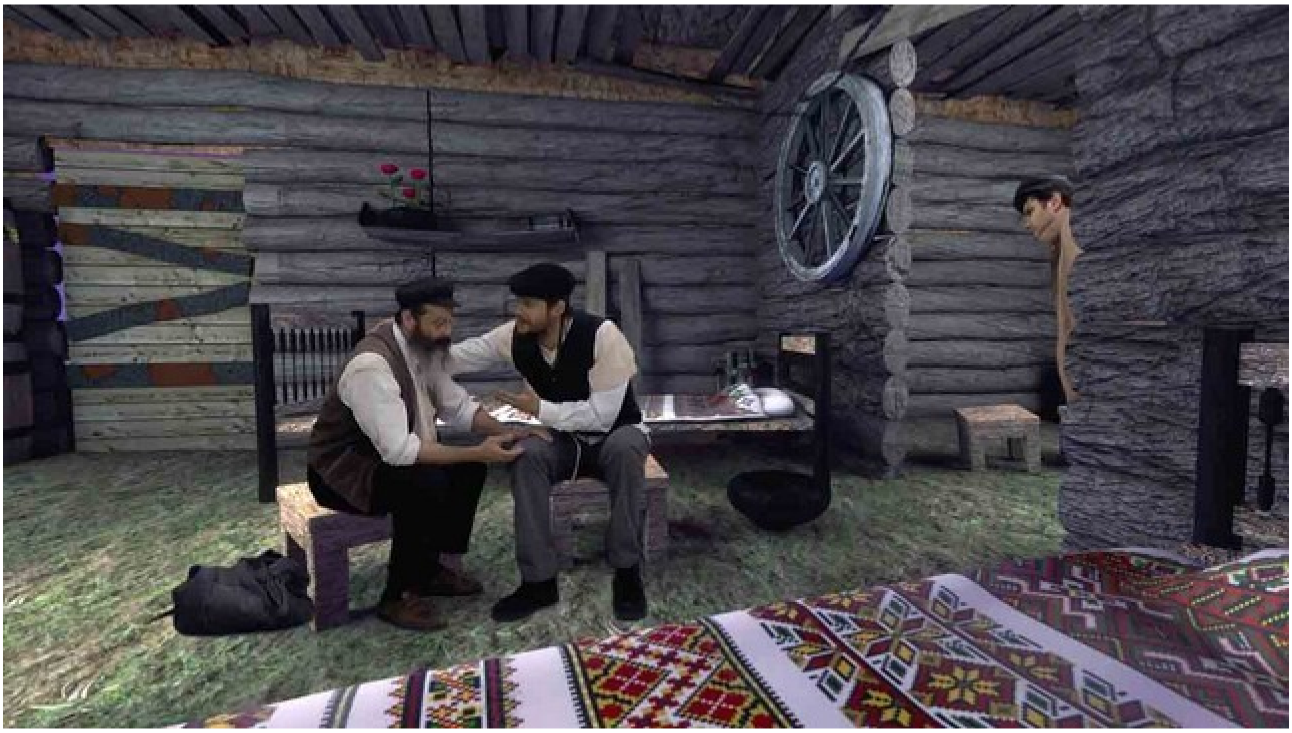
And so it was populated. And I could get out a few nice renders. I could show it to some people in hopes to convince them to work with me on this film.

Needless to say, this didn't work. But people from a local film studio found out about this project of mine. And offered me a job. At age 15, to remind you.

My job would be to do the same exact thing I wanted to pull of. But for a project that's actually happening. They had no budget. So they would get actors on a greed screen, they would act out a scene. And then they would simply put some background.

Usually they used stock assets. They had downloaded a library of them out of the Internet. And they would make a basic 3D Max scene with those assets. Render one frame, and put it as a background.

They knew I was trying to do something similar, but with much more quality. So they asked me to build backgrounds to some of their stuff. It was an old house thing.



This is what the final frame of the movie looks like. All the background here is custom. Not a single downloaded model. There was an outside thing also done by me. But unfortunately I can't find an image of it.

When I came and made something that looks better than what they are able to do, using Free and Gratis Software, all custom, and with great speed. The producer of the place started coming up with grandiose ideas. About how he would make a huge movie with an epic plot.

But in the same time, he wanted to cut my pay since I was a child. You can legally have a job in Isreal since 14. But in age 15, the legal minimum was about 15 sequels per hour. For the same kind of CGI work, an adult would be payed 200 sequels per hours. This was an insane cut of money.

They mixed me with mud since I was not 18 yet. I could try maybe doing my own stuff, since now I know I can. But then I can't still register a company. And nobody takes me seriously since I'm a child.

Rebelliousness and Youth Rights

I wanted to rebel. Both from using Proprietary garbage. And both from treating children like garbage. Something in me wanted to promote a kind of respect for children.

There is a movement of anti-ageism called Youth Rights. I just recently found out about it. My ideas back then were more or less the same.

Let's promote an equal treatment of adults and children. With equal right, equal opportunities. And so on. If you compare Ageism to something like

Sexism. The claims that women are payed less are actually debatable at best.

With Ageism. Kids are literally payed less, since legally you don't have to pay them a lot. If at least the minimum wage was the same for the kids. Also it's arguable that for a child doing the same job will be harder. So the minimum wage for children should be higher then for adults. Just using simple logic.

Of course there are problems with this approach. Like if you remove the 18+ thing completely, then what? How do we make sure kids don't die from alcohol and so on? Those things are interesting to talk about. I already made an article about it. Criticizing the whole 18+ system. And thinking about possible alternatives.

I thought making a movie project around this idea would be interesting. So I came up with a name for it "I'm Not Even Human". And started writing.

Writing Process

This was before I was able to develop my own writing software, so I used a simple text editor for it. I already written some stuff before. I had a few drafts of the infamous Sinking In The Fire movie. And I was during a test movie that ended up too big to pull of called Wrong Hate.

The renders from Wrong Hate though look kind a nice to this day. I still want to maybe make it one day.

With I'm Not Even Human I wanted to make a kind of dark film. With a lot of violence and sex references. Something that children are not allowed to see. My rebelliousness wanted to design a way that this kind of movie could pass through the censoring filter.

So instead of it being a movie with actors, it should be a cute looking cartoon. With characters nice and cute, so the parents would see it and think that it's fine to watch.

I know it was probably not the smartest idea ever. There was a bit of backlash from this movie. People yelled at me all kinds of profanities. But it was a cool idea non the less.

I turned on a full Quentin Tarantino mode. And wrote a film about a child that avenges the death of his pedophile. And they talk about drugs, sex and rock'n roll using the coolest, cursiest language possible.

I went so far into the profane darkness of the movie that I think I beat Lars Von Trier with this first draft.

It literally had a scene where Pito, the pedophile character argues with the main child character about taking Cocaine or Concentrated Caffeine since the child is too tired. And they want to have sex later.

Imagine what you be if this ended up in the final movie. I think the movie would be so outrageous that it would become illegal. Even though I knew I couldn't show them having sex. I talked just a little too much about it in the first couple of drafts.

The idea with Pito was to illustrate a kind of crazy concept. In a world where everybody disrespect children to the point of where they are literal property. The only person who respects our main character is a pedophile. A person who has the most interest in treating kids equally to adults. But with a very dark twist.

Bootstrapping the movie

I thought I needed to make the movie in the sequence of the script. So for example the first scene is the spaceship crash. And we see a bunch of peaces of the spaceship fly randomly. So I have to build only this part in Blender, animate it, and I have a shot.

But then I realized that we have a problem. I need 3 characters to be able to do it. And they were 79th (the main kid without name) Pito (the pedophile) and Bill (the crazy robot).

So I built all 3 of them, designing a kind of process for building characters. And designing a rig for them. I didn't really need Pito that much in the first scene. But I needed his goggles. Because later, the goggles are going to be with 79th. And in the crash I want to see the goggles.

So I animated this scene, basically having only these assets. But then I had to start the movie. Actually showing the town and all that. So I had to develop a lot more assets.

About this time I started the Blender Dumbass Channel. And so I started documenting the process of things somewhat. But trying to avoid any kind of spoilers.

I spend a little too much on the background car thingy. It was an unimportant asset, but I thought I had to perfect everything about it. So I had multiple videos about only the rigging process of the car.

Here is a good example of such a video...

lbry://@BlenderDumbass:c/advanced-rigging-in-blender-part-1-lol:a

By the way. This video started the running joke of having the little girl as a Logo of the channel. I found an image of the "evil looking child" when I was editing the words "She did it to me". While recording I thought about Destiny. And how I had destiny to cut my hair.

But to blame it all on the evil child was the right kind of joke for me back then. I was rebellious. And showing an image of a child that invoked the right mix of respect, fear, and amazingness felt just right.

Later I realized that this image came from a very problematic source. Basically it was done by, as what I understand, a semi legal studio. It's legal to do what they do in a country where they are. But it might be illegal in the rest of the world.

Basically think about it as a modelling agency. They photograph little girls with all kind of sexy outfits. Never naked. All consensual, with parental agreements and other legal documents. All totally legal in their country. But clearly targeting pedophiles.

This is why this girl is no longer the logo of the channel. But doing the right kind of, rebellious joke. I put the face of Pito as the logo instead.

I don't really think that what this do should be illegal. I had written a deeper article about my thoughts on things like that. And also I think kids should be able to decide whether they want to be photographed like this or not. Since well, Respect the Children.

I just think I realized that rebelliousness in not the best way of convincing people. Instead a carefully worded discussion is the right way.

Organizer . py

Later I realized that making assets requires a lot to remember. While I want to have a checklist of what things to do in order. So I would not sit there and guess. Every time I finish one task, I want to look into some list and see what to do next.

So at first I just written down the process and printed it as a kind of paper checklist. I would start one for each character. And it made it very easy to work on the movie.

But later I realized that I can use a bit of smartness to make this even better. And I developed a little GTK application to solve all of my problems. This is the first ever video about the Organizer.

lbry://@BlenderDumbass:c/organization-py-suzanne-waze-in-blender:5

You can see in some place during the video I press "FORCE BUTTONS TO WORK" button. This was the first ever version of the program. And it had a very large amount of issues.

The organizer grew to solve all of the other problems I might encounter along the way. This was basically our own in studio software that make us the best possible production company. But having only one guy.

Now the Organizer is called VCStudio. You can trying using it yourself.

Voice

I was reaching the scene in the film where the epic conversation about drugs should begin. I was trying to find kids to voice it. But non of them could speak well enough in English. And a lot of parents that knew English didn't want the words I written to even be read by their children.

I would give them the script and they would try to pronounce "Cocaine" or "Pedophile" and the parents were like... No!

So I had to think outside of the box. And invent something to fix my problem. The first idea I had was to record myself talk, and using editing magic make it sound like a kid.

I made a few tests, tried researching the difference between the sound of an adult male and a child. But ultimately gave up on it. Since it was too hard and never produced reasonable results.

So then I made a decision that in some way made the movie much more interesting. I decided that 79th is mute. He can't talk what so ever. And with other kids, I would just avoid them saying words.

To find references of mute language was way easier than to simulate a child's voice. So I did that. Only the problem was, I couldn't meaningfully cut out the conversations of 79th and Bill (the robot) without effecting too much of the plot.

So I found the next big thing. I literally gave the child a device that reads his thoughts and produces a voice that says what the child wants to say. Similar to the computer of Stephen Hocking.

I used a basic text to speech generator to make all of the sounds that 79th does after a certain part in the film. And the robotness of those sounds made for some very interesting jokes.

Movies are not finished being written when the screenplay is done. This is why the writer is most likely on set, when the movie is filmed. The director might have an inspiration to change a thing. And the writer is needed to either rewrite something, or to consult with director on whether it will work.

This idea of cutting down most of the dialog, in my opinion, saved the film from being overly criticized as "promoting pedophilia". Think about making a privacy respecting software and being claimed as a "supporter of terrorists" since now they can use your program to plan acts of terror. A careful approach is always needed.

I can't write on the site "terrorists are using this software, so therefor it's secure". Even if it's true. This would be dumb as hell.

The movie is still not careful at all. But since I had to cut so much out of it. It's at least watchable. And hopefully next time I can make a very good film that communicates this even better.

Release

The movie was released on Evil Tube (YouTube) on June 1st 2018. And later re-released on Odysee. I think this is the right time to take a look at the film.

lbry://@VCS:7/Imnotevenhumanshortfilm:3

Next Project

I learned a lot while making this movie. I learned to be more patient. I learned not to push ideas with force, but rather formulate them in a way that the people may get my points instead of yelling at me.

Rebelliousness is good inside. But being Rebellious for the sake of it will not lead to much. You have to get a goal and then carefully craft your way towards it. Since the floor is lava. And just jumping at the goal would probably get you burned alive.

I'm currently developing a movie that will be "family friendly" but with the same exact Respect Children message. Instead of focusing on drugs, sex and other stuff, this time I will focus on driving cars.

This is one thing that parents may break the law quite often. Giving kids to actually drive their cars without having a drivers license. And it feels like a solid foundation for a Respect Children movie. Since even though it's happening, nobody gives a crap.

The new movie will be called Moria's Race. At the moment you can find a link to the Blender.Chat (Rocket.Chat) group about the film here. I will post the advancements of the film there.

Happy Hacking!

Programming ?

? gnimmargorP

lbry://@blenderdumbass:f/programming:6

Sometimes you want to program something cool. And sometimes you want to waste your time with pointless programming. Recreational programming. Maybe even, pardon me, sexual programming. It's a very self-observational type of programming.

I want to talk to you about programming. Yes. To you. Not to myself. And not to this weird, middle aged man sitting behind you. I want to talk to you. You specifically. Since if I would talk to myself. I would be talking to myself. I don't want to be talking to myself. I want to be talking to you.

Now you are probably asking me right now "But what's wrong with the middle aged man that's sitting behind me?". And I gonna answer. What's wrong with him? He is too creepy for my taste. It seems like he is going to steal my thoughts and program something with them. Hell no.

You knock the man out. Prepare a cup of hot and tasty roasted sweat. And we are going to literally dive into the world of programming.

The world of programming

Programming is an ocean in the Hello World. I had prepared you that we are going to dive in. So we are diving in. But be prepared for another thing. It's not water we are diving into. Programs run on computers and computers feel very sparkly when put into water. So the ocean of programming is done from a cup of hot and tasty roasted sweat.

When we learn to swim inside the Hello World we learn to brainfuck ourselves. It's a very crucial skill. Think about this brainfucking idea. Space is actually white. Unless you set on a dark theme in Emacs. And then the whitespace becomes black.

Black

Black? This is a clue. Bet 1 LBC that black is a clue to something. I know. Stay with me on this one. "Black" is half of the word "Blackmetal". And Blackmetal is Metal. Metal is what a computer case made of... Wait. I know. It has nothing to do with computer cases.

Metal is a type of rock music. And black metal is a type of rock music too. But in the same time black metal is type of metal. Which is it self a type or rock music. Black metal is both a type of rock music and a type of a type of rock music. Brainfuck. This is not what I'm looking for.

I got the clue. I'm following this trail. But I think that I'm going in circles. Circles? Maybe it has something to do with circles? I know!

This will sound like a conspiracy theory but stay with me on this one. Circles are drawn best with programs. Right. No human can draw a circle by hand that's better then a program could do. It's most likely the case, since, well, programs are made of on 1 and 0. And 0 is a circle. This is why programs do circles better.

Now, what else is a circle? A drum. A drum is a circle. What else is a circle? A nob on an electric guitar. Also think about this for a second. To tune a guitar you need to turn a nob. And turning - is moving in circles. Just like I do right now.

I got it! It's Rockstar.

Rock vs Religion

I still didn't pass my foobar-mitzva so I'm not very qualified when it comes to the Church of Emacs. But I think Rockstar is not used to write the holy scriptures. The system source code is written most likely in C. And all it's applications are written in E-Lisp.

In the church I'm from, we have do not have any programmers. I'm the only programmer in the whole church. But I'm not a saint. I sinned yesterday. Running a program I cannot program on.

Confusion

I think it's fine to be confused.

Happy Programming!

Just Casually Made A Password Managing Program

I needed a password manger so I made one.

lbry://@blenderdumbass:f/just-casually-made-a-password-managing-Program:a

We all have sensitive information stored somewhere, on some platform. Or ever worse with online banking and Odysee we have finances somewhere protected only by a mere login and password.

A lot of bad people are trying to steal those. A lot of malicious crackers are trying to get advantage of your stuff. So a good password is crucial. And also using the same password in more then 1 place is insane.

For the most part it's not an issue anymore since people have password managers now a days that do all this hard work for them. But I had a few issues with that.

My problems with password managers

Most of them are non-free. A lot of password managers that are advertised are non-free, proprietary garbage. I wont use such a software for what ever purpose. But especially to keep my passwords. Non-free software tends to be malware. So this is a big problem.

Too complex. Even though there are a few password managers that are Free Software out there, still they seem too complex. And still a bit too hidden for my taste. I wanted something very basic. So I could hack on it at any time.

I might not have this computer. I want to be able to simply plug a file and get my passwords out of it. With or without the password manager at hand. I might not be able to use this computer. I want to be able to hold the file with passwords on a thumb-stick, somewhere. And access it at any moment. Without relying on an internet based dis-service.

No UI. I want my password manager to be terminal based.

I just made myself one.

Funny how easy it is to make a password manager. I built one in less then 4 hours. Spending most of the time on it's logo in the ASCII art style.

```
##########################################################
#                                                        #
#                      .-..-. .-.                        #
#                      | || | | |                        #
#                      | | \ \| |                        #
#                  .-  | | _\  / _                       #
#       ____       | \_/ // \|_|/ \                      #
#      |  _ \       \___/ \_/'-'\_/                      #
#      | |_)|____                   ____                 #
#      |  __/    \.----.----.-----./    \.----.----.     #
#      | | |  /\  |  --|  --| | | |  /\  | -- |  \ |     #
#      | | |  __  |--  |--  |  .  |  \/  |  \ |  / |     #
#      '-' '-'  '-'----'----'-----'\____/'----'----'     #
#                                                        #
#                                                        #
#                                                        #
#  (C) J.Y.Amihud 2021            GNU GPL v3 or later    #
#                                                        #
##########################################################
```

J.Y.Password. This is the name of the program. It's 2 python3 files. 9.3KB in size total. You can get a copy of it at:

https://notabug.org/jyamihud/JYPassword

Just run the run.py in terminal and there you go.

Short Tutorial

It will tell you to "login". It will be your file creation zone. Or file opening zone. Just type either a name or a full URL of a file. It should be a .hcu file. (HCU stands for Hasher Coder Usable).

If you just type for example blenderdumbass it will make a file blenderdumbass.hcu in the same folder as the software. Later you can login into the same file typing blenderdumbass again. Or typing blenderdumbass.hcu. It will add .hcu if it's missing.

It will let you choose a password to encrypt this file with the HCU algorithm if you don't have any file yet. If you already have the file, just type in your actual password. Note: this password will protect all the other passwords. So be sure to type in a good one.

Then if the file doesn't exists yet. It will ask you if you want to create one. So tell him yes. And let's add some stuff.

As soon as you are logged into the file. You can type help to get all the commands available to you. You can list the sites you have saved logins to. You can add a new site using news. If you just type the name of the site it will present you with all logins it knows on that site. You can use newl to add a new login. Or change a password on an existing one. If you just type the login name you will get the password for it.

It also has an option to auto-generate passwords for you. So you can be safe.

Warning

I wrote it in 4 hours. Who knows how stupid I am. I'm Blender **Dumbass** after all. It's under GNU GPL v3. Meaning I have no responsibility if the software breaks, or if it's any good. See the license.

Also **try breaking it!** Maybe we can make some cool peace of software there if we gonna bombard it with pen-testing.

Happy Hacking!

Law 2.0 - Beyond Democracy

What if the legal code of the law was uploaded to a Git repository where anyone could find vulnerabilities and give a pull request?

`lbry://@blenderdumbass:f/law-2.0-Beyond-Democracy:d`

We live in a society. How ever joking this may sound. This is true. We do live in a society. A society filled with other people. People that should be able to live together. And not hurt each other by stupid mistake.

This is why we have law. And this is why we have law enforcement. Law is a set of guidelines to follow, so nobody gonna get hurt. Or at least to make the suffering minimal. Law enforcement is a force that make people who do not want to comply, comply. Without a man with a gun, there is no need to do what's told.

In democracies we select people who develop and chose laws. We select those who craft what is acceptable and what is not. We select people who judge the situation and makes law accordingly.

But I think Democracy is lacking technical wizardry we could borrow from other industries. Let's break up what I think Democracy is lacking and how it could be addressed.

Note: I'm not a political specialist. I'm a Blender Dumbass. So take all what I will present with a grain of salt. Actually take all what everybody has to present with a grain of salt. Let's begin.

Reasons

All laws originated somehow. Similar to code in a software, laws were designed to patch a vulnerability in some area of live. For example. Somebody would kill too much people. So the law was made to make killing illegal. People used to take too much that is not their own. So theft became illegal as well.

You look at most common laws, the reasons of them are self evident. It's illegal to kill. Since people's lives matter. And so on. But there comes a law or two when people are not sure about the reasons of a particular law.

In some countries it's illegal to smoke Cannabis while in others it's legal. Why? In some countries prostitution is illegal while in other it's legal. Why? When you look in a constitution, it only says what is not allowed, or what is allowed. Sometimes with some conditions. But rarely it gives an explanation of why.

In Jewish religious law... You may not believe me, but Blender Dumbass here used to learn to become a Rabbi. I quit since I wanted to be Free. And religious traditions were too much of a hustle. Anyway... In the Jewish law you have that book "Shulhan Aruh". It's a short book with rules. Similar to the constitution of most countries.

It was written twice. By two different Rabbis in two different centuries. The older one is a bit broader in explanation. The newer book is a concerned version of the older one. The older book is a rewriting of "Gemorah". An even older book. That was commentated and referenced through out all the other books.

If you actually look at all those books, they all filled with links to one another. Mentions of exact phrases, and sometimes even numbers of pages that now has to be preserved in new prints, to stay consistent with the rest of the books.

This is a pre-electrical, pre-computer era Wikipedia. If you don't get why a certain law is a law, you can find a reference to some page in a different book that explains the same thing with more detail. And if you still don't get it. You will have a reference to an even earlier book. Where you may find the original investigation of this idea.

Sometimes you will find a reasonable investigation. Like the "Maim Ahreinim" ritual of washing the tips of the fingers after food. If you look down the rabbit hole or references you will find that this law became a law since a lot of salty food was eaten 2 thousand years ago in Israel. And that people needed to be reminded to wash away the salt so they would not hurt their eyes and other organs later.

This tradition doesn't necessarily makes sense now, since we have a running water in our sinks and we have soap. And most people are disgust when leaving food pieces on their hands. So we wash our hands anyway. But for back then, it made sense. And the sense was clearly communicated.

In the real law you have to look for information of why the law the way it is separately from the constitution. There is no link. And this is a problem. Sometimes you can't even find the reason. And all you can do is interpret the reason from the text of the law it self.

Sometimes a law is in the constitution that goes against the scientific consensus. But since it's the law, people tent to question the scientific consensus and not the law. This is ain't right. The law should have the link to it's original investigation. Where the full reasoning of this particular law is documented.

So when you look for why in your country to can't smoke weed, you could click a link, or find a page that will explain you exactly why, who was hired to do this decision, and how this decision was made.

Why is it important? Because you, as a citizen have the right to know. And you have the right to judge the points made by those reasoning. Not the just the law it self. But the points that brought the law. What if a critical mistake was made on the part of the researchers upon which now there is a law?

You have to be able to point out exactly the mistake. To point exactly how you found it, show the reference or a link in the constitution. And in this way be able to show where you disagree with much more clarity. Rather then speculate for eternity on how the law came about in the first place.

Branches

In the world of Software there is a concept of branches. It's when to test an idea, a new feature, the developer is forking the software into a separate copy and working on this separate copy. And when the feature is finalized, this copy is merged back to the main branch.

With Free Software the idea of branches is even more developed. Everybody can make a branch on your software. And you can make branches on anybodies else software. And develop it into what ever direction. Then, if you want, you can present your changes. And they may end up in the main branch. Or in some cases. When the main branch's developer doesn't want your changes. You can just have multiple branches developed simultaneously. Sometimes exchanging ideas and peaces.

Usually in today's world some version control software is used to make this possible. For example the famous Git and services based on git such as NotABug, Github and others.

Imagine for a brief moment that the law was uploaded to a Git like system. And everybody could fork it into their own branch. Changing anything and later presenting a pull request. A request for given change to be implemented in the main branch.

We can think of it like this. The main branch of the law is the law of a given country. Anybody can get a copy, with all the reference links, of course. And judge it, make changes, and come up with new things. Removing some outdated laws. Making some new laws and so on.

This person who edited the law can create a pull request. Anybody can see what changes were made exactly. And the person will have to provide links to all studies and reasoning that lead to his changes.

As soon as the pull request is open, a formal voting starts. Where all citizens of the country can vote on either one of 3 things. Reject, Accept or Revision. Revision means that they didn't like some detail about the implementation. They can leave their notes. And if voted for a revision the person who pushed the law, will be free to address these issues. And when addressed a new vote could begin.

Also what's interesting to imagine is a formal issues page. Visible to all citizens. It's like a bug report place for law. Where everybody who has an issue can complaint. And anybody else can take action on the complaints and edit the law for the complaint to disappear. Then do a usual pull request for the voting.

Think about a court case where a sneaky lawyer found a loophole and saved some bastard from consequences of his evil actions. It's clearly a security vulnerability. A bug. This bug could be reported and fixed. So let's have an ability to do so.

The key factor here is that it's should be everybody's right to do so. At least with in a specific country. If I don't like that it's illegal here to smoke Cannabis, for example. I can either make a formal issue about it. Or I can edit the law will all the references and reasoning on why I think it should be legal. I make a pull request. And if the majority votes "Accept", by the next day it's legal to smoke Cannabis.

This could be amazing.

Conclusion

I know I may be a bit over-reaching with this ideas. But this is the point of such works. I present a ludicrous idea. And I take it completely seriously. So we could have a theoretical and even maybe practical discussion on the topic.

In my current opinion, I don't see any problem with this approach. It seems like the best thing ever.

Happy Hacking!

TENET is Real!

S A T O R
A R E P O
T E N E T
O P E R A
R O T A S

lbry://@blenderdumbass:f/tenet-is-real:b

Tenet a 2020 film by Christopher Nolan. A film that despite all the restrictions and lockdowns was still shown in cinema. A film that is made now but utilizes as many practical effects as possible. A film that is made now and is shot on physical film rather then on digital.

But despite all of it, Tenet seems not to resonate well with the audience. With two hundred million dollar budget, the movie grossed just above three hundred million. Yes it's a whole one hundred million about the budget. And I would love to have this money. But compared to other films by the same director, this one is a flop.

You can say a lot of things. Blame it on the pandemic. Blame it on something else. But fact stays fact. The movie grossed less then they were expecting to make. Some people pointed out that the movie made less because of the lacking of Quality. Characters are not as well defined as in other Christopher Nolan movies. The plot is too complex for the average viewer to understand. And so on.

I think what made this movie so floppy is that fact that the filmmaker is just too smart. And those kind of people have to dumb them selves down in order to communicate with the rest of us. After watching it a few times and digging around about the film, I can say for sure that this is the case. Christopher Nolan is just too freaking smart.

Plot of Tenet

Just in order to get to some of my other points let's look at the plot of Tenet. I gonna prepare you, I gonna spoil hell of a lot of the movie. So if you didn't see Tenet yet, go grab a ticket to the near by cinema. Because watching this on a TV screen is sacrilegious. This film was designed for cinema. Shot on 65mm and 70mm film that gives a cinema viewer the sharpest image even on the biggest of screens. This is nowhere near anything digital. If you haven't seen a Christopher Nolan movie in cinema, I suggest you to do so. But anyway...

The movie begins in an Opera where our protagonist is on a mission. We don't know anything about what is going on. His mission is to retrieve some metal thing. He is being captured by Ukrainians and tortured to death.

He wakes up on an English speaking ship with a new mission. The only thing he knows about it so far is that it has to do something with the word "Tenet". And that it's a password to access all kinds of things that will help him along the way.

He learns about a time reversion technology. A technology that takes an object or a person and flips it's entropy, so the object is now moving back in time. Relative to the rest of the world. If a person is flipped, they will experience as if the world is flipped instead. Relative perspectives.

This leads him to Andrei Sator. A Russian businessman who has some form of access to this reversion technology. In order to get closer to Sator, our protagonist uses his wife Kat. He uses a fake Francisco Goya painting, allegedly faked by a man named Arepo.

Looking for answers, Protagonist and his new partner, Neil, are lead by Kat to a Freeport located in a Oslo airport. This is a storage facility that does not tax items stored in there. They want to find clues about this reversion technology in there.

They make a huge airplane crash on the site, to be able to break into the Freeport's security to access Sator's things in order to find clues. They find a strange machine. As soon as they find it, two men in full body armor emerge from this machine. One attacks the Protagonist. The other runs away from Neil. Both men escape.

They realize that this machine is called a Turnstile. And it is a device that flips the time of the things that enter it. With a help of Kat our Protagonist finally meets Sator and tries to make a deal with him. That has something to do with Plutonium-241.

Sator makes a deal with the Protagonist to steal Plutonium out of a moving truck. They arrange a huge heist mission to steal it that involves very large vehicles. But what they ended up retrieving wasn't Plutonium. It was one of those metal things, that the Protagonist retrieved in the Opera in the beginning of the film.

It's when they attacked by the reversed Sator. Who retrieves an empty case from the Protagonist who was claiming to have put there the metal artefact. They are ambushed by Sator and his men, and the Protagonist is captured. Neil requests backup from some cavalry. That makes the Protagonist confused.

To interrogate the protagonist about the whereabouts of the artifact, Sator wounds Kat with an inverted bullet. Making a small inverted wound oh her, non-inverted body. The decision was made to invert Kat using one of the Sator's Turnstiles. They are helped by the Tenet Operatives. The cavalry that Neil requested.

Protagonist learns that Neil is a Tenet operative. And he knew about the inversion long before the Protagonist. But since saving Kat is not a Tenet Operatives mission. They are not given a Turnstile to invert her back, when the wound will get better.

So the decision is made to return to Oslo Airport at the time of the Plane Crash to access the same Turnstile they had found in the Freeport to invert all of them back.

While entering, the Protagonist is being attacked by non-inverted himself. So he escapes into the Turnstile to end up in a chase with a different Neil. Revealing that the man that emerged out the machine in the beginning is actually the Protagonist that was inverting himself back to the normal time.

After this, they realize that those metal artifacts are a part of a larger machine called the algorithm that was developed in future and if used, will reverse the whole world at ones, cancelling entropy of everything, and causing a total annihilation. The end of the world.

Also they realize that Sator is trying to end the world together with his own life at the time of the Opera operation. So the Tenet Operatives together with Kat and the Protagonist need to make one last operation involving multiple reversed soldiers and multiple non-reversed soldiers. To retrieve the algorithm from Sator's men, before Sator will die, to save the world from ending.

I know it's a bit complex. But what did you expect from a Christopher Nolan movie? He made Memento, The Prestige, Interstellar and Dunkirk, a smart war film. His plots are famous to be hard to understand. And this is why I love him.

The Protagonist

There was a lot of criticism that the protagonist of this film is lacking character. His name in the film and in the credits is literally "Protagonist". And he him self says "I'm the protagonist" as a kind of self aware joke of the film.

There was a study of this film being an exercise in writing. An attempt to answer a question of whether it's possible to write a movie with the protagonist being a blank slate. A character without a character. A movie where we follow a man on a mission. And that's it.

He already experimented with this in his previous film Dunkirk. Where there is no scene of the soldiers sitting around and talking about life, or remembering the days before the war. The entire movie from the beginning till the end is just the war with no name people struggling to survive.

In Tenet we meet the Protagonist during the mission. We end with him in the mission. And the whole movie is just a one long mission. This is strange because we don't feel like the protagonist is any particular person. But it gives us an ability to reflect anyone on him.

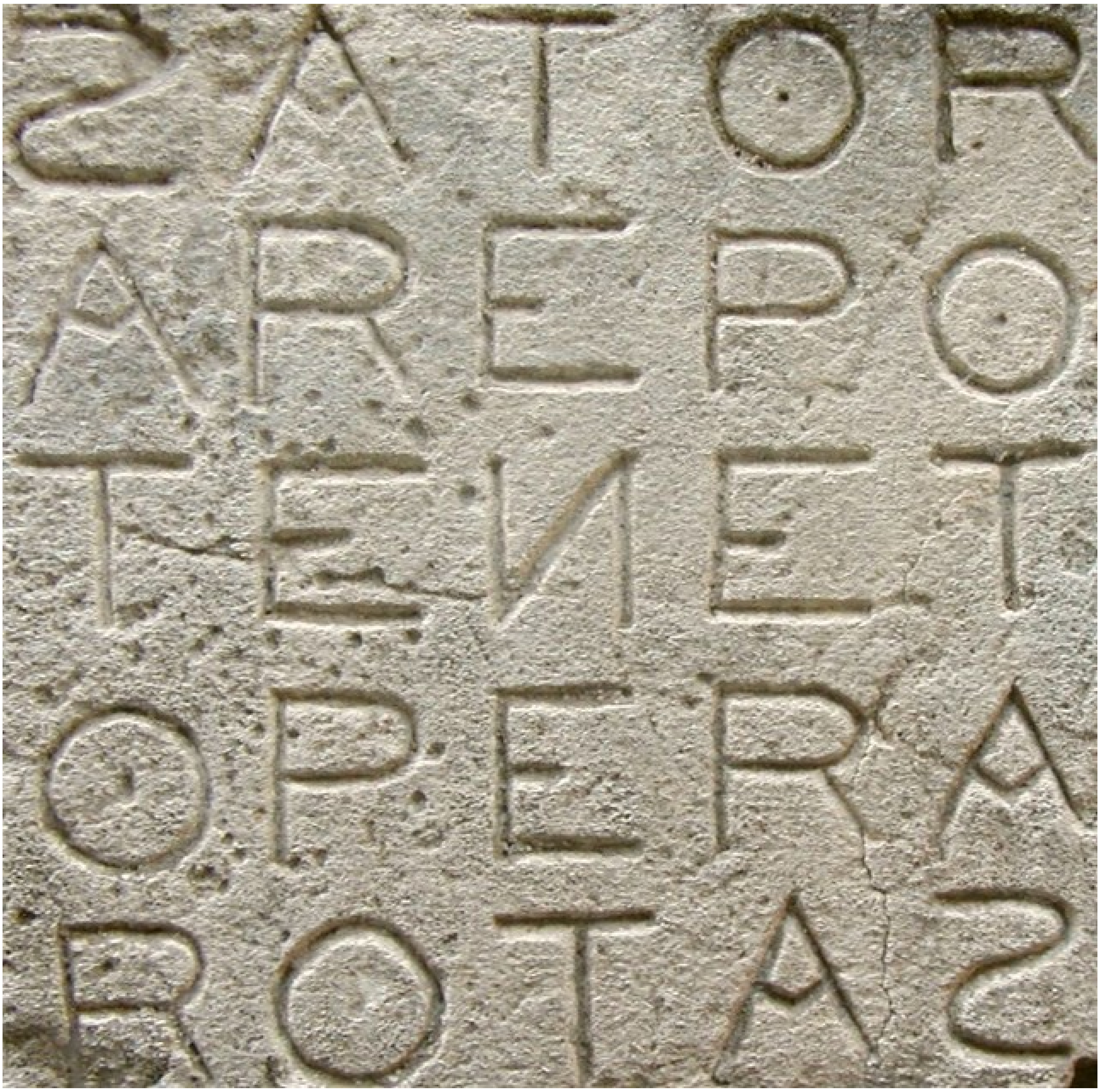
Since he has no traits, anybody you know can be him. You can be him. And so this is suddenly you, going through the mission. At least this is what I want to believe. Maybe Christopher Nolan was trying to sabotage the movie intentionally.

There is a plot hole in the film that is just glaring with obviousness. If Christopher Nolan was trying to sabotage the film, it will make sense why this plot hole stayed in the movie. With his attention to details, he must've known that this plot hole exists.

When Protagonist meets himself in the Oslo Airport. They have a fight and he is trying to shoot himself in the head. Knowing that it's him in the past. Effectively trying to kill himself.

This is a one issue I have with this film that I can't stop thinking about. But maybe I'm just too stupid to understand it. Maybe Christopher Nolan knows the answer to the mystery of that scene. Or maybe this is one of the those spinning top moments, that were designed to never make you stop thinking. Maybe this plot hole is intentional. Maybe Christopher Nolar experimented with a weird way of making people go back to see the movie again and again, trying to answer a question that is unanswerable.

Sator Square



A Sator Square is a two-dimensional word square found in multiple places in the world. It's believed to have some religious origins. And it's very prominent in the ancient world. It's a square containing the words Sator, Arepo, Tenet, Opera, Rotas. That approximately mean Founder, A name of a Person, Preserve, Opera and Rotate or Invert.

This is no coincidence that Andrei Sator is a businessman, a form of Founder. The painting was faked by Arepo, a name of a person. The preservers of the world are the Tenet Operatives. The movie has an Opera scene. And it's all based on Inversion of time.

Maybe inversion of time is real and Tenet actually exists. And the future is trying to communicate with us through the past by putting those Sator Squares everywhere. Or maybe Christopher Nolan is too smart for his own good.

The Algorithm

This is fascinating that the evil machine in the movie is called the Algorithm. And the one who wants to ruin the world with it is an evil businessman.

Christoper Nolan shot all his movies on film exclusively. Trying to avoid CGI as much as possible. Going for CGI only as the last resort or as a tool to enhance practical effects.

Christoper Nolan is a filmmaker who designs the movie with a cinema in mind. He is a big proponent of IMAX screens with IMAX 70mm film used to shoot and screen the movie. Since this gives the audience the best possible experience. The sharpest kind of image. The loudest and the fullest possible sound.

Something that's against Christopher Nolan's vision is the infiltration of new technologies such as 3D cinema, Digital Photography and Streaming Services.

He had broken the long term relationship with Warner Bros. in 2021 because of their decision to stream all their new films simultaneously as showing them in cinemas. People are not gonna experience the movies properly from now on. They gonna see small, pixalated copies of the films, from the comfort of their homes. No more going somewhere for a special experience. Just casual clicking on the film in a Netflix app.

Streaming services are not praising film-making. They are praising eyeballs. Look at the YouTube algorithm that is designed to make YouTube the most engaging place possible. Similar to the Netflix algorithm. Or any other streaming service algorithm.

The algorithm that was made by evil businessmen to ruin the world of film. Similar to how Sator wanted to ruin the world using his algorithm.

Conclusion

I think Christopher Nolan is too smart. And this is good. Since I want this. I don't want a dumb movie designed to satisfy my dumbass brain. I want something that will blow my mind every time I watch it or talk about it. Or blow my mind when I just make a simple internet search about it.

I discovered the Sator Square just by looking the word Tenet on Wikipedia. Think how much it blown my mind. Good job Christopher. Keep the good work.

Happy Hacking!

AI: Artificial Intelligence - Spoilers Resistant Film

Sometimes there is a movie so good that knowing it's spoilers makes for a better experience. I call them Spoiler-Resistant movies.

`lbry://@blenderdumbass:f/AI_spoiler-resistant-film:6`

When somebody reveals you important plot points of a story, usually film, we call a spoiler. A spoiler not because it make the movie more aerodynamic. It's not coming from the same word that describes a rear wing of a car. It comes rather from "spoiled milk" or similar words. It's arguable that even the wing of a car, when added by amateurs, spoils the quality of that car. This is why, maybe, it's called a spoiler. When somebody reveals you important plot points of a story, it spoils the story's emotional punch. I makes the story more predictable, shocking moments less shocking and tension less tense. This is why we tent to avoid spoilers before watching a film. And this is why warnings such as "spoilers ahead" appear on reviews of story driven media. Knowing the outcomes, knowing what's going to happen, rubs you, the audience, a chance to experience it first hand. A chance to be shocked, scared and surprised by the movie your are watching.

But can a filmmaker, writer or a game designer craft a story in such a way that it will be resistant to spoiling? Can a movie be done so well that even though you know exactly what's going to happen, even though you memorized the film by heart, it will still deliver the emotional punch, the way it was indented? In this article I will argue about one film that I think possesses such a quality. Of course, since this article is about knowing plot points ahead of time, I will be revealing the plot points of the film in this article. For some of you it will be spoilers. But the point of the article is to argue that they are in fact not spoilers. Since they do not spoil this particular film. But if you want to avoid them regardless, you've been warned. The movie we are going to talk about is a 2001 film by the director Steven Spielberg called AI: Artificial Intelligence.

I don't want to make chapters in this article. I want it to flow more like a well edited film. With scenes, organically, go from one thing to another. I would call this chapter "My Experience" or "The Time I Cried From This Film", but both of those titles make not a lot of sense. Since the thoughts about this film in my head resemble less a constructed whole I can break apart into digestible pieces. And more a web of interconnected neurons, thrown in a mix of history, and love for film in general.

Stanley Kubrick is another filmmaker I want to mention here, since like AI, his film 2001: A Space

Odyssey, even though shot more than half a century ago, still holds up on a technical level and it's visual effects. Same I would like to say about AI. It was made ironically in 2001, twenty years ago. And it holds up very well, both visually and on a level of film-making that it yet to be matched. I mention Stanley Kubrick for one more reason. AI ends with the words "FOR STANLEY KUBRICK" after the end credits. Originally it was supposed to be another Stanley Kubrick film, but he died in 1999 so Steven Spielberg took on directing it. When Kubrick was still alive there was already a debate between him and Spielberg about who would direct the movie. Kubrick was convinced that Spielberg was a better choice, since he has the necessary emotional, storytelling skills to pull a movie of such emotional caliber. Stanley Kubrick, one of the best, if not the best, film-makers of all time, was convinced that he himself couldn't make the film the way it was supposed to be made. But Steven Spielberg could.

The movie has a controversial last act that was criticized by many to be too emotional. A scene two thousand years in the future from the main events of the film when David, the robot child meets with his mother for one last day. Many critics agree that this whole act should be cut out of the film entirely. And that the film should have had ended when David is stuck in the Amphibicopter, underwater in the ruins of the old Manhattan. Many attribute this scene to the overly emotional nature of Steven Spielberg, claiming that he has "ruined" the work of Stanley Kubrick. But interestingly enough, Steven Spielberg himself was against that ending for a very long time. And Stanley Kubrick was one pushing for this ending to stay. The movie is about a robot child on an adventure to find a fairy in a dystopian, near future. What does it have to do with the two thousand years later scene with his mother? Why would a film-maker on a scale of Stanley Kubrick go so far for this emotion? Well unfortunately I don't have an answer. But I can speculate. This two thousand years later scene, was and is, the best cinematic experience I've ever had. You can see the Amphibicopter stuck underwater as David, the robot child is starring on a statue of a blue fairy, begging her again and again to make him a real human child "Please, Please, make me into a real, live boy.". The camera graciously floats away as the voice of legendary Sir Ben Kingsley tell us what happened next. How the

Amphibicopter's light had turned off and David could see the Blue Fairy only during the day. And then how the oceans had froze. And David could see the Blue Fairy through the ice. Barely making our her silhouette. But still begging her day and night. And thus two thousand years had passed.

What comes later in the film is misunderstood by a lot of people. David is approached by a group of someone. They look strangely like aliens. And the other movies of Steven Spielberg makes this a very strong case. What's interesting is that they are not aliens. They are super-mecha. A type of advanced future robots that merely resemble aliens. If you look closely, they are made of some substance resembling liquid glass. And inside of which they have their circuits flowing. With their thoughts visualized right inside their heads. It's a genius design, unfortunately it took too much inspiration from an aesthetic Spielberg made in his other film Close Encounters Of The Third Kind. Where the normal Grey aliens look possessing similar grace and slim bodies to the super-mecha in AI. When these super-mecha find David, he suddenly wakes up in a fake copy of his previous house. Where he used to live ones. He was kicked out from this house since he was believed to be too dangerous to other inhabitants of the house. To the organic people living there. His whole quest to find blue fairy was an attempt to become real. So he would be accepted to this home again. So his mother Monica would love him. He wakes up in that same house two thousand years later. When no humans are alive. When all oceans are frozen. When only super-mecha, him and his toy Teddy are alive. He finds a blue fairy there, in the house. Alive and talking to David. He asks her ones again to become a real child and she says that it's impossible. He asks if his mom comes back soon. And she says that it's impossible too. A tear comes out of David's eye.

To explain the significance of this scene, you have to look at another scene from the movie. The one where Monica leaves David alone in the woods. Just after he tried to cut her hair, and nearly drowned Martin, Monica's real son, they decided to get rid of David. But Monica couldn't do that, since she found a real emotional connection to him. So instead of going to Cybertronics, the company that created David and ultimately will be able to take him apart into pieces, Monica decides to give David a chance to survive and decides to leave him in a forest. He cries to her to forgive him, but since he is a mecha, no tears fall from his eyes. He is unable to produce a teardrop. But in the scene where David is talking to the Blue Fairy in the last act, and she tells him that it's impossible to turn him into a real boy, he cries real tears. David is a real boy by that point.

When I watched this film for the first time fully. Like from the beginning till the end. I was about 13 years old. Before that I certainly knew about the movie. When I was a kid I had seen it on TV multiple times. But never fully. It was one of those films that I knew the plot of, very well. But never seen it. The movie was effectively "spoiled" to me. When I sat and watched the full version, I was amazed. Since this was the first ever film that made me cry. No film before AI: Artificial Intelligence was able to make me cry. And only this, so called "spoiled" film, was the one that ended up bringing me the strongest emotional punch. The next day I was bragging about how good this film was to my mother, and so we watched it together. She didn't know much of the plot prior to watching it. I cried again, she didn't have any reaction. I think it's safe to assume that I cried from the film because I was spoiled. One movie critic, unfortunately I can't remember his name, said that when he saw the movie originally in 2001 he was very disappointed. But then he saw it again, years later, knowing all of the story, being "spoiled", he had the same emotional response to it as I had. He cried.

David finds a peace of hair he cut from Monica's head two thousand years ago and the super-mecha agree to return her to life. But there is catch. If they do so, she gonna live only for one day. When she will go sleep, she will never wake up again. David doesn't think too much. He agrees right away since he was designed to be so young. And the best day in David's life begins.

In my opinion this film is about trying to do something hard. Trying to achieve something that seems impossible. If Kubrick had decided to leave with David being stuck in Amphibicopter underwater. Then the lesson is, no matter how hard we try, failure is inevitable. No happy end. On the other hand. If we are willing to try making one thing so much that it will take two thousand years for one day of it being a success, does it even worth it? I think what Kubrick was trying to tell is that, yes. It worth it. Because this is Where The Dreams Are Born.

Happy Hacking!

I Invited a Professor of Robotics to Talk About Software Freedom

Pito Sage – A professor of robotics from the University of Bnei-Brak agreed to share some of his own thought about the Free Software movement and User-Freedom.

lbry://@blenderdumbass:f/I-Invited-a-Professor-or-Robotics-to-Talk-About-Software-Freedom:d

Today is a very special article that I was building to for a while. I asked one other person to write a couple of words about Software Freedom. Something that we are talking about on this channel a lot.

And so I'm happy to present, Doctor of Computer Science, Professor from the Robotics center at the University of BneiBrak, Software Engender, hacker and a personal friend of mine. Dr. Pito Sage.

When Jeison asked me to write an article for his show, I was confused at first. What should I write about? My experience as a babysitter? But then it struck me. Freedom is what interest our Blender Dumbass. To be totally honest I'm not a big Free Software guy. I have a Penguin painting on my wall. I love Linux. I'm not much of a GNU / Linux man. Though, I totally understand the reasoning for GNU being a part of the name.

My relationship with open source software is different. It's more of a thing I use, not because of my personal Freedom or my privacy. But probably because it helps me build things I want to build.

With closed software, when it comes to making robots, or any other technology. You either can't use it at all. Or you are lucky to have one program that is not too shitty for you to use. Usually is one of those code available programs like the Unreal engine. That is not very much open source. Or should I say Free Software?

See, I don't care much about the legality of Free Software. I don't care if I have the four freedoms legally. Quite frankly, if I have the four freedoms even illegally, I'm happy. Since all I need is an ability to build upon, modify and hack software to fit my robots.

But I realize that if I want to sell these robots to anyone in the future, I need to have a legal means to Software Freedom too. This is why I tent to avoid closed source software with a restrictive license or without a license when crafting any systems with in the robots.

For my own personal use. In my opinion if nobody knows, who gives a crap. If somebody reverse engineers a game that is otherwise proprietary, and now, even though illegally, I can modify this game.

Playing around with various things inside this game. This is enough for me.

But then again, I'm a programmer. And I understand why Richard Stallman insisted on ability to share. You need to be able to hire me. I can do that job of modifying your copy, or selling my copy to you secretly. But this may result in a court case with the developers. Or who ever holds the copyright.

Even though I'm a penguin rather a cow. I think our gnu is right. What do you think?

I guess this is all I have to say about the Free Software. I hope this essay of mine is not too short. And you found something of value in my humble rumbling to you.

Bye Bye, Pito Sage.

This was Dr. Pito Sage. Greatest robot designer in all humanity. Of course for those of you who follow Blender Dumbass, you realized that Pito Sage is a fictional character from a movie "I'm Not Even Human". And his part was written by me, J.Y.Amihud, Blender Dumbass. The writer and director of that movie.

Happy Hacking!

GPU

GPU – Graphics Processing Unit. Usually a card, bought separately from a computer, requiring additional drivers and other thing in order to operate.

lbry://@blenderdumbass:f/@blenderdumbass:f/gpu:6

This is a CGI (computer generated image) done by me. I utilized two computers to make this image possible. One of them I don't have right now. It's where I did most of the work. Modelling and texturing and setting up shaders. But the final image. The one you see here was finished on this laptop from which I'm typing. It's a pretty low powered machine. Not the most low powered one. But it even doesn't have a dedicated GPU.

It has what's called an Intel HD graphics GPU. Which is basically a fancy way of telling that all the GPU job is done by the CPU instead. The GPU or the Graphics Processing Unit is designed specifically to render an output graphical images. CPU or Central Processing Unit is designed more for a normal, linear type, programmed computation.

On a low power Intel CPU, without a dedicated Graphics Processing Unit, this frame above took this computer about 2 minutes to render and composite. If I would make a cartoon using this computer. And I would render all 24 frames of each second on here. Each second would cost me 48 minutes. Less than an hour. Which is 2 days of continuous rendering for 1 minute of a film. Considering that frames are exactly the same in complexity. A full feature 90 minute movie at this pace would be done in about 180 days, or about half a year.

Difference between GPU and CPU

A program is a set of instructions. *Do this then do this the do this and so on.* For example, a program that opens a text document and presents the text in the file on the screen of your computer, might have the following instructions.

```
filename = input()         # Ask a filename from the user
fileread = open(filename) # Open that file
text = fileread.read()     # Read the whole file
print(text)                # Print the text
```

This was a python code with only 4 instructions. Any other computer program is written in a similar way. The algorithm that produced the image at the start of this article is also written this same way. Only perhaps not in that same programming language. The algorithm that produced the render is called Cycles. And you can see it's full source code here. If you clicked on the link, you probably seen that rather than it being just 4 instructions. It's like 12 folders of files. And each of those files can get way beyond hundreds or even sometimes thousands of instructions. These are so called lines of code.

This is because a straight forward and dumb program of showing a file is way less complex than a path tracing rendering algorithm that takes shapes, textures, shaders and many other things into consideration. All of these features has to be written down in code. Making the sequence of execution way larger.

If you try to execute fileread = open(filename) before you finished executing filename = input() it will give you an error. Since it doesn't know yet what filename does the user want to access. In this case running the programs in a specific order matters. Since if you try to run them out of order you break the program.

But for a path tracing rendering algorithm such as Cycles it's not that big of an issue. Individual rays are such programs that can break if you try to mess with the order of execution. But two separate rays have no obligation to run in a specific order. So in theory you can compute one ray using one computer, while computing the other ray using another computer. And then combine the results to get 2 rays in a time of one.

Early on programmers realized that drawing graphics usually doesn't need much order of execution. And separate elements are very easy to compute individually. So a Graphics Processing Unit was developed. It was a device similar to a linear processor. That is packed with a huge amounts of tiny processors. Each of them is low powered and slow. But in the combination. Knowing that it can do a lot of computation simultaneously, it was able to draw images faster then the normal CPU.

It has a very noticeable drawback, though. For graphics is good. But most other programs that rely heavily on the order of execution, it is not. So this has not a lot of potential for most other applications. Until of course programmers found ways to exploit it. Designing programs that split them selves into threads. And that have multiple simultaneous functions executing at ones.

Today GPU are used primarily in computation. And not very much in graphics anymore. Since a lot of new algorithms take advantages from this inherent design of small, low powered but tightly packed computing devices. It became such a trend that modern CPUs are actually many small CPUs. Not as small as individual cores of the GPU. But CPU are now also cores separated.

The frame mentioned above took 2 minutes on a CPU. But on a similarly priced GPU it's expected to cut the render time to one minute or less. Bringing the rendering time of the feature film from half a year to less than 3 months.

CUDA NVidia and Monopoly

Most companies natural tendency is to strive for monopoly. This is obvious even with counties where monopolies are illegal. Microsoft had a court case where the prosecutor was asking to split the company in two for their anti-competitive practices. Unfortunately it didn't happen. So it's not a surprise that NVidia, the company that manufactures most of today's GPUs is using it's market dominance to create a form of monopoly.

Monopoly is when one company controls all of the supply for a particular item or service. Monopolies are very dangerous. Since as soon as one company hold such level of dominance, they can dictate the price of the market. They can manipulate and control their customers to worse quality products that cost more. They may take advantage of their position to do all kind of nasty things, within our outside of the company's interests.

Even though Monopolies are rare. Duopolies or near Monopolies are much more common. They are not as bad. You can switch off YouTube and go to Odysee. But in the current world there is still usually one company in each market that is disproportionately dominant. And therefore can have many of the same nasty things that a Monopoly might have.

YouTube is tracking their users and adding second, third and tenth advertisement for each video. Gmail makes it harder and harder to use third party Email applications like Mozilla Thunderbird. Microsoft has a pretty much universal deal (more or less 80%) with computer manufacturers to have Windows preinstalled, so they could collect all kinds of information about computer users.

I could continue and continue, but you got the point. Companies are usually there only for their own interest. And your interest comes only to persuade you to surrender something to them. It used to be your money. But now it's often your Freedom, Privacy and so own. In exchange for pointless, often highly addictive, dis-service. When a drug pimp comes to you and offers you to try out a drug without paying a single dollar, gratis. It's not because he wants you to "experience the feeling". He might say "You will feel very good". But his motivation is to get you on this drug, to get you addicted, so you will have to surrender to him.

NVidia saw the potential of GPU computing to become a viable marketing strategy for their Graphics Cards. They developed a special CUDA protocol. That they hold in secret. People can build software that uses this protocol. To split a given task to very many threads (like single rays in a render) to be computed in a very short time. With the rise of popularity of the crypto-currencies, NVidia got a rise of popularity too, mainly due to their CUDA. The GPU computing.

Now NVidia is the dominant player. Rivaled to some extent by AMD and lately by Intel, but NVidia still is the company that dictates the market. And since they are so dominant, they often dis-respect and out right subjugate their users.

NVidia Subjugated You if you use CUDA

On this channel I talk very often about Software Freedom. Most of my follower are familiar with the 4 essential freedoms of software. If you are not, click on the link. To make that render up there, I used a GNU / Linux operating system. Free Software operating system. I used Blender, the 3D modeling / animation tool. Free Software 3D modeling /

animation tool. I used Cycles. A path tracing rendering engine. Free Software path tracing rending engine.

All of these programs I used, if I need to, I can run, study, modify, share and contribute to. These programs are built and released to respect my freedoms as a computer user. I could use a graphics card by Nvidia to render this image in less then half of the time. But this will include Nvidia's gratis but proprietary, non-free driver. The only driver that can reliably compile instructions for CUDA.

There is a Free Software alternative driver called Nouveau. It's not developed or maintained by Nvidia. But rather by people who try to provide poor Nvidia's users some freedoms. The whole driver is reverse engineered. It's when somebody tries to figure out how a device works by cleverly poking at the device. Trying various things. And ultimately coming up with something that resembles it's protocol.

It's illegal for you to do it if you use their proprietary driver. It's EULA (End User License Agreement) states that you can't reverse engineer it. But it only applies if you agree to it, if you run the software.

In order to make CUDA work with wholly Free Software system, you need to never agree to Nvidia's EULA, reverse engineer the whole protocol. Rebuild a driver for it. And then use it. Not a first day experience. And more often then not, not a very smooth one.

You can see the Feature Matrix of the Nouveau driver to see how far they implemented the features and on which graphics cards. Usually the older the card the more features are implemented. Meaning there are some quite old cards with a working CUDA while the user keeps their Freedom.

The other computer, the one I don't have right now, had this exact setup. An old enough GPU to allow CUDA to run via Nouveau, Free Software driver. And it gave me just a tiny bit more speed then this laptop's CPU.

Conclusion

When making a Free Movie I want to make the movie in Freedom myself. And also I gonna probably make it longer then for half a year. My film "I'm Not Even

Human" took 3 years to make. And it's only 30 minutes long.

I know there are possibly better GPUs from AMD that have a freedom respecting driver. But with today's demand for the GPUs I think CPU is the way to go. On another computer that we have at home, unfortunately which is not mine, but I can ask to use it. That computer has a CPU with speed comparable to a very good GPU. With 20 simultaneous threads computing at the same time. I guess I'll manage.

Happy Hacking!

Reverse Engineering

Reverse Engineering – and act of learning how a certain thing works without having plans, or prior knowledge about this thing. In software, reverse engineering is used when a programmer wants to figure out how a certain (usually proprietary) program operates.

`lbry://@blenderdumbass:f/@blenderdumbass:f/Reverse-Engineering:e`

We all want control over our lives. And some of us, humans, want also control over lives of others. So called Freedom vs Power. Freedom is when you control aspects of your own life. You clothes, your software, things you will eat today. Power is when you control other peoples lives. Dictating what they will eat, what they will wear and what functions will have their software.

Freedom is essential. Power is most likely evil. Except in few instances. Like for example if one person is trying to take away other person's freedom by utilizing his own. Turning Freedom into Power. Stopping such behaviours with another form of Power, either law enforcement (police) or with clever tricks like copyleft, implemented in some Free Software licenses, is okay, and sometimes necessary. Copyleft makes sure software is free, having the 4 essential freedoms, giving people control over it. So they could decide what function will it have and in what form. But copyleft is also a form of necessary power. Making sure that people who copy the Free program do not strip away it's freedom. Requiring those using code under GNU GPL (GNU General

Public License) in their own projects to release the whole thing under GPL.

Without the police there would not be a possibility for freedom. Because then those who have largest muscles and more guns would control the lives of those who don't. Law Enforcement by being a form of power, balances this to such a degree that even though some people still try to take control (mafia, gangsters) they are faced with a great level of resistance. We allowed some individuals to possess power, so they would protect us from those who would otherwise take all of it.

Without the GNU GPL or other Copyleft licenses the world of Free Software would be a dark place. Every software company out there would simply use our programs to build their proprietary technologies. Using our work for their gain. And never share anything back. The fact of the matter is, free software is built on the idea of collaboration. Similar to scientists that publish scientific papers and then improve on each other's work. Free Software allows programmers from any place to develop any given Free program into any direction. Publishing his copy and therefore letting everybody else to benefit from it. Allowing companies to do this too, is good. But allowing them to take our programs, develop them further, and never share their developments back with us, this is already exploitation.

Further more they may be able to change our software to harm people and put this malicious software into devices. And if the software is not copylefted well enough they are not required to give people any freedom with the software on that device. Turing our work into their cheap spy machines and subjugation systems. But if the software is copylefted. They will have to provide the source code of this machine and the installation guide, so we could edit out any nasty thing, improve upon the software. And have a good product.

Hardware And Computers

When you think of a Free Software operating system like GNU / Linux you may imagine a following image. A task bar located on the left. Another tiny task bar on the top. Terminal. An application installer. Stuff that you can see by installing the latest version of Ubuntu. But all those things are just individual programs. The task-bars are just peaces of Gnome.

The application installer is a UI snapd client. Non of it has anything to do with the Linux kernel. But all of it together forms an operating system.

One very big part of an operating system, are those programs running on the background that do a very important job. But you probably don't think much about them. These are so called drivers.

When a given peace of technology is made, the engineers develop an interface with which a computer can talk to this device. Let's say a printer is installed into your USB port. What exactly should be the signal so it would print the document that you just typed? They can publish the protocol in some way. And a programmer of a text editor can add this protocol to his program. Making it being able to print pages.

But more often than not these protocols are very complex. And there are more than one peace of software that might use this device. I have Emacs, Libre Office, Gimp, IceCat Browser, PDF Viewer, Image Viewer and many other programs that I want to be able to print. Imagine what a waste of precious mega-bites it would be if every single program would implement the printer's specifications. And not just for this printer. For all models of all printers out there. This would be a disaster.

So instead of simply publishing what commands does the printer needs to print a document. They release a peace of software called a driver. This driver is talking to all programs. Any program that wants to print something just sends what should be printed. And the driver is handling the conversation with the printer it self. This fixes the issue of needing to implement every single printer. Just some standard protocol for all the drivers would be enough. But it gives a different issue. Those who make those drivers usually don't care about software freedom. The drivers are software. But most often than not this software is proprietary.

Reverse Engineering

In order to fix this issue, in the Free Software world we use a technique called Reverse Engineering. Let me try to illustrate reverse engineering to you.

So a hacker buys a printer. The printer comes with a peace of proprietary software that will make this printer work. But the hacker is not happy with it. He wants to run this printer in Freedom. And with it also give people Freedom when they run this specific version of printer. He looks online and doesn't find a Free Software driver for this printer. Only the proprietary one that he doesn't want to use.

He takes this upon himself to run the proprietary software driver ones, in order to catch signals that it send to the printer. Maybe this is how he can figure out what are those signals and how to code them into a Free Software alternative.

He loads the program and the first thing he sees is the EULA (End User License Agreement). You have to press *agree* or the software will refuse to run. It's not a Free Program so you cannot edit this out. You have to agree. But our hacker is smart and he doesn't agree just yet. He decides to read the agreement first.

After 38 paragraphs of boring, text in a nature of "*We are not responsible if anything breaks, and we might spy on you...*" our hacker gets to an important bit. "*If you agree to this document, you agree not to reverse engineer this software or the printer and you* *agree not to let anyone else to do so.*". In order to reverse engineer this printer. He need to agree not to reverse engineer this printer.

The dilemma is quite serious. From one side he can agree to the terms, but then not be able to reverse engineer the program. At least legally. A thought or two of getting away with it comes through his mind. Maybe if he doesn't tell people that he ran the program, there will be no court case. But he decides to be safe. Since what ever he discovers by running the program might put other people, those who will use the Free Software driver, at risk too.

Even if he doesn't tell others, you can't underestimate the developers of the protocol and of the driver. Technology companies are usually the worst offenders of Freedom vs Power. They will specifically design a license agreement and a protocol so to catch those who dare questioning their dominance. The software is probably specifically designed to catch whether our hacker tried to reverse engineer it.

So okay. He doesn't run it, what else can he do? He can de-compile it. De-compilation is a process of producing a basic source code type file from a compiled, binary peace of software. It's usually a big mess and it's missing all of the comments that programmers write in the sources. Also it will not have most of the variable names. So to understand a decompiled code you have to be very clever.

Just to give you an example, let's compare a JavaScript code from two different websites. One site is from people who respect your freedom. The other is from people who want to keep as much as possible secret from you.

From Google we can see this JavaScript code:

```
try{
var s_,s_aa=function(a,b)
{if(Error.captureStackTrace)Error.captureStackTrace(this,s_aa);else{var
c=Error().stack;c&&(this.stack=c)}a&&(this.message=String(a));b&&(this.cause=b)}
,s_ba=function(a){return a[a.length-1]}
```

Even the link from google is something like:

https://www.google.com/xjs/_/js/k=xjs.s.en_GB.5QOqjvyqHMc.O/m=cdos,dpf,hsm,jsa,d,csi/
am=QBFAAAAAAAAAAECgAAAAAGAMAAAAAAAAUAgAkAwOBADA0DwyAAAQAIAjoIAUAgQAAABMYD
9AwH8TAHAJNmEAAAAAAAABuAQySg1IFAQgAAAAAJlaOQEIASA/d=1/ed=1/dg=2/br=1/
rs=ACT90oGFC-CxP80dHWbL2CHMxSWgTUh5fw

And from Mediagoblin we can see this JavaScript code:

https://media.libreplanet.org/mgoblin_static/js/header_dropdown.js

```
$(document).ready(function(){
  // The header drop-down header panel defaults to open until you explicitly
  // close it. After that, the panel open/closed setting will persist across
  // page loads.

  // Initialise the panel status when page is loaded.
  if (localStorage.getItem("panel_closed")) {
    $("#header_dropdown").hide();
    $(".header_dropdown_up").hide();
  }
  else {
    $(".header_dropdown_down").hide();
  }
```

Just by looking at those two examples, you can with certainty say that the one on the bottom from the Mediagoblin is way easier to understand even for a simple person that has nothing to do with programming. One of the reasons why it is. It's because it has comments written in pure English. While the script from the Google is all mumbled and randomized.

People at Google do not want you to be able to understand what this JavaScript of theirs does. It probably does something nasty. But it's hard to know for sure since you have to guess. Reading their JavaScript is nearly impossible.

When our hacker decompiled the code of the driver for the printer, he was faced with something similar to the code of google. He was clever and made it a bit more readable, by adding spaces in places where a programmer would. Let's do this with the example from google. Just to illustrate what I mean.

```
try {

    var s_, s_aa = function(a,b) {

       if(Error.captureStackTrace)
           Error.captureStackTrace(this, s_aa);

       else {

           var c = Error().stack;
           c&&(this.stack=c)

           }

    a&&(this.message=String(a));
    b&&(this.cause=b)

    },

s_ba = function(a){

    return a[a.length-1]

    }
```

Did it become easier to read? For starters we can see that now there are variables such as s_ ,s_aa and s_ba which are very *understandable*. I have no idea what those mean. But since I can read try and error I can deduce that it tries to do something that it might have an error for. And that s_ba is a function that returns the input a without it's last character. And I know for sure why s_ba is not called something like without_last or exclude_final_character. Since if it would be readable like this, it would be so much easier to understand. And Google doesn't want that.

Also you have to keep in mind that this is only an example. A short excerpt taken from the actual code. With a real program it will be so much more work than this.

But our hacker was smart and stubborn. He got himself a big cup of coffee. He sat on the decompiled code for one hell of a night and understood all of it. For his surprise he discovered another bummer. The driver didn't even contain the code to talk to the printer. It would download it only when you click agree. Not before. He tried to run the downloading manually. But there was a complex DRM system preventing him from doing so. He had to either find ways to break the DRM, which is illegal in some countries. Or find a different approach to reverse engineer this damn printer.

Different Approach

Our hacker connected the printer via the USB to the computer. And wrote a simple program that would send various, random pieces of data to the printer. He would record what the printed is sending back. And would try to see whether changing one thing would influence another.

If you ever had a guitar, you are probably familiar with tuning it. You move the nob a bit forward and it makes the string tenser, raising the pitch. You more it a bit backwards and it loosens the string, lowering the pitch. But have you ever tried to tune a nylon guitar with a weak neck? Nylon has a tendency to loosen up. You have to tune it, then re-tune it few minutes later. A weak neck will bend with the tension of the strings. Loosening all of strings while you are tuning one of them. Tuning a nylon guitar with a weak neck is hard as hell. But it's easy compared to the type of fiddling out hacker is doing.

He has an unlimited amount of things he can change about the data that the printer sees. And sometimes he may get just right and make a printer do a thing. But it would not be replicatable. Since the printer is designed to be hard to reverse engineer. Our hacker will go through enormous sweat and tears, migraines and sleepless nights just to get one function of the printer to work. And it would probably not be the printing. Since this is what the printer was designed for. And this is what they made the hardest to figure out.

But the hacker is not stopping. As a warriors that goes to war and dies for the Freedom of his country. He will not surrender to some printer. His Freedom is more important. And he is also motivated to liberate other people. So he is not going to give up. Thus years pass and the printer is slowly gains more functionality. The scanner works. Then it can print black and white. And finally, three and a half years since the beginning, he constructs a fully Free Software driver to talk to this printer.

The driver perhaps misses some features of the printer. Such the RGB color lights. "Who needs that?" - says the hacker - "The printer and the scanner works, so my job is done". He uploads the driver and people with in the free world now can use the printer in freedom.

Casual Consumer - Critic

Joe has this same printer at home. He watches some videos about privacy and grows with concerns about software he uses. Currently he is a slave of Microsoft Windows, a proprietary driver and a huge amount of other malicious, non-free software. So called *Linux* users are telling him that *Linux* can do everything that Windows can. That he has nothing to fear. And that he must to give it a try.

He decides to try this *Linux* thing out. He struggles with internet tutorials to understand what's a Live USB. And then half an hour later he has a working Ubuntu installer. He boots the Ubuntu Installer and tells it to install the operating system along side his Windows. Just so if doesn't like it, he may return to Windows.

For some magical reasons, it works. And he is now booted into his new system. Just so happened to be that hacker's printer driver was installed with the system. And he tests it out. Prints a document and it works. But now he wants to show off the RGB lights thing. He enters the setting for the printer and cannot find this setting in the familiar spot. He searches online. He doesn't find any information about it. Suddenly he thinks that "*Linux isn't capable of running the RGB lights*" on his printer.

Later he decided to buy a new version of the printer. The one he has is more then three years old. He connects the new printer and nothing works at all.

This Joe doesn't complain about the work our hacker has done. He doesn't complain about the Free Software, since he probably doesn't know what's Free Software. He complaints that this thing called *Linux* doesn't do what he was told it was able to do. Joe will boot into Windows and forget about his Ubuntu partition. Occasionally he would show people how much of *hacker* Joe is, by booting for a few minutes into the GNU / Linux. But he would never transition fully.

This is happened because the *Linux* users that told Joe about it, told him that it would be the same exact experience. If it would, there would be no difference what so ever. Basically, if the *Linux* users were right, when booting Ubuntu, Joe would see Windows booting. When using it, he would see as if he uses Windows.

They probably told him about the privacy. But they failed to explain the Free Software side of thing. They failed to explain the hard work, hacker and other programmers do to maintain the Freedom that comes with GNU / Linux. They failed to mention the GNU.

Conclusion

Before I gonna conclude this huge article. I want to mention one video. Unfortunately it's on the Evil Tube (YouTube), but it's worth to take a look. It's about a group of people that took a task to reverse engineered the HTC Vive VR Controller. Here is the link to Invidious. Invidious is way to load YouTube videos without running their proprietary JavaScript code.

You may help our cause by helping to reverse engineer things. You don't have to write whole drivers. If you just publish what you find about a given device. There are most likely people who will build a good driver using your knowledge. So please take a device people need. Like Nvidia GPUs and reverse engineer them. Give us the Software Freedom that we deserve.

Happy Hacking!

LBRY / MarkDown Browser

Something about the modern web just rubs me the wrong way. I want the web to be simpler. Without constant Java Script bloat and other BS.

`lbry://@blenderdumbass:f/@blenderdumbass:f/lbry-markdown-browser:0`

What I'm about to show you is not released to the public, yet. It's code that works and that is not very hard to re-implement if anybody wants. As soon as I finish this release I will publish a working example.

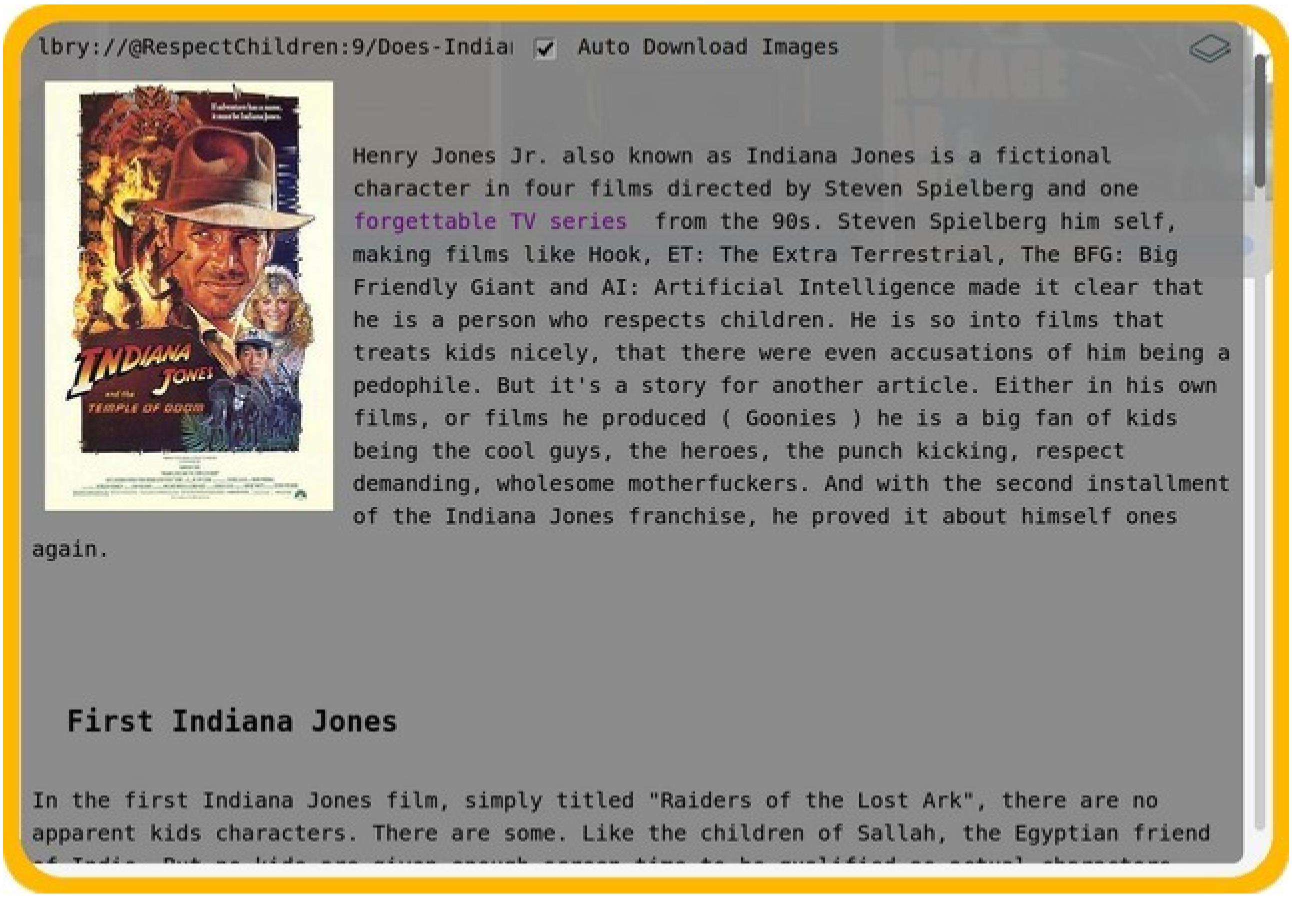
HTML

Originally, Internet was a way to share information. A protocol of formatting this information was developed. HTML. HyperText Markup Language. And this protocol became the standard. If you wanted to write an article or publish a web page. You would write an HTML document. This document would be hosted somewhere. Some server. Probably even your own, home computer. And people would be able to get a copy of this document over the network. And use a peace of software (browser) to transform the HTML language, that resembles code, into a page, a document, with rich text, embedded images, and links to other documents like this one.

Images embedded in HTML were just simple links to a file that was not HTML. You browser downloads the HTML and reads it. And it sees that an image should be embedded. A link to this image is also provided. So the browser can send another request. This time to retrieve the image. And compose everything into a tight text document.

Internet was a way of publishing articles, documentations, scientific papers. Not a sophisticated app store it grew to be. Today people don't use software. They use websites that substitute software. The HTML is obsolete, JavaScript is ruling the Internet.

JavaScript is a way to program any functionality into a web page. Making a browser into an operating system. And operating system without installed software. But with software browsed. You load a page and with it you load tons on malware. This malware gets executed for you to get videos, chats and even this article. But what if there was a better way?



Henry Jones Jr. also known as Indiana Jones is a fictional character in four films directed by Steven Spielberg and one forgettable TV series from the 90s. Steven Spielberg him self, making films like Hook, ET: The Extra Terrestrial, The BFG: Big Friendly Giant and AI: Artificial Intelligence made it clear that he is a person who respects children. He is so into films that treats kids nicely, that there were even accusations of him being a pedophile. But it's a story for another article. Either in his own films, or films he produced (Goonies) he is a big fan of kids being the cool guys, the heroes, the punch kicking, respect demanding, wholesome motherfuckers. And with the second installment of the Indiana Jones franchise, he proved it about himself ones again.

First Indiana Jones

In the first Indiana Jones film, simply titled "Raiders of the Lost Ark", there are no apparent kids characters. There are some. Like the children of Sallah, the Egyptian friend of Indie. But no kids are given enough screen time to be qualified as actual characters

This is the software that I develop. It's called VCStudio and you can get a copy of it at:

notabug.org/jyamihud/VCStudio

Note that at the time of writing the repository still doesn't have this latest version that you see on the picture. But soon it will be there. As you can see you can read the text, there is an image. And the url of this article starts with lbry://.

When I started developing software I started using online services such as Github and Notabug to host the code and to have a version control. With it, there came a feature to write little .md documents to explain functionality. You could just simply type the text directly and it would be enough. Or you could use a smartly developed markdown language to enrich the text, embed images and mark links.

This markdown language started appearing in more places. Blender's Devtalk, LBRY comments section. Even this article is written in markdown. This is how I can do *this text italic* and this text in a box. If you edit the url of this article and change odysee.com to spee.ch you will get the source code of this article. Not the HTML + JavaScript version. The .md markdown version. The one I uploaded.

When you make software, you want people to understand how it works. So I written a bunch of little .md files to explain different parts of VCStudio. Later I realized that Markdown in not very hard to implement. So I added an ability to read those documentation articles, directly in the software. I made a tiny Markdown browser. And if a link in the text was to another .md file. It would be able to open that link too. What you see on the image above, is this documentation browser.

LBRY:// vs HTTPS://

As I already said. If you change odysee.com to spee.ch you get the source code, the raw file. You can download videos and images and other things from odysee like this. Or make links to raw files in your sites. Instead of linking the page.

If you didn't know Odysee is a front end to LBRY. A protocol that is trying to rival HTTPS. Hypertext Transfer Protocol Secure. LBRY is not as simple as HTTPS. It's not one person hosting and another person connecting and downloading. It's a whole new approach. Multiple people have copies of the publications. Multiple nodes of data about the publication coexist. An actual crypto-currency coin (LBC) is holding all of it together. It's HTTPS but resistant to censorship and with a built in tipping system.

One problem with LBRY currently, is that there are not a lot of LBRY browsers. There is a LBRY Desktop app. Which is an LBRY browser. And there is Odysee.

Which a web site that let's you brows LBRY over HTTPS.

Combining the Markdown articles and LBRY. You can imagine that this new system is actually Internet 2.0. It's ones again, documents, articles and scientific papers. Ones again in a simple format. One again without JavaScript apps. But just lacking a good LBRY / Markdown Browser.

By coincidence I upload updates about my software to LBRY. And I wanted to use the newly formed Markdown browser to see update information too. For this a simple hack of changing the lbry:// to https://spee.ch/ was a good enough, dirty solution, to get the markdown data, that I can parse into an article. So I did it. And now if you link to lbry:// and the publication is an article, it will load it directly into the software.

On the screen above you can see this article that I wrote in a different channel. I just wanted you to know, in case you are intrigued by the part visible in the image.

Conclusion

This LBRY / Markdown browser of mine is not a native lbry:// browser. It can't play videos. It will break if you try to give it an image and not an article. It assumes that everything is markdown article. But it's a cool toy, a cool thought experiment.

What if websites were developed using Markdown? What if Markdown websites were placed on the LBRY? Maybe then, this markdown browser would make sense?

Happy Hacking!

Web Site Score System

How to know whether to trust a given website or not? Well I think I might be able to give a solution.

lbry://@blenderdumbass:f/@blenderdumbass:f/Web-Site-Score-System:4

We all use Internet now a days. We watch videos and read articles on Odysee, we post things to Mastodons and GNU Socials. We use websites to read news and find information about the world. And we use other websites for entertainment and recreation. But the key here is that we all use websites.

I want to dedicate an article about what I would conciser a good website. And why. For this, of course, I need you to have some basic knowledge in how they are made. So we are going to take a very short look at what websites are. And how it all works. And then I will present you with my score system. Please read this, first part as well. Since I want you to see where my reasoning comes from.

What are websites?

Websites were originally collections of documents. You would have an index.html file on a given server. When you wanted to load a website like example.com you typed this domain name into the web-browser. The browser would send a request to the DNS service and receive something like 93.184.216.34 back. Which is similar to a phone number. Without even telling you this number, the browser would call it and receive a file. Usually in a form of an HTML document. And usually it would the index.html file that is stored on the server.

Unless some other conditions are met. Like for example the URL could have something after / and the server would give you a different file, depending on what is written after the /. In simple sites every / means a directory, and you just specify a name of a file you want to get. Like for example gnu.org/music/free-software-song.html would give you free-software-song.html from the folder music. But let's comeback to the example.com that you received earlier.

The file would look something like this:

```
<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <style type="text/css">
    body {
        background-color: #f0f0f2;
        margin: 0;
        padding: 0;
        font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open
Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;

    }
    div {
        width: 600px;
        margin: 5em auto;
        padding: 2em;
        background-color: #fdfdff;
        border-radius: 0.5em;
        box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    }
    a:link, a:visited {
        color: #38488f;
        text-decoration: none;
    }
    @media (max-width: 700px) {
        div {
            margin: 0 auto;
            width: auto;
        }
    }
    </style>
</head>

<body>
<div>
    <h1>Example Domain</h1>
    <p>This domain is for use in illustrative examples in documents. You may use this
    domain in literature without prior coordination or asking for permission.</p>
    <p><a href="https://www.iana.org/domains/example">More information...</a></p>
</div>
</body>
</html>
```

Somebody would prepare this file ahead of time. And this somebody would know the HTML format, so your browser would know what to do with it. In the <head> portion of the file you can see that the

<title> is "Example Domain". So the browser knows to change the title bar, or recently the tab bar, to this name. Then there would be a bunch of other settings. Stuff like background-color:, font-family: and margin:. That would describe the theme for the document. It's called CSS. Cascading Style Sheets. This could be either a part of the HTML file. Or loaded separately. And finally there would be the <body>. The documents it self. That when read by the browser looks something like this:

> **Example Domain**
>
> This domain is for use in illustrative examples in documents. You may use this domain in literature without prior coordination or asking for permission.
>
> More information...

In this file, the author included a link. A link to a different page written similarly. HTML has also options to embed things into it, things that are not the part of this HTML file. For example with the <frame> function you can embed another HTML file into this one. It will be loaded separately, but rendered as one whole peace. You can embed images, and recently, with the HTML5, even videos. But I guess the most powerful and in the same, the worst addition to this format is JavaScript.

JavaScript

JavaScript is a programming language that your browser most likely can interpret. Think about a python script and the python3 program installed on your computer that is able to run this script. A python script is a program. The same is true with a JavaScript. How often did you install software with an unknown origin? How often did you simply trust a script on the internet? If I give you a .py (python) file. Would you simply run it? With web browsers there is not even a question. JavaScript is there, embedded and it runs automatically. When you load the page, the script runs.

A few months ago I talked to one person online about my program VCStudio. He was willing to give it a try, but he didn't trust me. So he read through the code, just in case I implemented something malicious. He found this file that requests an update information about the software. He was convinced that he will confirm everything to be good and then, when he launches it, it will download something malicious. He refused to run the software. Even though nothing malicious was implemented.

Now, let me ask you this question. How often this person checks the code of the JavaScript that is running in every single web-page today? The answer might shock you. Not only he doesn't check it. Not only he doesn't care to check it. But it's often impossible to check it.

I know, the title of this article is "Web Site Score System". And I know with each new chapter of books I read, I type longer and longer articles. Perhaps soon there will be book length articles, that I will release simultaneously in a printed form. But don't be too judgemental, I will get to the point, soon. Or at least I hope so.

The example.com example I gave you earlier was written by humans. It was a website specifically designed to give people examples about how the web works. But this is not how the web works. Have you heard about Wix or Square Space or other websites that let you create websites? Those are website applications, basically a very large JavaScript programs. But this is not what I'm complaining about.

When the website is not designed by humans writing HTML code, it's designed by a program. For example, you will set a button in a particular place in the page. And a web compiler would write something that will implement this button. Those who made this web compiler are not very interested in the HTML to be readable. Since their sources are not HTML. They are working with a graphical interface to create the site. Sometimes it's even inevitable that a web compiler would be used. Some do a great job at sustaining a good level of HTML source readability. Like wikipedia.org. But most often than not it looks like a pile of goo.

While with HTML it's a minor inconvenience. Humans would still be able to read most of it with ease. When it comes to JavaScript, web compilers turn program code into an unreadable, unrecognizable fart.

Free Software insists on having the software source code available, for people to be able to read and modify. Most software, in order to work, has to be compiled into a binary file. Something that the CPU can read. But not something that humans can read. If you are an expert, you can try de-compiling, and reverse engineering this binary program. But it requires an insane level of concentration. And for most reasonable sized programs it's completely unfeasible. So readable sources are essential.

JavaScripts made by web compilers are very similar to binary programs. It's readability is so bad, that most humans will give up, the second the look at the code. The funny thing it, that it's not binary. It's JavaScript. It should be human-readable. But it's not. It's so mumbled, randomized and utterly insane, that no human will be able to read a reasonably sized program like this. And while some websites have this for pure convenience of having a website editor. Other do this to hide malicious features.

In a browser you can right-click and View Source. Go to Google.com and do that. You will be surprised of how hilariously unreadable the web-page's source looks. A page with 1 input bar, a logo of a company and a few collapsible icons on the top, looks like it contains all the world of viruses. Google definitely doesn't want you to know what all this code does.

Which brings me, finally to the list of specifications. A score system.

The score system

The score will be a number from 0 to 8. The higher the number is, the better the score. I will list a bunch of features that a I want every website to have. If a website has a given feature, he will be rewarded with one score point. So lets begin.

0 Works over Tor

The website should not restrict from what IP address the user is connecting to it. If a user wants to use VPN or Tor to keep his identity private, he should be able to do it. Sites that break functionality, or refuse to load over Tor will not get this point.

1 Readable Sources

This point will get a site that has it's sources readable. Meaning, even if they use some kind of web compiler to create the HTML and JavaScript code. It should be still easy enough to understand. Especially JavaScript.

2 Free Software JavaScript

JavaScript is software. And this software should be Free. So a user could have the four essential freedoms with the JavaScript. There are programs like LibreJS that check each JavaScript file for a license. But obviously if the file is compiled and mumbled, there is not license section on it. So at least the source from which the JavaScript is compiled should be Free Software. Similar to Free Software compiled to a binary executable.

3 Works without JavaScript

Some browsers do not support JavaScript. Sometimes a user might disable all JavaScript from running. For example the Tor Browser has a function to disable all JavaScript if you select the highest security setting. The website should not break it's *core functionality* if the JavaScript is off. Maybe the developers could make a separate HTML5 version of the site that perhaps looks worse. But still gives the users the *core features* this way.

4 No Data Collection

Websites should not collect data. A website that sells physical things, might know your credit card number and your address, so they could charge you and send you the things that you buy. This is okay. But a search engine doesn't need to know those things to operate. So it should not collect the data in the first place. Sometimes analytics about usage of the website is beneficial to it's developers. In this case, they should ask for permission to collect this. Giving the user an option to opt out of this data collection at any moment. The website should not break it's core functionality if the user doesn't want data unrelated to this functionality to be collected.

5 No Data Keeping

If a website, like the one I mentioned, that sells you physical things, collects your data to send you the things that you buy. As soon as this transaction is over. This data should be removed from their servers. They should not keep it any longer. They could keep the data about what item was bought and in what amount. But not the credit card number and the address of the person that bought it. This has nothing to do with their book keeping. So they should erase it as soon as possible. And even with book keeping. It's beneficial, when the season is over and the record is not needed anymore, the record should be erased.

6 Free Software Server

To make the web service one extra step more Free. The source code of the server should be available. And should be Free Software too. Since if a person doesn't like how the site operates, but likes most of it's functions. This person could make his own, similar site, editing out the nasty bits.

7 Free API

If a website is more complex then simply storing HTML documents, it must have an API to access it's stuff. Especially if the website is a web application. So people would be able to write their own clients for this website. For example LBRY protocol of Odysee could be considered a Free API. The API also should not require sacrifices. So everybody could build software using it, and no accounts, or paywalls should be there to restrict it.

Let's score some sites

Name	0	1	2	3	4	5	6	7	Score
Azlyrics.com				•					1
Google.com	•								1
Youtube.com	•								1
Gmail.com	•			•				•	3
Github.com	•		•					•	3
Odysee.com	•		•		•		•	•	5
Duckduckgo.com	•		•	•	•	•	•		6
Notabug.org	•	•	•	•	•	•	•	•	8
Gnu.org	•	•	•	•	•	•	•	•	8

Happy Hacking!

AI is Popular Because Pedophilia is Not

I’m about to prove you the sentence that’s written above this sentence.

lbry://@blenderdumbass:f/ai-is-popular-because-pedophilia-is-not:1

DISCLAIMER!!! In this article I decided to dig objectively into a topic that people do not want to think too much about, since it's very grim. So, you've been warned. I hope you can be a respectful reader and will not jump to conclusions just because I want to answer a question that seems obvious.

Multiple film directors have stated that they avoid working with children and animals. Some stated that they are hard to control and thus require more resources. Others said that the reason is more in the aesthetics. It's way more interesting to see what infants are doing, and how they react to things, rather then if the same was done by an adult actor. And thus it's a *cheap trick* to use children. It's gonna be too simple to make an interesting movie. And that might result in a lesser film.

When an infant does something (*anything*), adults usually adore him for it. If a little boy behaves like a gentleman, people around him will react much stronger to it, than to a real, adult gentleman. When a kid drives a car in a film, or uses guns, or does anything dangerous. Audience are much more amazed and the scene is much tenser and scarier than if this same scene had adults doing those same things. Why is it? And how is this related to AI?

Reproduction

All live forms want to reproduce. Since mortality is inevitable, all life forms want to share with each other their genetic code and produce a copy of them selves, called in a human language "Child".

In humans the reproduction system is more advanced compared to some basic live forms. We tent to judge very hard, our potential partners. And our abilities to survive, together with attentiveness are very crucial things, we have to possess in order to be deemed reproducible.

Evolution works this way. We tent to choose those who can survive. Because if we choose those who can't, the species might go extinct. Lately with the advancement in medicine, almost everybody can survive, so the reproduction qualities are shifted to attractiveness instead. Both physical attractiveness and a *pretty live style* are may get you *laid*.

But there are two different types of physical attractiveness. Both are linked directly to our reproductive system. And they are often confused with one another, since their regions in the brain are close to one another. One of them we call "Sexy" and the other we call "Cute".

"Sexy" - is when a person possesses a more reproductive quality. For men it might be qualities as a protector. Strength, Competence, Dominance, Maturity. And of course a large and strong sexual organ. Which signals that this person is able to reproduce. For females it's ability to carry a baby, have milk (breasts) and so on.

"Cute" - is when a person looks like he deserves care. Usually smaller, with big eyes, thinner, and younger looking. This is a picture of a child. Of a baby. This is important to reproduction, since kids need to be cared for, after they are born.

But you probably noticed that people tent to find other people sexually attractive for their "Cuteness". Why is it? This is simple. Cuteness makes a person care no matter if it's a baby or not. But if you add to it also sexiness. An ability to reproduce. A person that possesses both qualities cannot be rivaled by those possessing only one of them. This is why women are also trying to be cuter, nicer, thinner, younger and so on. And lately even men do the same.

This is also very possible that it works because of how close the two are in the brain it self. All of the brain is divided into regions of cerebral activities. And the reproductive region has both *sexiness* and *cuteness* very close together. Since sex and carrying for children are parts of the same program. The reproduction.

If you read the studies of Dr. James Cantor. If a person has a brain damage, either a knock on the head, a headache, or a different type of brain damage. Especially in the early ages, when the neuroplasticity is stronger and the brain is still forming. It can cause the two to "connect" and to wire together, creating an anomaly. This anomaly is quite frequent and it's called pedophilia.

Basically pedophilia is when a person perceives cuteness and sexiness as one and the same. And in some, extreme cases. Sexiness doesn't even register for this person. Only cuteness exists for both tasks of the reproduction. And thus, those individuals suffering from this neural dysfunction, tent to find little children similarly attractive to how a regular men see attractive women.

This is not a strong anomaly. Regular men and women can tell apart attractive children from non-attractive ones. Most people even feel sexually attracted to people in their late teenage years. Since they are literally indistinguishable from young legal adults. Many girls as young as 15 lie about their age to get laid. And it works surprisingly well. A lot of men do not even know that they are breaking the law. It's no wonder that a knock on a head in a specific area can increase it further.

Monsters! Perverts! Criminals!

When I was 14 or so, me and my father, together with a large crowd of people, came out of a synagogue where we used to prey. It was a Shabos evening (Evening of Saturday) and a lot of people were going to the food gatherings. A huge crowd of people came out at ones from the synagogue as usual. But this particular time I noticed something. A little girl, about 4 or 5 years old, was crying, about 50 meters ahead of where we were. We were coming towards her. Before us, in the same direction went a significant portion of the crowd. A lot of them payed attention the girl's existence. Looked at her in a glimpse, but continued forward. We were in the end of the crowd and no one before us had seem to care, to help the girl. When we got closer, my father took it upon him self to ask her what happened. She said that she lost her parents in the crowd. And so my father took it upon himself to help her find them.

Why from so many people nobody cared to help the girl? Why with the neural system clearly evolved for millions of years, to care for babies, everybody resisted helping? A lot of people noticed the girl. A lot of people saw and heard her crying. Why did nobody helped her apart from my father?

The answer is quite obvious. People now a days are scared of children. They like children, but they are afraid to be with them, talk to them and otherwise interact with them. Because if you do that, somebody might think that you are trying to hurt the child. People resisted helping the girl, because they wanted to avoid persecution for this. They resisted the urge to help the poor kid because they didn't want to be seen as pedophiles.

From the previous chapter you understood why pedophiles exist. But why are they so unlikable is another thing I want to answer. For some of you it may be obvious. But it's not as obvious if you actually will try to answer it. The majority of arguments for why it is, break logic rules. A lot of them have circular reasoning. It's when the proof is creating a condition for the proof to exist in the first place. And it's not very scientific to say the least.

To make it simpler for myself. I will start with stating the fact. Pedophiles are not liked. And from this I will try to reverse engineer the answer. I could go the other way, which is more scientific. But I don't have the ability to conduct large scale study.

If you look into the history, this hate toward child sexuality wasn't there half a decade ago. It was almost there, but not in this level. In the early 20 century when a teacher would marry his student, even if she would be a little girl, that was considered totally acceptable. For example the 9 year old Eunice Winstead and her husband Charlie Johns. If you look even earlier, you can see that Child Brides were quite common. But now a days, people avoid helping children, so by mistake, not be confused with pedophiles.

If you look not that far into the past, into 70s. Major motion pictures (movies) could casually include pictures of naked children. A good example of it would be Superman from 1978 directed by Richard Donner. Where you can see full frontal nudity of a 4 year old, little Clark Kent. If similar stunt would be done today, people would be so outraged, that the movie will be outright "canceled". Case in point is the movie "Cuties". Which, even though didn't have child nudity, caused so much outrage, that it feels like we live in a world filled with Neanderthals.

I think technology is what made it so unacceptable. Pornography illustrated to a lot of people what sex is. And what perversions are. Before video and photography, the only pornography people could get was cartoons.

Psychology developed and matured. Perhaps not fully. The effects of sex on children are not very well studied, since all the doctors are able to get are the cases of abuse, where the children had complained to authorities. A lot of cases, like those 15 year olds who lie about their age, go undetected. And if even both lovers know the ages of each other, with the increasing dislike to pedophilia, both of them will keep it a secret. A little girl who wants the relationship will understand that talking about it will make this relationship disappear. So doctors have nothing to work with. And they cannot conduct a study themselves without going to jail.

And this brings me to the logical fallacy about all of it. The studies that they can get data for, show that kids have anxieties and develop post traumatic stress. Since what they lived through, was abuse. This is why they complained to authorities in the first place. This is why the data exists. But this is from what all the conclusions are made. People love children and they don't want them to be hurt. And if something like this is told, people react with anger. People who love children made it harder to love children.

Similar thing is happening today with adult women. The "me too" movements and the ultra feminists make it harder and harder for men to express that they feel attraction to women. Some women are abused. It's true. It's happening. And it's worth fighting with. But the unintended consequence of it makes it look like all attractions lead to rape. And it's simply not true. If the trend continues, we will see a crying woman in a desperate need of help and a large crowd of men, ignoring her, to avoid being confused with rapists.

Where AI comes into all of it

Sorry for a depressing chapter. But I thought I needed to give you context.

AI is a program that learns by it self. Remember the film Terminator 2? Where the scientist explained that what he builds is a chip that will be able to learn. It was the most science fiction idea out there in the late 80s and early 90s. But now it's done simply, using software.

Basically AI is a program to which you feed a lot of data. The more data you feed into it, the smarter it gets, the more accurate will be the results. You can think of each data point as a learning example. The more examples it gets the better it will understand how to deal with a issue at hand. Sometimes you may generate the examples using software too. So it will learn on it's own. Maybe even let it try to tackle the issue with it's *learned* settings. And randomly tweak them, to produce various different results. Each iteration of such test may refine the settings further. In other examples it might get a lot of images of say, cats and be asked to recognize a cat in an image previously not seen by it.

AI in it's nature resembles the brain of an infant. As an infant is learning about the world. Encountering things he didn't see prior. AI does the same. And if people love to see kids do interesting things. Because watching their creativity in tackling issues is interesting. With AI it's quite similar.

There is a huge trend of videos and articles about simply trying to give AI programs inputs and watching the results it gives. This is too simple of an idea to be interesting. But the nature of AI being Infant-like, makes it popular. It's the same type of "too easy", that those film directors that I mention in the beginning afraid of. Perhaps they can now avoid kids, animals and AI in their films. Since it's too easy to make it interesting.

But the equation of why AI is so popular also has to do with the fact that so many individuals were bad people. Because kids were hurt so often by those who are believed to possess love to them, which most likely wasn't even the case. It resulted in a global trend of adults avoiding children. So they would avoid infants and avoid observing them as a result. AI would be a great substitute for this. AI is not cute as it doesn't have big round eyes, and a short body. It's cute because AI is at it's infancy.

Conclusion

I know I did a lot of digging into perversions to prove one tiny point about software. But I guess, what I was trying to do is to give you a change to dig with me. A lot of people afraid of certain topics. And breaking through this barrier by starting first, can liberate others to express their opinions on the matter.

Happy Hacking!

Making Money Of Free Software

When people hear Free Software they think Gratis Software. And even with the attempt of calling Free Software Open Source it still didn't help the perception. People think that if the source code is available, there is nobody willing to buy it anymore. I want to prove them wrong.

`lbry://@blenderdumbass:f/Making-Money-Of-Free-Software:f`

Contrary to inclusion of the word "Free" in Free Software, it's still can be sold. A lot of people do sell Free Software, and a lot of them are very successful. There is even an article about selling Free Software on the GNU project's website.

I want to express my personal views on the subject. This topic is quite often takes up a large portion of my cognition. And it's obvious why. I'm a programmer and I work at a store. And there is nothing in this world that will make me work on a peace of proprietary garbage. At least in a way in which I will be aware of this fact. Potential Free Software profit ideas are visiting me. And this is what I want to talk about it.

But before we start. Free Software means software that respects users freedom. Not software, copies of which you can obtain without paying. It's usually the case that Free Software is gratis. And it's making the "Free" part in Free Software so much more confusing. But it wasn't the original intention. Free Country, Free Speech, Free Software. Free as in Freedom. Not as in "Free Beer."

The full definition of Free Software you can find on [gnu.org](gnu.org)

Binaries are not required for Free Software

Software embodies knowledge. A set of rules and instructions to make a specific computation possible. If math can have equations designed to solve a problem, software is just the same. Software is a large equation, written more often than not, in a programming language rather than in pure math. But math is a programming language too. It's just a bit too abstract to program real hardware with math.

If you take any math discovery, or any other scientific discovery. A scientific paper, a text with information, is often enough to convey the findings of the scientists. They do their job, and come up with various tests and hypothesis. And write them down. Some may do the extra step and actually build something using this knowledge. But the knowledge is what's important. Not that it's used.

With software, the scientific paper of the research, the software source code, is what gives knowledge. This is what is important. People may build it, and it should be build-able. Or it's not software. But the build itself is not necessary, to be qualified as Free Software, or as software in general.

So you can make a Free Software program and just never release the compiled version of the program. So only people who can go through the hassle, will build it from source, when they want to install a copy. This will be reasonable enough. You will preserve all the four freedoms if you do so. If people can build it. They can use it. A level of expertise is required. Which is to be expected. A person using any software have to know how to use the computer first.

So if source code availability together with the Free Software License is all you need to make a Free Software program, why not make a business of actually selling compiled copies? Why not publish your findings, scientific paper, software source code, in such a way that everybody can look at it, scrutinise it, hack it and extend it. And then simply open a business of building and selling a "product" that's using this *knowledge*? In other words. Why not charge for a Binary Executable, while keeping the source code *Free*?

You may think that it's a ludicrous idea and that it can't possibly be done. If there is a gratis version of the program, you will definitely install it the gratis way, rather then paying for the privilege of not dealing with building from source. Financially it makes a lot of sense. If you have to pay one way, that's might be a little bit simpler, and get the same thing gratis, that may require a bit more to go through, you will most certainly take the rout of gratis. And never even think to pay.

While this statement makes logical sense, it ignores two fundamental things about humans. One, is that humans are extremely lazy animals. And the other, is that humans are usually very far from logical. Case in point is the Free Software audio design tool Ardour which costs (at the time of writing) $45 for a compiled copy and is distributed under the GNU GPL license. Another good example is a game Lugaru which costs $9 for a full copy, but is also under the GNU GPL license. Both programs have their source code publicly available. Ardour on Github and Lugaru here. And both programs still successfully sell binary copies.

You don't have to share your copy

When it comes to the GPL license, a lot of people complaint, that if they release software with GPL code in it, the whole thing should be on GPL. This gives a false impression like they *have to share* their modifications. But they are not. They are merely *free to do so*. Only if they keep the license.

If you are not obligated to share your copy at all. What's stopping you to put a barrier on you sharing it? Tell people that you will share it only if they do something for you, like give you a specific amount of money. Then they receive both the binary and the source code in one large package.

To be qualified as Free Software, you only have to give the source code to a person that already has the copy of the program. In other words, only when a person gets from you the license. So, in theory, you can package them together to avoid releasing source code to those who didn't pay. *As supported by section 6, sub-section d) of the GNU GPL v3*

> Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the

> Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. ...

If you give them such copy with a Free Software license. Let's say GNU GPL. They will have the same four freedoms. And they will be able to share it further. But since they payed for a copy, they are going to be less likely, to give it away.

This was done by the Free Software Foundation with the early GNU software. In the early 90s you could pay to get a copy of GNU software, if you wanted to get it directly from the *official* source. You could persuade those who already got the copy to give you one gratis. And it would be totally legal. But people love *official* sources so much, apparently, that Free Software Foundation was surviving of selling software in that period.

Not selling, using

Another thing that you may consider, is doing some other business for profit, using your software. For example you can open a store. And sell physical goods there. But for a store to operate you need an accounting software. You can develop it and release it as free software, but in the mean time benefit from it indirectly in your actual business.

Similar to this idea, you can make films with your software. Like videos editors, 3D modeling software and what I did, my own production assistance software. You may make games with your own Game Engine. And write books, articles and other software with your own text editor.

Also you may become a bit more clever and make a platform with your software. From which you benefit directly. Like BitCoin (GPL), LBRY (MIT) and other similar things.

Conclusion

When people say that they need software to be proprietary to make money, I usually feel like it's an insult. Programmers are clever. This is what they do. Programming requires you to be clever. So why aren't they using their cleverness to make money, while keeping user freedom? **Happy Hacking!**

Hamlet - The Allegory for Free Software

Did you know that Hamlet could be used as a good metaphorical piece when describing Free Software?

`lbry://@blenderdumbass:f/hamlet-allegory-for-free-software:9`

To Free or not to Free? That is the question that so many people have today. With the popularity of Creative Commons licences and Free Software, a lot of artists contemplate an idea of whether they should allow other artists to adopt and use their work.

I've recently seen a very epic movie by, soon to be, my favorite director / actor Kenneth Branagh.. It was a 4 hour long, 70mm film adaptation of a famous William Shakespeare play, Hamlet. And in my opinion, not the movie it self, by it's existence, could be a very good allegory to what we call Free Software. Software that respects users Freedom.

Kenneth Branagh didn't write the dialogue

The writer of the 4 hour long epic is credited to be Kenneth Branagh. The director of the film. And the one who played the Prince Hamlet. He did made a lot of very good work on the screen play of this film. But the story, the characters and even the dialogue, the words these characters are saying, are not his own. The dialogue is written by non other, but William Shakespeare.

Kenneth Branagh was Free to do anything with the play Hamlet. Since it's currently in public domain. The play is Free. Libre. Available for anybody to do anything with it. So what Kenneth Branagh did, was to fork the play, and adopt it to the cinema. For which, I may add, he was nominated with an Academy Award of best writer.

Even though Kenneth Branagh didn't write the story, or the dialogue. His additions and subtractions to the work were no less noticeable. And no less worthy of even the Academy Awards. His film is epic and noteworthy even though most of the writing was "stolen" from somebody else.

Kenneth Branagh is a pirate, in today's sense if that word. William Shakespeare should have been the only person benefiting for his work. How dare Kenneth Branagh take his work and make a very expensive 70mm drama movie from it? But he did dared. And he made a work of art. Based on another work of art. Calling Kenneth Branagh a person that violently attacks ships, for that, is more than silly.

William Shakespeare didn't write the plot

I carefully chosen to say that William Shakespeare written the dialogue. But I didn't say that he written the plot. It's widely known that Hamlet and other works of William Shakespeare are based on other works.

William Shakespeare took plays and stories that existed before and modified them to become his plays. He forked stories that other people written and that were free. And added to them his brand of wit and well-spoken intensity. William Shakespeare is no less of a "pirate" than Kenneth Branagh.

Freedom of modification lead a story that otherwise would be forgotten to become one of the best stories. Because a person that knew exactly how to improve it was free to do so. A movie was made from it, that was weirdly very enjoyable, because a writer / director, known for bringing energy to old stories, took upon himself to adopt it even further, into his own work.

Prince Hamlet, Ofelia and other characters, the story of a ghost and madness of a depressed individual, stayed with us and adopted through centuries, because they were free to do so. If Disney was back when Hamlet was written. It would die a death of time.

But sadly Kenneth Branagh's film is not Free

Probably not due to the director's intend, but rather to the contractual obligation with people who gave money to the production of the film. The adaptation of Hamlet by Kenneth Branagh has one noticeable difference, compared to the play by William Shakespeare, or the numerous sources that existed prior. The film is not Free. The film is copyrighted. And if anybody would want to adopt the film, not the play, they would, most likely, be sued by who ever holds the copyright.

The reason of why it is, has to do with that the source material and the play were not copylefted. Copyleft is a way of making a peace of work, in such a way that if people adopt it in any way, they have to allow further adaptation. Usually, with strong copyleft licenses, it says to release the work with similar, if not the same, license as the original. Preserving the work's adaptation-ability. Preserving it's Freedom.

If the sources were released, let's say under Creative Commons Attribution Share-Alike license. And all of it was done in short enough period of time, so the copyright would cover all three versions. As you may know, copyright is not forever. Then if William Shakespeare wanted to release his play. He had to use the same exact license. The Creative Commons Attribution Share-Alike. Which means that if Kenneth Branagh would want to make a movie out of it. He had to release the movie under the same license, the Creative Commons Attribution Share-Alike. And thus, we would be Free to do what ever we wanted with the movie too, as with the other sources of it. Unless if we release it. Then there is only one restriction. We have to release it under the Creative Commons Attribution Share-Alike license to allow further re-adaptation.

The copyleft was introduced with the first GNU General Public License (GPL). From the first version, our hero, Richard Stallman, wanted to give everybody the Freedom to do what ever they wanted with the software. But he knew the kind of problem that could happen with it. Movies on famous works of literature were already been made. And even movies about Hamlet were done prior to the GNU GPL. But they all went proprietary. The sources were Free but the adaptations were not.

For Richard Stallman this was a bug. Bug in a license's legal code, not a bug in the software. So he took it upon himself to fix it. To patch the vulnerability. And make a license that will protect it's program's freedom against those who by malicious intent, or simple carelessness, would otherwise make it proprietary.

Of course people with the malicious intents, didn't stop, so further security patches were made to the GPL. First in the GPL version 2, to which executives at Microsoft screamed "virus". And later the GPL version 3, which scared away even Linus Torvalds, for it's utter protection. I mean Linus's business is selling Linux the kernel to developers of largely proprietary operating systems embedded into devices. And they don't want you to be able to install something else there. So GPL v3 is too much for Linus's sake.

But even though GPL v3 is so strong. The Free Software Foundation is urging you to write that your program is not merely on the GPL v3. But on any later version as well. Since GPL v4 would soon be needed, to protect our Freedoms even further.

Conclusion

Freedom is important. And restricting freedom will not give us much good. As William Shakespeare and Kenneth Branagh shown us. Letting people to adopt your work, could be very beneficial. So if you do works, let people adopt your works. If you make software, make it a Free Software.

To Free or not to Free? There is no question. To Free!

Happy Hacking!

Misinformation is Free Speech

Should people be criminalized by saying things that are factually wrong?

`lbry://@blenderdumbass:f/misinformation-is-free-speech:0`

This statement is false. This is a paradox. If it's false, than it should be true. And if it's true, than it's should be false. What is it, true or false? What information is correct in this instance? Neither, true or false, is correct and also both are correct simultaneously. This is a paradox. A paradox that if a person with power would include something worded similarly in his speech, this speech could be called "misinformation".

So many politicians, members of social movements and even some scientists and software companies are trying to push an idea of censoring "misinformation", or as they might call it "fake news". Illegitimizing works that are *factually incorrect*, at least in theory. Most often than not, though, they are trying to censor ideas they personally disagree with. But even if they were for censoring *factually incorrect* thoughts. If this would become a reality. This would be a disaster. Let me demonstrate.

Works of fiction would be illegal

If *factually incorrect* literature would be illegal, most tellings of stories would qualify as illegal. For example a story of Titanic. The movie Titanic. It will be illegal. Since while it recreates factual things about Titanic, it also includes fictional characters aboard the ship. Like the main, Jack and Rose, characters.

Showing, or talking about the story of Titanic the movie, would be illegal. Since you will include Rose and Jack. That are not *factual* characters. But than even if you would exclude them, you may not be able to talk about it either. Since by mistake you can spread other "misinformation".

It was widely believed, among the scientists, that Titanic sunk as one large piece. But since they found the ship under the ocean, they confirmed that what people had seen and told, that the ship broke in half, was true. Scientists disregarded testimony of survivors since their "science" couldn't explain it. And only believed them when they saw it with their own eyes.

Here the fiction and misinformation was those testimonies early on. And scientists were telling the "truth". Until one event that flipped the two. Made the scientists be false and the testimonies true.

If a law was made that made it illegal to say that Titanic broke in half. Than the discovery of it's remains would be illegal to be said. From the other side, if what's illegal is say what's not *factually* the truth. Than hypothesising on Titanic being one peace would be illegal. As soon as this discovery, that Titanic was two pieces, would be made, all those scientists that spread "misinformation" before hand, that it sunk in one peace, would go to jail. Even if before that, it was the *factual* "truth".

Titanic would sink. All people that say that it's broke in half would go to jail, since science doesn't believe them. The factual truth for the law would be what the scientists would say. Later the remains would be discovered and those scientists, who said that it was one peace, would also go to jail too. Since all these year they were spreading *misinformation*. Do you see where the logic breaks?

And this is only Titanic. A story based on true events. Don't even start on what would be with stories that are wholly original.

Science would be illegal

Science is about asking questions and answering them. The questions are called hypothesis and the answers are called proofs. Hypothesis would be illegal. Since they are talking about things that might or might not be true. And are usually not yet proven by current science. So asking a question would lead you to jail. And even trying to answer it could.

A lot of research. Ways to prove a certain hypothesis. Will lead to jail too. Since in psychology, for example, to make the research, scientists may lie constantly to the people they are studying. To measure their responses to various things. Going fully truthfully, telling people only factual truths, would lead no room for experimentation with that information a subject is given. And thus will halt the field of psychology as a whole.

Casual mistakes would be illegal

How often you were telling people something you believed to be true, just to discover later that you was incorrect? And think about what pressure it would be, when each word you say, could lead to jail.

Since you may, by mistake, say something *factually incorrect*.

Or think about how much more pressure you can add to it, knowing that each person has it's own context. Then you have to account for all contexts, and make your word correct is such a way that all people will get only the *facts* from it and nothing more.

Conclusion

While some views, conspiracy theories and other bullshit exists. It is just an annoying rub, compared to the shier nightmare, fighting with it would be. I hope Freedom Of Speech and other Freedoms will be protected forever, despite some companies an individuals lobbying for their persecution. I want to be able to lie openly and be Free to do that. Since I want to be able to speak.

Happy Hacking!

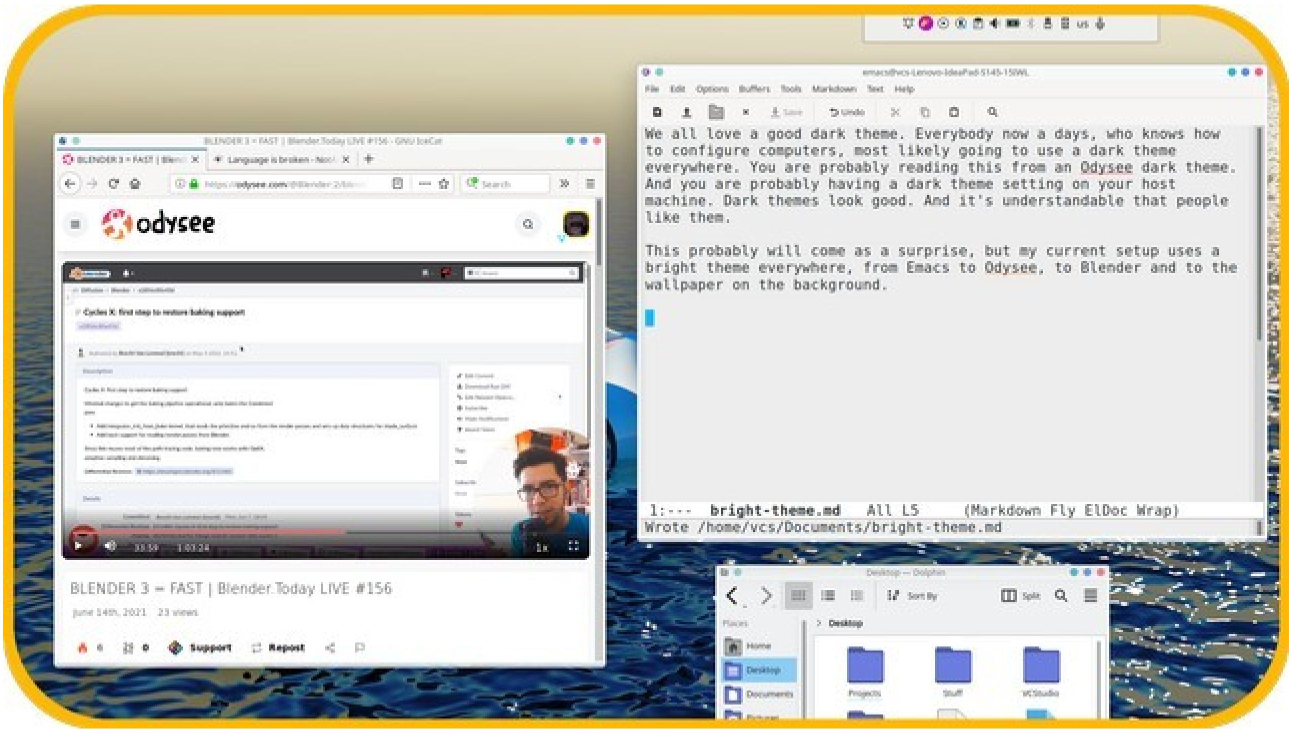
Bright Theme

Everybody likes a good Dark Theme on their computers. How about a good bright theme?

`lbry://@blenderdumbass:f/bright-theme:5`

We all love a good dark theme. Everybody now a days, who knows how to configure computers, most likely going to use a dark theme everywhere. You are probably reading this from an Odysee dark theme. And you are probably having a dark theme setting on your host machine. Dark themes look good. And it's understandable that people like them.

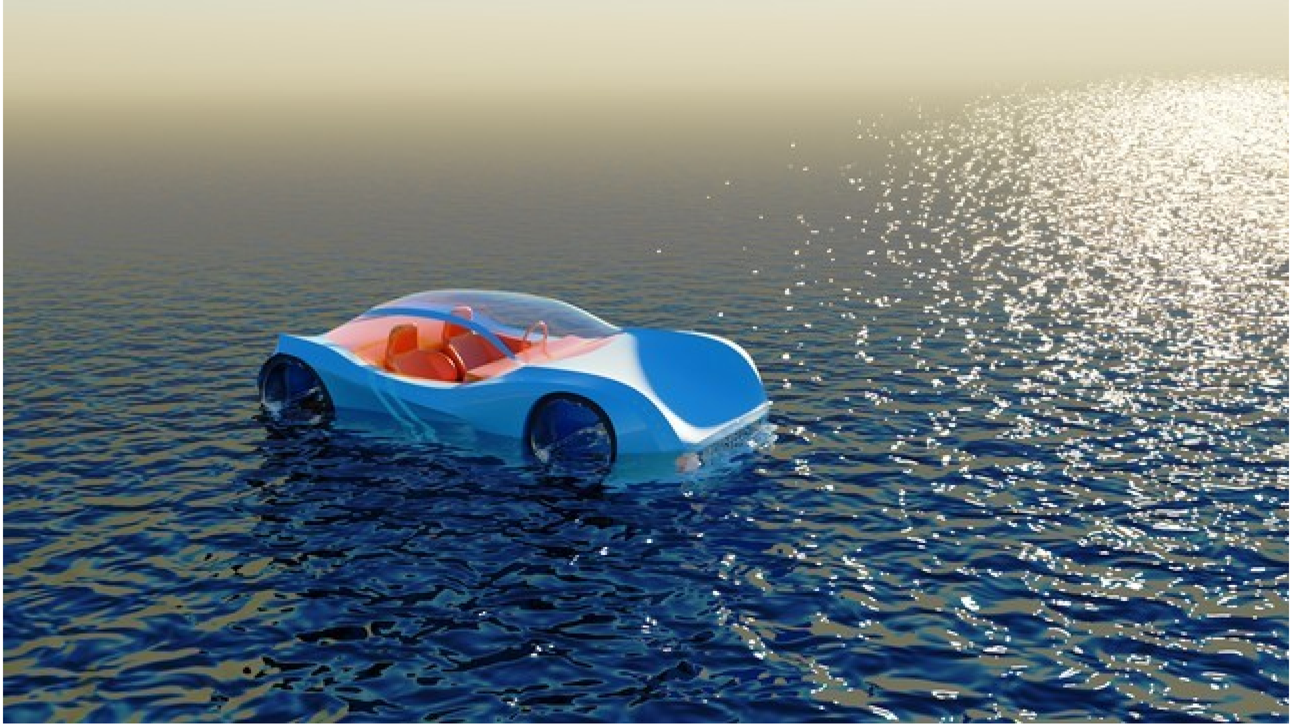
This probably will come as a surprise, but my current setup uses a bright theme everywhere, from Emacs to Odysee, to Blender and to the wallpaper on the background.



```
We all love a good dark theme. Everybody now a days, who knows how
to configure computers, most likely going to use a dark theme
everywhere. You are probably reading this from an Odysee dark theme.
And you are probably having a dark theme setting on your host
machine. Dark themes look good. And it's understandable that people
like them.

This probably will come as a surprise, but my current setup uses a
bright theme everywhere, from Emacs to Odysee, to Blender and to the
wallpaper on the background.
```

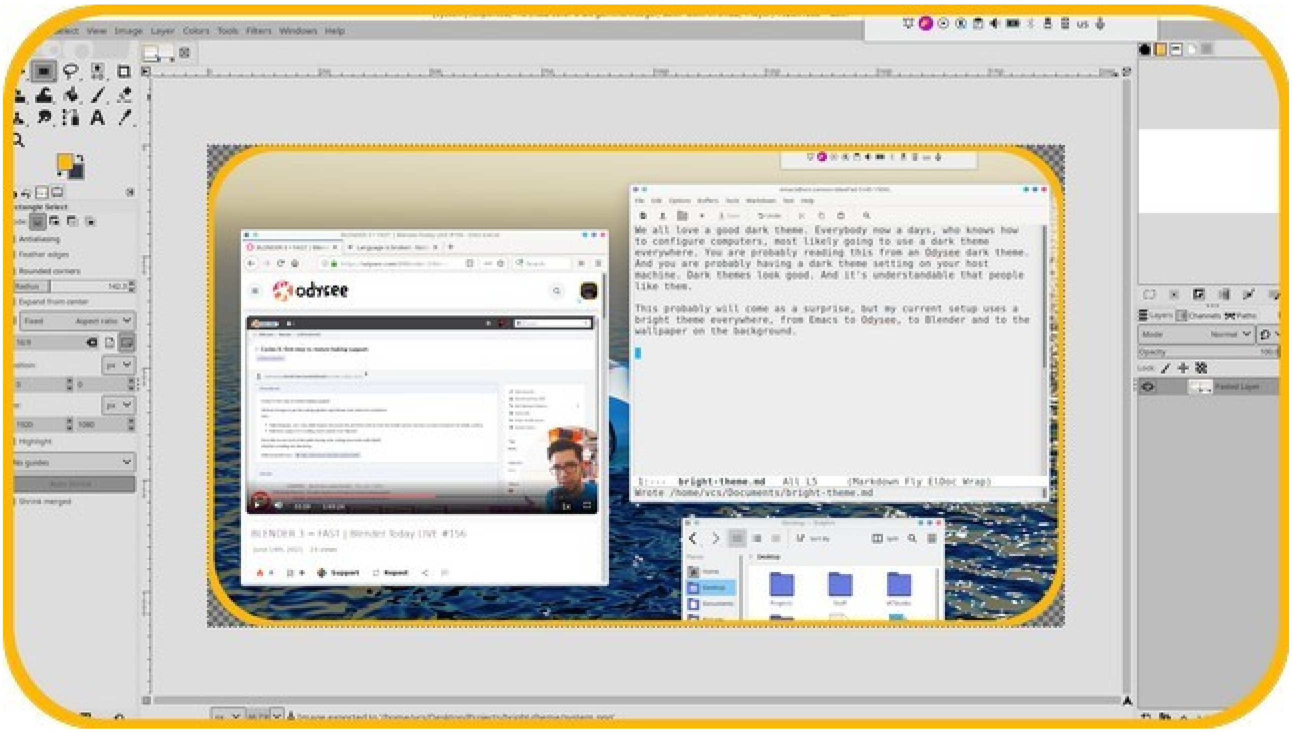
```
1:---  bright-theme.md   All L5     (Markdown Fly ElDoc Wrap)
Wrote /home/vcs/Documents/bright-theme.md
```

This is what my system looks like. All bright. From Emacs on which I type, to Odysee it self. No fancy deep purple. But a white, pink. Who does that?



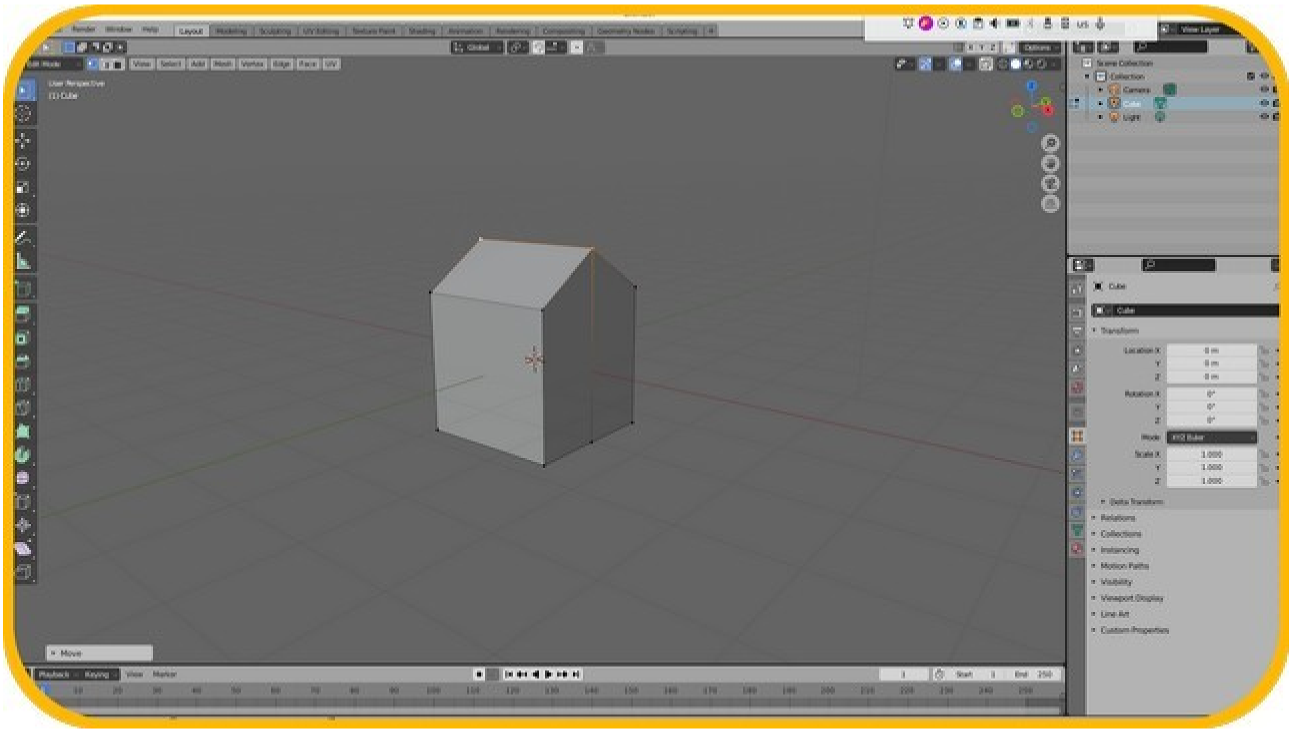
https://notabug.org/jyamihud/WallpapersAndRenders/raw/master/MoriasRacePosters/01.png

Also here is the full resolution copy of the image on the background. Just in case you want to use it, I didn't put the yellow border. You may use it under the terms of CC-BY-SA. I made it specifically for my theme, with assets from my unfinished movie "Moria's Race".



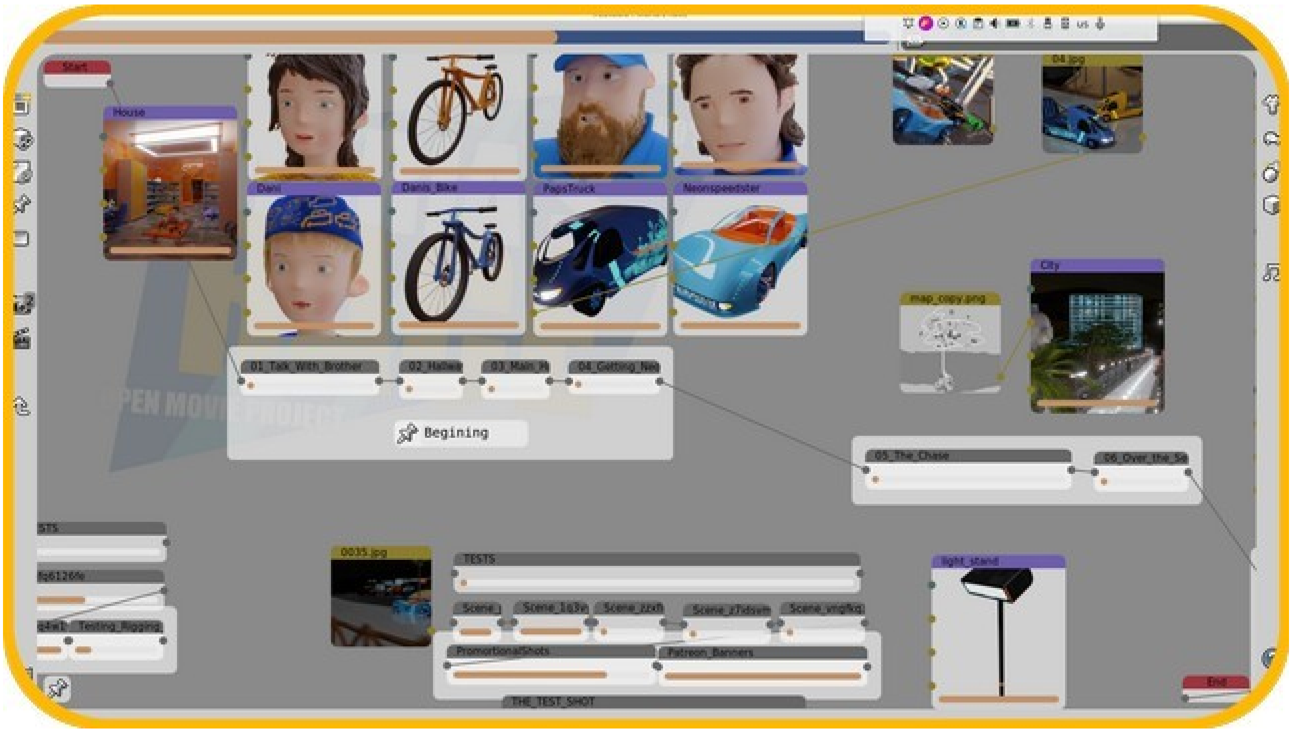
```
We all love a good dark theme. Everybody now a days, who knows how
to configure computers, most likely going to use a dark theme
everywhere. You are probably reading this from an Odysee dark theme.
And you are probably having a dark theme setting on your host
machine. Dark themes look good. And it's understandable that people
like them.

This probably will come as a surprise, but my current setup uses a
bright theme everywhere, from Emacs to Odysee, to Blender and to the
wallpaper on the background.

1:---  bright-theme.md   All L5     (Markdown Fly ElDoc Wrap)
Wrote /home/vcs/Documents/bright-theme.md
```

And this is how I add the yellow borders in GIMP. GIMP that's also themed bright.



Location X 0 m
Y 0 m
Z 0 m
Rotation X 0°
Y 0°
Z 0°
Mode XYZ Euler
Scale X 1.000
Y 1.000
Z 1.000

I wouldn't lie, Blender's bright theme is a bit darker then others. But is still a bright theme. Very bright indeed. You also probably saw me use it in the Blender Dumbass archived channel.



And of course even my own VCStudio is themed bright. To make sure everything is bright. And non of the elements are still themed dark.

But why...?

It seems like there are two answers. One is a bit more scientific. And the other I'll leave till the end.

But in order to get the other, you have to get this first one first.

So what I want to do is very simple. I want to avoid being too familiar with my computer. Because if you know what you are doing exactly, you tend to not experiment. And experimentation is a key in art. You can't do the same movie, or the same book, or the same painting over and over again. You have to change projects. And with them the look and feel.

When I started using Blender at age 12. I started with a version 2.49b. Blender is a hard program to learn. So I took my time. Slowly gathering information to add to my arsenal of tools. And suddenly Blender Foundation hit us with Blender 2.5 with completely redesigned UI and everything.

I was not happy. Since now, after not even knowing one version of that program, I will have to re-learn everything again. At least this was my rational back then. So I avoided Blender 2.5 and Blender 2.51 and so on. Until at age 14, 2 years later, I gave up.

I saw a tutorial about Blender that involved the new video-editor and the tracking system that was added in some of the 2.6X versions. So I had to try doing something with it. I got a copy and installed it.

To my amusement it wasn't even hard to re-learn. It was the same program. But with some things changing places. And with a different theme. I had irrational fear of something I would quite like. I kept using modern Blender ever since.

I'm not a "disto-hopper" but I'm defiantly a Desktop-Environment hopper. And if I don't do that, I hop themes. And KDE makes it very simple to install various themes at will. So I installed this bright one. And changed all software to look fresh.

This accomplishes something. I feel like I want to use it more, since it's fresh. I feel like now I'm not afraid to figure out something difficult. Since it's a new program, with which I'm so familiar. It's not a new program, but my brain perceives it as such, since it's bright and I'm used to seeing it dark.

After a few months on sitting like this, I will change to dark again. Making this freshness comeback to me. And switch between the two, every time the old one is too familiar to me.

The second reason...

If you skipped all of it just to read the second reason, go and read the previous chapter first. Now that you have read the previous chapter. Let's continue. I lied. One reason has nothing to do with the other.

The other reason is that most people use dark themes today and I don't want to be most people. I know it's stupid to decide something in your life because it's against the majority. But what can you do? This is one of those things that makes me tick. I don't have a phone since they are so popular. Together with the zero freedom, zero privacy of it. But mainly what made me so incorruptible to temptations of getting a phone is just one thought. A thought that I will like everybody else.

Maybe is another way of getting to freedom. Since if you are a slave to trends, you are a slave. But then being a slave of the opposite of trends, is also being a slave. So this is why I would like to have a device like Librem 5 when... well... when I'll be rich enough to have one.

When I was 15 and 16 the rebelliousness of me was way stronger then it is now. Which is, to be honest, a trait of many teenagers. I would rebel the norms of everything. Just because they are the norms. Now I find a better way to deal with it. If the norms are beneficial to me and people I love, then I like these norms. But if they either are not beneficial. Or have no effect. Then I will either break them completely, or ones in a while.

I didn't lie after all. Changing the theme benefits me. And I'm free to change the theme. So why the hell not?

Happy Hacking!

Why Free Software is More Secure?

Which one is better? Living in the house with an unknown person that promises not to do anything bad to you. While doesn’t even allow you to lock your door against him? Or living by your self where you control everything including who can lock and unlock doors?

lbry://@blenderdumbass:f/Why-Free-Software-Is-More-Secure:b

A lot of people will tell you that Free Software, or as they might call it "Open Source" is more secure. I have to re-explain it every time. Free Software doesn't mean gratis software. It means software that respects user's freedom. "Open Source" was originally proposed as a substitute term for Free Software. Since "Free" also means gratis in English. Regardless of the name you call it. Free Software is perceived by many as more secure.

But rightly so, a lot of people criticise those statements. Talking about "security by obscurity" and other techniques that cannot work when the source code of a peace of software is publicly available. If I can see how it works, than crackers can see how it works as well. And crack the code.

Let's address security with Free Software. And by the end I hope you will see that Free Software is the only software that we can trust to be secure.

Obscurity doesn't work

Think about two rooms. One that has no apparent door. Only you know what corner to push in order to enter. The other room has a very apparent door.

Metal. Thick. Bulletproof. With steel beams coming to all directions into the walls. To which room would you go to save your self from a person that wants to kill you?

Software that implements "security" by not telling how it works. It's the first room. It's usually a basic algorithm that is not hard to break. The only strength it has is that it's unknown. Free Software for security such as GPG is the second room. It's so obvious that it's unbreakable for those who reads the source code. That they wont even try.

Transparency vs Obscurity

I was learning to be a Rabbi. When I was a teenager I was in a religious, Jewish school. As you may know, Jewish people eat only kosher food. But you probably don't know the extend of the kosher food traditions. And what it takes for a given food to even be called kosher.

There are books upon books of various thought processes and rules that describe food and draws a line between kosher and not kosher food. From how the animal was killed, what ingredients used, to the specific time when vegetables grew and whether the fire was turned by the right person. A huge amount of variables come into telling whether food is kosher on not in the first place. And as you may tell, not a lot of people know all of it. So how do they know whether it's okay to eat the food?

On food that is kosher, there is usually a logo of a kosher certification agency. Here is an article about it from Chabad.org. A quote from the site:

> Please note that while these three are the largest kosher certification agencies in the U.S., there are hundreds of other kosher certification agencies – with varying levels of reliability – and each with its own certification symbol. Speak to your rabbi to determine the reliability of a kosher symbol you may encounter.

As you maybe noticed that the *reliability* of kosher certification agency is very important. It's because similar to software being more or less malware. Kosher agencies could be more or less competent or/and knowledgeable to decide what food is kosher and what food is not.

This is a widely accepted in the orthodox Jewish communities to believe companies that have the largest track record of publicized mistakes. Those companies that tell more often that they found a problem with kashrut of something. But why? Wouldn't it be better to trust those that had no mistakes? Wouldn't it be more kosher, if people checking it, are not having any problems?

And the reason is simple. All certification companies will get into a rough employee or a manager that didn't think through much and made a decision that undermined the kashrut of the product. And all of them will find casual mistakes. Those that are more reliable will speak out about it. To prepare the people not to eat those foods. Risking reputation of themselves and the company that produces the food. But those who are less reliable will shut up about it. Certifying non kosher food as kosher, to save their position in that company. The risk of trusting those with the "perfect" track record is immense. Since a lot of this track record is just lies.

Now let's talk about similar problems with software companies. Free Software with it's source code publicly available, makes it easy to independently verify things. Independently find problems and publish them. And then later fix those problems. And the fixes may come from independent sources, as well.

For companies of proprietary software it's different. They will most likely encounter less bugs, since they have less eyes. But also they have an incentive to shut up about them. And not tell anyone about the bug's existence. Similar to those kosher certification agencies that shut up about the problems with kashrut. They do that to "save" the image of the company and commercial-ability of the product.

Being transparent about how the software works, in it self makes the developers more trustworthy, if you are looking from the right point of view. Since now people will know about the mistakes. And know when the fix is coming. Compared to proprietary software where all you can do, is to blindly believe that you are okay.

Proprietary software has no security against it's proprietor.

Some *features* for some people can be *security vulnerabilities* for other people. For example some people might use services like Google-Storage to save user data in their messaging app. For the developers of the messenger it is convenient. Now they would not need to setup servers. And care about storage sizes. But for the users this might be a security vulnerability. Since all of their photos, videos and maybe even messages, are now stored on a server of Google.

With Free Software, if a developer implements such a feature, this feature can be edited to switch to a different service. Or to use a custom server. With proprietary software you are stuck to what the developers have put upon you.

A lot of those proprietary dis-services, software and operating systems, are claiming security of their products. Whats App, the privacy nightmare messenger, claims that their messaging is very well encrypted and private. Apple make bold claims of security of their devices and operating systems. Letting people know about every little thing any given program is trying to do on a computer.

But think about this for a second. While Apple may make it more secure against other malicious actors.

They are not going to prevent themselves from accessing your devices and data. Similar to how Facebook, making Whats App more secure from other people. While still forces user agreements that makes What App the least secure app against Facebook themselves.

Companies are actors too. And they can be malicious. If they develop proprietary software, they may be very good with security. But you can't know if they are not using it against you. They maybe gonna make the most reliable and secure peace of software, that will protect you from all the other crackers, but themselves. Allowing only one malicious actor to undermine all of the security. The company who made the software. The proprietor.

With Free Software while this technically can happen. It's impossible to continue for long. Since people will be outraged. The software will be forked. And all the malicious features removed. There is no proprietor in Free Software that can control it. It's Free. People, users control it.

Conclusion

Free Software can have bugs. All software can have bug. Since with each new line of code, the bugs are more and more prevalent. With Free Software you can fix those bugs and people do that. While with proprietary garbage, they will shut up, deceive and abuse, because they can.

I want you to read numerous examples of why I don't trust proprietary software here:

https://www.gnu.org/proprietary/

Happy Hacking!

Let's Read Windows License

For those of you that still uses Windows, let me give you an idea of what you are allowed and not allowed to do with this proprietary operating system.

`lbry://@blenderdumbass:f/Lets-Read-Windows-License:9`

DISCLAIMER! I'm not a lawyer. And what I'm about to do, I'm doing it purely for fun. So don't use it as a legal advice. I warned you.

We all love Free Software and Free Software licenses that give users the freedom that they deserve. From MIT and Apache that give people freedom to do anything. To GNU GPL and similar licenses that protect user's freedom, so nobody could take it away from us.

But how often does a user reads a license of a program? How many people have read the licenses of Microsoft products and other software products? How often have people thought about what exactly are they signing up to, by clicking that Agree button?

Today we are going to find out. Because I'm going to read this document. Which is called MICROSOFT SOFTWARE LICENSE TERMS WINDOWS OPERATING SYSTEM on the official Microsoft site. Before we start. I want you to know that I took down my guard to read this document. On GNU IceCat the Microsoft website is broken. I had to use Brave. And it blocked 21 trackers on a page with legal code. Microsoft clearly should not be trusted.

Let's start

The license, after the title starts with the following words:

> IF YOU LIVE IN (OR IF YOUR PRINCIPAL PLACE OF BUSINESS IS IN) THE UNITED STATES, PLEASE READ THE BINDING ARBITRATION CLAUSE AND CLASS ACTION WAIVER IN SECTION 10. IT AFFECTS HOW DISPUTES ARE RESOLVED.

Which are quickly followed by:

> Thank you for choosing Microsoft!

This is very funny to me. Since first they assume that I chose Microsoft. Which I didn't. And second they are warning us about a "Clause in section 10" before saying hi. Imagine you are walking into a store and the employee is smiling at you and says "Warning! You have to notice something nasty about our business. Oh and, hi.".

Later the license goes into various legal things like for example switching blame on the manufacturers of the computers if you buy them with Windows preinstalled. And telling obvious things like, for example "By accepting this agreement or using the software, you agree to all of these terms". Which is just marvelous.

Section 1: Overview.

Section 1 - a. Titled "Applicability" talks about what parts of Windows are under this licence. In short. All of it. Including things that are not software like icons and sound effects. So no luck if you want to use Windows icons on your GNU system. This is probably illegal.

Section 1 - b. Titled "Additional terms" talks about possible additional restrictions they may impose on you with the specific software you may install. Basically letting people to be aware of that various software comes with various licenses. Good job Microsoft.

Section 2: Installation and Use Rights.

Section 2 - a. Titled simply "License" starts with something very interesting.

> The software is licensed, not sold.

Which is the perfect explanation of why you should not trust those proprietary software companies. You "buy" something that never becomes yours. And for them it's clear why. Since they don't want you to have any control over the software. Unlike with Free Software. Where you may do almost everything. (Apart from maybe building proprietary software from it. Thank you GNU GPL for existing.)

Now let's look at something very sinister. They are giving permission to use the software. And then they add:

> for use by one person at a time

Which means. If you are playing with a friend a game which does a split screen. Or watching a movie using a computer with a family. You are probably breaking the law. And Microsoft has you by your balls.

And then they talk about how if you installed a copy that they called "non-genuine", probably referring to people who "pirate" Windows. That this copy doesn't become genuine if you run an update. Sense of humour from Microsoft.

With Free Software all copies are genuine. Since you are Free to give or sell copies to other people. They are maybe not coming from an original source. But they are always genuine.

Section 2 - b. Titles "Device" talks about the word Device. Describing either hardware or virtual machine. Meaning if you are installing Windows in a virtual box you are still under the term of the license. And you are equally screwed.

Section 2 - c. Titled "Restrictions" talks about things you are not allowed to do with the software. Which is interesting to look into. Because a lot of people might not realize that such restrictions even exist in the document they "agreed" to.

> use or virtualize features of the software separately

Meaning you may not use a program of Windows separately from Windows. Like trying to break it apart and use all kinds of .dll files and various programs in your own operating systems. This is why Wine is hard. They can't take Windows parts and use them. They have to build everything from scratch.

> publish, copy (other than the permitted backup copy), rent, lease, or lend the software

Basically denying us the Freedom 3 and 4. To give or sell copies of software. Denying us the ability to help our neighbour. Dividing people. Stopping people from collaborating.

> transfer the software (except as permitted by this agreement)

And even not as copy. Simply giving somebody to use while you are not able to. Like as you may sell somebody a book. It's not allowed. This is insanity.

> work around any technical restrictions or limitations in the software

Basically telling you. That if there is bug, you are forbidden to try to fix it. Of course they are probably talking about DRM and such. But the way it's worded simply tells you that if you have *any* problem with your system. You cannot fix it. Only Microsoft can.

With Free Software you could at the very least hire somebody to fix the issue. And if you know how to do it yourself. The source code is provided to help you do it.

> use the software as server software, for commercial hosting, make the software available for simultaneous use by multiple users over a network, install the software on a server and allow users to access it remotely, or install the software on a device for use only by remote users

A lot of it describes using Windows software as a kind of SaaSS machine. Letting people use Windows from a far with a browser or an API of some kind. Basically making it illegal for you to run any remote control software.

But the other part , "use the software as server software", may have a wider range of restrictions. For example. Simply running Tor nodes, LBRY desktop application or Torrent may get you in trouble with Microsoft. Since you are hosting things from your computer. Turning it into a server.

> reverse engineer, decompile, or disassemble the software, or attempt to do so, except and

> only to the extent that the foregoing restriction is (a) permitted by applicable law; (b) permitted by licensing terms governing the use of open-source components that may be included with the software; or (c) required to debug changes to any libraries licensed under the GNU Lesser General Public License which are included with and linked to by the software;

This is the section that makes me angry and brings me joy in the same time. It talks about not being able to reverse engineer anything from Windows. Which is a very crucial thing if we want to have any level of compatibility with it in the Free Software world. Somebody has to reverse engineer the software first. Then somebody else will make a Free Software alternative for it. But this sections says that it's not allowed to reverse engineer Windows if you Agreed to the terms. Meaning anyone who installed Windows themselves cannot legally reverse engineer it.

From the other side they are mentioning the GNU LGPL license. Right in the Microsoft Windows licence. The GNU Lesser GPL is designed as a bit more push over license than regular GPL. Which is used for libraries of Free Formats such as OGG, PNG and such. So they could be implemented into Proprietary Operating systems, giving people there ability to use those formats too.

Apple for example stays away from these libraries. Since they want complete and total control. Microsoft are a bunch of lazy bastards.

> when using Internet-based features you may not use those features in any way that could interfere with anyone else’s use of them, or to try to gain access to or use any service, data, account, or network, in an unauthorized manner

Basically banning you additionally from trying to hack anything.

Section 2 - d. Titled "Multi use scenarios" also have subsections in it. But all of them are trying to nail on the head one simple idea worded differently each time. You have only one license for one person to use it on one computer. Any multiplications of the usage are restricted. Again. Two people playing a game at ones. Or watching a movie at ones. Need two licenses.

Then it gives a weird restriction on Remote access:

> No more than once every 90 days, you may designate a single user who physically uses the licensed device as the licensed user.

And then follows it by:

> Other users, at different times, may access the licensed device from another device using remote access technologies, but only on devices separately licensed to run the same or higher edition of this software.

So if you want to have a remote control. It should have a separate license for this Windows. Or you can use the remote control only ones every 90 day. This sound like they are coming up with restrictions out of blue by this point.

And Section 2 - e. Titled "Backup Copy" talks about that you are allowed to make only one backup copy of the Windows Operating System. So if you are using Microsoft cloud for it, you already can't copy it for backup on a physical disk. Only one copy, I'm sorry.

Section 3 Privacy; Consent to Use of Data

This is a big one. Since everybody is concerned about it. Let's see what the Microsoft license has to say about Privacy.

> Your privacy is important to us.

Yeah. I totally believe you. Corporate talk 101.

> Some of the software features send or receive information when using those features. Many of these features can be switched off in the user interface, or you can choose not to use them.

This is like telling that some applications use internet. This is quite obvious stuff so far.

> By accepting this agreement and using the software you agree that Microsoft may collect, use, and disclose the information as described in the Microsoft Privacy Statement (aka.ms/privacy), and as may be described in

> the user interface associated with the software features.

Basically they are telling you that they have this link, that has information of what they might or might not access on your computer. Think about why the list is not here... It's because they want it to be changeable at will. When you check it first time, it's gonna be soothing and nice. And then if they need to, they are going to change the terms. If they were honest about it, they could publish these Privacy policies right in the legal code. But then they would have a responsibility to follow what they wrote. Doing it with a link like this strips them from the responsibility.

But even if they are not going to abuse this power, the words "Microsoft may collect, use, and disclose the information" are just too much for me already.

Okay let's load the link and read it. I'm just curious.

> Microsoft collects data from you

Okay. Fine. No need for further reading. This is enough.

Section 4: Transfer

This section talks about different versions of this license for different countries. Since some things are illegal to do in those countries. And Microsoft is going to do those things where it's legal. Bastards.

The Microsoft executives are just asking "What is the maximum amount of user abuse we can legally get away with in this country?" and make a license for that country accordingly.

Section 5: Authorized Software and Activation

This is a weird mess of a section. But let's go over this. First it's says:

> When you connect to the Internet while using the software, the software will automatically contact Microsoft or its affiliate to conduct activation to associate it with a certain device.

Meaning you will always be connected to Microsoft servers while you are connected to the internet with Windows. This is scary already.

> If activation fails, the software will attempt to repair itself by replacing any tampered

> Microsoft software with genuine Microsoft software.

This is frightening. Think about making a change in the operating system. Removing a malicious feature. Just to later have it regrow it's claws on you. Remind me of the scene in Terminator 2. When the T1000 broke to frozen pieces later to melt and reconnect back to start attacking again.

Section 6 Updates.

This section talks about the automatic updates. And that they are forced on you. Also you will have no choice of who you are downloading the updates from. It's going to be only Microsoft. With Free Software, you may change the server from which the updates are coming. And also:

> By accepting this agreement, you agree to receive these types of automatic updates without any additional notice.

Which is just nasty.

Section 7 Downgrade Rights.

This section basically states that you may use a previous version of Windows as long as it's still supported by Microsoft. Windows 7, if I remember correctly, stopped being supported a year ago. So you can't use Windows 7. And only Windows 8. Also they are not required to give you the copy.

Section 8 Geographic and Export Restrictions.

This section briefly talks about a restriction policy. So if they don't want you to use it in some country. You can't take your computer to that country. And provide you with a link to see the current policy of restrictions.

Section 9 Support and Refund Procedures.

Section 9 - a. Titled "For software preinstalled on a device" talk about a strategy they use quite often. In the late 90s there was a huge protest of the GNU / Linux users under the Microsoft headquarters. They wanted their money back. Since they were forced to buy hardware preinstalled with Windows to simply delete it and put a better operating system instead.

But Microsoft shown them that little section in the license. Saying that they need to go and complaint to the store that sold it. And the store didn't want to give them the money. So they used that following thing.

> If you are seeking a refund, contact the manufacturer or installer to determine its refund policies. You must comply with those policies, which might require you to return the software with the entire device on which the software is installed for a refund.

Basically the store might not give you the money if you want to keep the hardware. You have to return the computer fully to get a refund. So Microsoft is basically charging money from GNU / Linux users using this section.

Section 9 - b Titled "For software acquired from a retailer" is talking about "limited support" that they offer to people that bought Windows without hardware. And that this can be refunded directly with Microsoft.

Section 10: Binding Arbitration and Class Action Waiver if You Live in (or if a Business Your Principal Place of Business is in) the United States.

This section starts with a very scary sounding lawsuit threats if you live in US or you have business in US and you are using Windows. And then talks in dept about how lawsuits are handled. To have potential battleground against anyone who wants to sue Microsoft I guess.

As I understand it. They are trying to make it hard for people in the US to have any legal power against Microsoft. So if they ever used Windows. While living in the US or having business in the US. They have to follow more steps to get some legal stuff done. This is plain evil in my opinion.

Section 11 Governing Law

This section tells you that laws exist.

Section 12 Consumer Rights, Regional Variations

This is once again letting you know that there are differences of this license for different countries based on law. This time giving examples. I'm not going to go through all of it. It's boring.

Section 13 Additional Notices.

This section illustrates various differences of various libraries and software preinstalled with windows like video codecs and antivirus software. That may have separate terms. And they brief on those terms.

No Warranty

And then they are talking in dept how they are not responsible if anything breaks ever. So you will not try to fight with them under any circumstance.

Conclusion

Don't use Windows. Use GNU / Linux. Even the worst GNU / Linux is not nearly as nasty.

Happy Hacking!

Future Proprietary Software Licenses

Now let's look at what a proprietary software license of the future would look like.

`lbry://@blenderdumbass:f/Future-Proprietary-Software-Licences:0`

Yesterday I made an article about all kinds of nasty and evil things written in the Microsoft Windows End User License Agreement. Their license is so evil that I have an idea of how a future proprietary software license will look.

Here is a draft of their future legal code:

By having a copy of the software, reading this text and or living on the planets Earth or Mars, you agree to the term of this agreement. To not agree, you must not have a copy, not read a single word a this agreement and not live on either Earth or Mars.

TERMS:

1 - Slavery

You agree to provide your labour to us at any moment we so choose. It could be physical, psychological, sexual or any other kind of labour any employee of our company may want to have.

2 - Law Suits

You agree to never file a lawsuit against us. If you do, you agree to death.

3 - Consent for Surveillance

You agree to give us all your data. If we fail to collect some data using our technology, you are now obligated to submit this data manually to us. Lying about any data, using Tor, VPN or any other anonymization service will result in death.

4 - Above Law

You agree to never complaint about us to the law enforcement. If an employee of our company does anything considered illegal, he or she are should not be reported. If you are a police officer and you are about to arrest an employee of our company, you should not proceed. Any violation will result in death.

5 - Software Warranty

This software has no warranty. It's not even supposed to boot or work or do anything. Our business is not responsible. We may or may not do our job.

6 - You have to pay

You agree to a daily payment in arbitrary amount. Those payments will be transferred from your bank account automatically. We will know the data about your bank account as mentioned in section 3. If you have no bank account, you will have to provide the payment manually, in cash, with travel costs at your expense.

Don't use proprietary software. This is literally what they are always doing.

Happy Hacking!

Using Telegram Client For Other Services

Telegram has issues, but in the same time it has a nice UI that we can use since it's Free Software. Where could we use it?

lbry://@blenderdumbass:f/Using-Telegram-Client-For-Other-Sevices:8

Telegram is a service that has divided a lot of people. From one side the Telegram client is under a good Free Software license (GPLv3) and from the other side it claims to be secure and private, while doing the number one sin of insecure messengers. Storing data on a server, somewhere, unencrypted. More than that, their server software is proprietary. And they are co-developing, using wholly original code, to avoid GPLv3 lawsuits, a different client. Telegram X. That is wholly proprietary. Meaning if you are using Telegram X, there is no Free Software anywhere in sight.

Despite all of that I am not feeling too bad about Telegram. Usually when I talk to people about Free Software, to illustrate my point, I ask them "Do you have Telegram?" which gives an answer "Yes" most often than not. And from there I can explain myself further. Drawing, for example, a comparison between Telegram and WhatsApp. Showing similarities. And giving them an understanding that the same features could be done separately by separate developers. And some may release it as Free Software. Software that respects user's Freedom. From that point giving them more examples, GNU / Linux, Blender and others becomes easy.

But since separate developers can develop same features separately. Why wouldn't there be something like Telegram but without the nasty parts? And there is. From IRC to Jami, Tox, Martrix and Signal. Other protocols and Free Software apps with them exist. Only there are some problems. And I want to address them.

My brother hates KDE plasma

My brother is by no means a pure Free Software user. He is trying. He switched finally from Twitch and Youtube to Odysee, as a publisher. Not as a viewer yet. He tries playing Free Software games on GNU / Linux. But also plays proprietary games on Windows. He has a dual boot on his machine. With 2 separate physical disks for 2 of the systems. And while he hates Windows with extreme anger. He still uses it more often than the Free Software system available to him.

He has a basic, normal installation of Ubuntu. With the basic, Yaru theme that comes preinstalled with it.

All that he has changed on the system, was to switch the desktop background image to a black color. Minimizing impact of desktop on the performance.

I actually trust him to render things. On my 2 movies he helped me with the rendering optimization. He was the main brain on what exact things to put in the Blender render settings to render the frames faster. Even though he is a noob in Blender. He is an expert on Blender render settings. Also he helped me with fine-tuning the algorithm for the rendering via VCStudio, our production assistance software.

But strangely enough, even though he knows that KDE Plasma is faster and more configurable than Gnome, he prefers Gnome. When it's time to render something, he doesn't bother with graphical interfaces at all. He presses Ctrl-Alt-F2 to get into a terminal only mode. And we render through there.

But why doesn't he like Plasma? From what I understand. Is that plasma is another graphical interface. Another thing to learn. Another thing to get used to. While he is already used to Gnome and Terminal. He is not used to Plasma. It's the same problem that so many people have with GNU / Linux to begin with. They are used to Windows. To how it looks, feels and abuses them. If a change is made, it's usually too significant, so they might not even try to use that software.

In my article about Bright Themes I touched upon my own struggles with the same thing. I was used to the UI of Blender 2.49 and couldn't transition to 2.5 for about 2 years. Breaking this fear lead me to realization of how cool it is to have more options. And made me into a Desktop Environment hopper. Now I love all kinds of different Interfaces. New things fascinate me.

But most people didn't break through it yet. And this is why using an IRC chat suddenly becomes kind of terrible for most people.

Also, polish is a big factor in what people will use. Telegram's UI is quite polished. Even Signal is not as polished, compared to Telegram. But most IRC clinets are a joke in today's world. So something must be done. But what could be done?

Telegram Client is Free Software... Hm...

The Telegram Client is Free Software. Meaning that Telegram's familiar to so many people UI, is Free Software too. Meaning we can fork Telegram, remove all what makes it connect to Telegram servers and instead make it connect to let's say IRC. Bringing all the IRC concepts to a program that looks and feels like Telegram, with all of it's polish, themes and animations.

Imagine that on the left, where used to be your Folders, it's instead various IRC servers. After that, where used to be your contacts and group, it's the various Rooms in the IRC. And the chat is chat. Looks the same, acts the same.

If somebody sends a link to an image, it's previewed as if somebody sent an image in Telegram. If you want to send an image or a file. It might use, I don't know, LBRY, to store the image. I know not the best idea since you'll have to wait for it to confirm. But maybe a different service could be used.

Calling? It can send the other person a Jitsi Meeting link, like it's done in Rocket.Chat. Jitsi is Free

Software and it can be re-implemented in Telegram's UI.

A person using this kind of IRC client will not even feel like using an IRC client. He will feel like it's a good old Telegram.

Maybe even if we don't remove the Telegram features, and just add the IRC to it, it can be very powerful. People will have an option to connect to IRC right through Telegram. Meaning more possibility to communicate with people that otherwise would not join the servers. And I think a pull request like that, could potentially end up in the "official" Telegram client. Why would they reject it?

But also there are Tox and Jami and other protocols that have file sharing and that have calls. That are easier to implement. And could be implemented. Maybe Telegram's UI could be used to teach so many people about the other protocols. About Free Software. Just by forking it's UI and building something that looks like it.

Other Software UI

While a lot of Free Software uses standard GTK or QT to build it's UI. A lot of other Free Software build their own UI. Blender, VCStudio, Ardour, Telegram, Rocket.Chat, LBRY and so many others. And while I already talked about using Telegram's UI to build something else from it. Maybe we can also use UI from those other programs, to build something.

And it doesn't even have to be similar in concept. Blender's UI to build a Web Browser. LBRY UI to build a file manager. Ardour's UI to build a Chat Program.

And it shouldn't even be only UI. You can take any feature from one Free Software and copy paste it into another Free Software. The possibilities are endless. Why aren't people doing such crazy things?

Conclusion

Free Software unleashes potential creativity that proprietary software are too afraid to give. So let's use it. Let's build things. Let's show those who uses proprietary software, how pathetic they are, for not being able to use IRC in a Telegram client.

Happy Hacking!

Let's Read GNU General Public License Version 3

We already looked at the nastiness of the Windows license. Now let’s look at the goodness of a nice Free Software license.

`lbry://@blenderdumbass:f/Lets-Read-GNU-GPL-3:1`

DISCLAIMER! I'm not a lawyer and I don't know what I'm doing. Also it's not the license it self. It's only a review. So don't use this document as a legal advice. You've been warned.

A lot of people have opinions about the GNU GPLv3. Some find it too strong in a copyleft. Making it "unusable" in commercial world. But I think this is why GPL is so good. Free Software is not about market dominance, it's about user freedom. And GNU GPL is there to protect it.

Couple of articles back I reviewed a license of Microsoft Windows. And it's utter evil nature. You have to read it to believe it. It's insane. They are literally making their license so they could do anything to you, and you could not do anything in return to them.

But let's now look at the GNU GPLv3. And discover why proprietary companies hate it so much. While I and so many other people love it so much.

Preamble

If you remember the Window License review it's started with a scary sounding warning. GNU GPLv3 after the copyright statement starts with the following words.

> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Which is good. This is a very strong legal document. Changing words in it may result in it having no power at all. You may write a license yourself. Copyright it to yourself. And make any change. But it will not be GNU GPLv3 anymore.

Making only verbatim copies of this document makes it possible to just say GPLv3 and nothing else. Literally every copy of the license is the same exact thing. So when you are looking for GNU GPLv3, you will find the exact legal code you were looking for.

Then it follows with:

> The GNU General Public License is a free, copyleft license for software and other kinds of works.

I love this statement. GNU GPL is not only for software. You can publish anything under this licence. I remember seeing plans for soft toys that were under GNU GPL. And also it's interesting what would happen if somebody would release a movie under GNU GPL.

> The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users.

Well. I guess this is what inspired me to make the review on the Windows License in the first place. I mean. Were you ever inspired by a legal code? I guess GNU GPL has this unique ability to inspire.

Then, the license teaches the reader about Free Software. That's it's not about price, but rather about user Freedom. And briefly talks about what this license is designed to do.

> Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

And after teaching the reader about Free Software, it teaches the reader about Copyleft as well.

> To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights.

And it proceeds into explaining how copyleft works. Note: That this is not Terms yet. It's just a Preamble. A kind of Read-me section. A briefing on the reasons why the Terms will be the way they are. A briefing made in such a way that people that read only the GNU GPLv3, not knowing what Free Software is, or what license is, or what Copyleft is, will know what it's all about and for.

Section 0: Definitions

The main problem with GPLv2 were lack of strong definitions. GPLv2 was written in the US with the help of the US lawyers. And some things were worded is such a way that people in other counties understood them differently. So this section 0 is here to clarify what some of the words mean.

This section will define variables to use later in the legal code. Words like "Copyright", "The Program", "modify", "covered work", "propagate" and "convey" will be explained and defined. So there would be no ambiguity about the meaning of those words.

These concepts were the main attack surfaces for the lawyers of proprietary software companies, that were trying to get away with use of GPL covered code in some proprietary program or device. And they were usually misrepresenting their meanings. So a concrete and set in stone meaning for each of those words is provided in this section.

Section 1: Source Code

This section clarifies the difference between the source code of a work and object code (binary executable) of that same work. So a judge will be able to understand the difference.

> The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

I like how it's not trying to go all software technical in here. But rather decides to word it as "the preferred form of the work for making modifications to it". For example. JavaScript that's sent into your browser in semi-readable "source code" which has no comments, random names of variables and written in one long line. This will not going to be considered "source code" by the GNU GPLv3. Since it's not "the preferred form of the work for making modifications to it".

Then it defines few more things. This are not normal words like "The Program" or "modify" that had to be explained in section 0. These ones are a bit more technical in nature.

They talk about "Standard Interface", the interface that's standardised. Say HTTP or something similar. "System Libraries" which are libraries that implement standard interfaces for more then one program. Like libogg or libpng. Or even things like the kernel of the system. These are parts that make the program work, but are not really parts of the program. Dependencies, if you will.

Then it talks about "Corresponding Source" as in source code and any kind of documentations and scripts that may be needed to install and compile the program. Turning it from source code to working object code.

Corresponding source doesn't need to include the System Libraries though. With GPLv2 it wasn't stated, so sometimes you may get a problem with it. Think about how much detail they went through to save those few people that use too much dependencies.

Also it should contain all files that may be needed to be used to build the program, like "interface definition files" that are used by system libraries to draw UI. And other stuff similar to it.

Section 2: Basic Permissions

This section is talking about what you can do with a program. It's stating the 4 essential freedoms in a more legal, bureaucratic way.

You may use the software when ever you want for what ever purpose. And the license is irrefutable. So if I release something under GPL and you got a copy, I can't tell you that this license no longer holds value.

Then it draws difference between "propagate" and "convey". Propagate meaning copying the program as is. Without modifications. Which is allowed without needing to give the source code. Many people torrent Free Software. And many people set up software repositories with only object code. It would be silly to ask all of those people to provide the source code as well. The source code is already available, from the developers who made the version.

Then it talks about hiring people for changing your copy. And that it's different than releasing modified versions. Since you may or may not release your modified version. Even if more then one person is working on the modification.

Section 3: Protecting Users' Legal Rights From Anti-Circumvention Law

There is a law against breaking DRM systems. Those who make those DRM systems can file a lawsuit on those who breaks DRM and it can be bad for those people. Years of prison. You can technically implement DRM using code under GNU GPLv3. But since it's under GNU GPLv3 you should be able to edit the source code. The problem is, that it can be seen as breaking the DRM. So this is a trick a proprietary software companies can use to make it illegal to edit sources of their software using GPL code in it.

This section states that they loose all legal power with such laws if they use GPL covered works for such systems. Any "Anti-Circumvention" law what so ever. Not only DRM related. Anything at all. If the implementation has GPLv3 code in it, it's now legal to Circumvent it.

Section 4: Conveying Verbatim Copies

This section says that you are free to give people exact copies of the source code if you keep all the legal stuff intact.

Section 5: Conveying Modified Source Versions

This section describes conditions if you want to give people modified copies.

> The work must carry prominent notices stating that you modified it, and giving a relevant date.

You have to tell people that a change was made and when. So they could compare it to other versions. And that they would know that it's a different version. Using Git solves it.

> The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

If you release a modified copy, you need it to be under the GNU GPLv3. And you need to change some legal stuff. As explained in section 7, I guess.

> You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

The whole software project must become GPLv3. Unless you was provided with a permission to do otherwise from the original developer. Who can write for example "Use under terms of GPLv3 or any later version" allowing in future to use GPLv4 for the entire project. By the way this is advised to say "any later version" when releasing GPL software. Since GPLv4 may appear soon.

> If the work has interactive user interfaces, each must display Appropriate Legal Notices;

> however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

If you have an about page, or a legal page in the program somewhere. Let people know that it's GNU GPLv3 so they would know. And you cannot write that it's on a different license. Or proprietary. Since this will confuse people.

Then it says that if a work is not compiled together with the bigger program to which it is a part of. It's not supposed to make this bigger program to become GPLv3. For example if it's a script to download things from LBRY. It doesn't make LBRY be under GPL.

Section 6: Conveying Non-Source Forms

This section is talking about conditions for releasing the software as a binary form. If you have GPL covered code in it. You need to provide the corresponding source in one of the following ways.

If you are releasing it on a physical medium. Or inside a device. You can add the source code on a disk somewhere. Or a storage medium. As a separate thing but in the same package. For example. You buy a phone and you have an SD card with the source code.

One other option is that you can make an offer to get it some other way. For example you buy a phone and you get a link. That will let you download the source code. In could be only you. And this offer should be valid for at least 3 years since the object code (phone) was acquired. This offer should not require any additional payment. The source code could be sent over the network or on a physical medium.

You may give or sell the software with the source code through the same place. Like having the same link where you can get both. You may even store them of separate servers as long as it's clear how to download the source code.

Or send it directly to the user personally.

Then it talks that System Libraries. That the program interact with and uses. Could be not parts of the source code. Since they can be obtained some other way.

Then it touches on something very interesting. If you are making a device with software under GNU GPLv3. You have to provide all information, so the user could install his modified version. And that the mere fact that the software was modified, should not break an ability to install it.

Section 7: Additional Terms

This section just makes everything a bit more clear. Going more in depth to make sure that this license is unbreakable.

Section 8: Termination

This section talks about punishment for those nasty people who do not want to share with us the source code of their modified copies. And how they can redeem themselves if the license is terminated.

Section 9: Acceptance Not Required for Having Copies

This section is interesting. It says that in order to simply use the program, you don't have to accept it. You are not required to agree to anything at all. You need to agree only if you wish to share copies or edit it.

Section 10: Automatic Licensing of Downstream Recipients

This section says that the software is automatically licensed under GNU GPL for those who get a copy from you.

Section 11: Patents

This section says that if you have a software patent that you implemented in a GPL covered work. You are, automatically, licensing the use of the patent with the software it self. And there are 3 paragraphs of Patent license there as well.

Section 12: No Surrender of Others' Freedom

This section basically says that if any other law, court order or license may contradict with this license. It doesn't excuse you from not obeying the license. You better not release the software at all then release it without source code. Even if the court says that you have permission to release it without source code. You still can't do it.

Section 13: Use with the GNU Affero General Public License

This section says that you are permitted to link and use code under this license with software under the GNU AGPL.

Section 14: Use with the GNU Affero General Public License

This section explains that FSF may release a newer version of the license. Like GPLv4 or GPLv5 and so on. And you are free to use any version unless the developer specifically mentions a specific version.

Disclaimer of Warranty

Since it's Free Software and everybody can edit the code and add or delete anything. Section 15 is designed to provide protection for those people who makes edits. So no warranty comes with the software unless stated by the developers.

Section 16 talk about this same concept in relation to liability. Meaning that no one is responsible if the software had corrupted valuable files or had other bugs.

How to Apply GNU GPLv3?

Then the license ends the terms and shows and example of what you can add into the source code files to make them become GNU GPLv3. And gives examples of hypothetical commands to show the license in the terminal applications.

Conclusion

Please use GNU GPLv3 (or later) in your programs. This will make a change. Since it will fight simultaneously with proprietary software, DRM and software patents.

Happy Hacking!

Blender Can Compete But Doesn't

Blender is a weird beast. The developers are smart enough people that they could, if they wanted to, to make a full replacement for Autodesk products. But they chose not to do it.

`lbry://@blenderdumbass:f/Blender-Can-Compete-But-Doesnt:5`

May 2002. A project that was developed in house for a little, now dissolved studio in Netherlands, is trying to survive. It's core developer is trying to find money for the project, to keep development active. He is not concerned with money. He is not trying to make the most commercially successful software out there. He is trying to make *good* software. And up to this point he made something that he finds "good enough". Something in which he sees a potential. Something that needs further development.

It was a year after Window XP. Two years before Ubuntu. Six years before Github. And seven years before Kickstarter. And yet the developer of this program tries to get funding for this project from the people on the internet. Using his own website.

He decides to release the program's source code under the GNU General Public License Version 2 or later. The latest version of the GNU General Public License at the time. Turning a private program into a Free program. This is going to help him develop it faster. Since now anybody can join the development.

Few month later, at September 7th, 2002, he has €100,000 from people on the internet. The crowdfunding worked. The development may continue. To this day, almost 20 years later, the development never stopped. Release after release. Feature after feature. The program that ones was Freed, became a powerhouse of abilities.

Blender. A Free Software program for 3D modeling, animation, rigging, rendering, compositing, video editing, simulating, sculpting, planning, camera tracking, chroma-keying, roto-scoping, scripting, designing, painting, ray-tracing, path-tracing, generating...

The list may never end. Since features will never stop being added. Since Blender will never stop being developed. And yet with all of this glory, so many people still avoid using Blender. And prefer using proprietary software like Z-Brush and Maya instead.

It is true that Blender has issues. Z-Brush is better at handling large amounts of polygons. Maya has more polished animation tools. And even though Blender Developers are trying to please as many people as they can. They still can't do all of it at ones. There are still problems in implementing some features that users want.

Copyright, Patents, Reverse Engineering clauses.

On January 31 of this year, 2021, that same developer, now turned chairman, or as some may call him a Benevolent Dictator of Live at Blender, Ton Roosendaal, posted a post on Blender's devtalk forum, titled "Copyright guidelines for devtalk". In this post he is talking about people, who post screen shots of proprietary software programs, in an attempt to persuade developers, to implement those same features exactly, as they are implemented in those other programs. He argues that it's not very good for Blender to do so, since it's may result in lawsuits from the companies developing those, proprietary alternatives.

A few articles back, I was reviewing a Windows License, in which one clause was, that if you agree to it's terms, you are not longer allowed legally to reverse engineer the software. And using the software is enough to *agree* to the terms. Not only Windows uses such clauses.

Take for example text editors. There are plain text editors like Emacs and Notepad. And this was all it was, for a very long time. The font and the size of the text was configured when printing, not when typing.

But then think about an executive in a software company, some time in the 80's, trying to make a fortune, on those who use computers for typing. He may make a good editor. Such as Emacs. But there is a problem. If people can edit the file in any editor, why would they stay with this particular editor? The file format should be different, so it should be supported only by this new editor.

But then, why would they use a file format that is not supported by their currently favorite editor? Incompatible format has no chance to stay relevant. But what if it's not a text format? What if it's text, but extended? What if they design a format that will be proprietary, but add features to it? Features like fonts, colors, page settings and so on...

Hooking people onto software like this, is called Vendor Lock-in. It's an attempt to make people use only one Vendor, only one product, service or software. It's what Google does with their library of dis-services, preinstalled on every android device. Making you dependant on them. It's what Microsoft does, by making proprietary formats, that can be read and edited by only their proprietary software. And this is what 3D packages do as well.

In order to break away from Vendor Lock-in, somebody has to reverse engineer the format, or the service. And build an alternative. Which is completely legal on it's own. Unless you agreed not to do it in a formal way. Which those companies will sneak on you in their "Click Agree To Continue" pages.

Blender needs compatibility with a lot of formats designed for Vendor Lock-in, in order to be useful for many people. Importing and Exporting models to and from formats of proprietary 3D packages. And sometimes people just want a feature, that exists elsewhere, but not in Blender. And so they scream and shout about it. While Blender Developers are not very allowed to just simply implement those features. Some lawyer, somewhere, is waiting for this to happen.

A lot of legal battles should be won first to implement some of those features. Maybe somebody is willing to do it, first, in their own little Free Software project. Maybe somebody is careful to reverse engineer a program without agreeing to it's terms. Which most likely means, without ever running it. Or, maybe the features can implement the same solutions, designed differently. Features that solve the same problems, but in a unique way. The Blender way.

Vision

Benevolent Dictators of Live are good for Free Software projects for 2 reasons. They are the people with the vision. The people who want a unique program that tackles things in a unique way. One way it's good is that it's usually bypasses a need to reverse engineer things. The vision will dictate a unique design more often than not. But the second reason, is that there is a vision to begin with. The software is being treated as a work of art. Which more often then not, results in a better software.

But from the other side it can be a problem too. Users might not want to follow the vision of the project. And they are Free do so in Free Software. Ton Roosendaal decided to delete Blender Game Engine with the release of Blender 2.8. His vision of a highly modular Blender didn't combine well with a terribly separate code base of the Game Engine. So he removed it, announcing plans for a redesign that might or might not happen. A redesign that will be more integrated with Blender's core code base. More integrated with his vision. But some people didn't like it. And so they forked Blender and made UPBGE. Preserving the Game engine, while adding all the new features.

There were numerous times where a pull request of a new, fully developed feature is rejected by Ton, because "it doesn't look like Blender". Sometimes the feature can be tweaked to match the desired aesthetic. But sometimes it's just too far from the vision to be accepted. Some developers might not even risk trying to add their features into Blender because of this. And will fork Blender to make their own thing instead. Like what happened with Armory Paint.

A lot of other software projects will implement features for certain things to simply work. This would be their first priority. For example in a feature film, when dealing with a huge scene, of let's say a city destruction scene, the software need to be able to handle all this data. Sometimes it will introduce various interfaces to work with the data. That might not give immediate feedback, or even a preview. But you will be able to use those features to make the thing that you are trying to make.

This is kind of against the vision of Blender. In this article, Blender Developers described this very issue and possible workarounds, to preserve the vision, while developing tools that work. One aspect of the Blender's vision is always being able to see everything changing live. If you drag a slider, it will update the geometry while you drag it. So you could fine-tune things. Or as they pointed out, Blender lets you sculpt while using a full render engine to preview it, that's updating live.

While it's great for the artist to see immediate feedback, this results in Z-Brush working better with higher amount of polygons. It doesn't need to update the whole thing. It just draws the polygons.

Perhaps, a lazy person would suggest Ton Roosendaal to change the vision and focus more on making Blender usable, and less on making Blender immediate and integrated. But apart from Ton's benevolence, he is also a Dictator of Live. And I think it's a good thing.

Blender has a chance, through hard work and tears to achieve higher standards than those software projects that are focusing solely on making features work. Think about this again. You can model and sculpt, while previewing all of it live, using a full render engine. And it works surprisingly fast. Perhaps Blender Developers will work few more years and beat Z-Brush in it's performance. Keeping the immediate feedback. Keeping the full render engine. Making it a marvelous program. They have a chance to achieve things, other software doesn't even dream of.

Conclusion

Blender can compete, but doesn't. Since Blender is not trying to. Blender is not about being the most commercially successful program. It's about being a good program. Something that will make it's

Benevolent Dictator or Live happy. Something that the developers will be proud of. Not something that feels like a cheap trick, to win in a pointless race.

Happy Hacking!

Blender Is Focused

A lot of people criticise Blender for lack of focus. But in my opinion there is plenty of focus in the team that is making Blender. It’s just not in the same areas that people want it to be focused in.

lbry://@blenderdumbass:f/Blender-Is-Focused:e

Not so long ago, I wrote an article, about why Blender doesn't out-compete the proprietary offerings in functionality. Some of the comments I received, were telling me that Blender is not focused. The argument was that by focusing on so many things, so many features, each and every one of those features becomes less usable as a result. And while I agree with this concept in general, it doesn't seems to be related to Blender very much.

Yes, Blender has a lot of features. Similarly to how a lot of features there is on a modern smartphone. You can not only call, but also access internet, do banking, photograph and record video. And so many other things. From a device designed originally for communicating only.

Are developers of Phones lacking focus? No their not. Most of the functionality of a phone comes from apps. They are designed by separate people, focusing only on one specific thing at a time. Together making this huge operating system, full of features.

I don't like phones. And while many other people do not like phones for it's "lack of focus". Preferring specific items for specific tasks, like, camera, map, compass, telephone, computer, etc... I don't like the lack of freedom on modern phones. But if a proper budget arrives, buying a Librem 5 or a PinePhone would be cool. I have no problem with having a tiny computer in my pocket.

But, you may say, "Blender is one app". One part of an operating system. Developed by one entity. You are right and wrong in the same time. Yes, Blender is one peace of software. But it resembles more an Operating System rather then one app.

Blender has an expensive library of addons. All developed by separate entities. Blender has an expensive library of build in Features too. Also developed by separate groups. Yes. Blender's features are developed by separate groups. Groups called Modules. There is a modelling module, with developers focusing only on modelling. There is a view-port module, with developers focusing only on view-port. And so on.

Those aren't groups of developers, because the system is a little bit more flexible. Blender is Free Software. So anybody at all can develop anything.

And so developers from one module might help the development of something that's not in their module. And some developers belong to multiple modules. Communication between modules is required to make a coherent whole, rather than many features, good on their own, but requiring you to re-learn everything every time.

If somebody is a developer that's good in one area, he might only focus on that one area. For example the famous Pablo Dobarro who is developing the Sculpt mode pretty much by himself. He is solely focused on the sculpting as if developing a competitor to ZBrush. He doesn't care about the rest of the Blender. But there is probably some body else that does. And wants to integrate Pablo's work into different other parts of Blender.

For example sculpting is drawing, only with geometry. And Brushes used for sculpting, can be used for painting textures, or painting weight maps. And so one code-base could be used in multiple places.

They are planning a new Asset Creation Pipeline Design which is going to involve all of the modules to redesign how the features that are already in Blender will be presented to the user. Basically removing the edit mode and the sculpting mode and making new modes that will be more general. Freeform mode will be a mode similar to Sculpting, but involving other features to form objects freely. Maybe scattering hairs and stuff like that could be all in Freeform mode together with sculpting.

Computer Aided Design (CAD) mode will take advantage of Blender's non-destructive modelling. Using modifiers and primitives to build complex object in a way where anything will be editable in the future. Perhaps UI controls for this mode should be designed. But it's only a matter of designing the interaction. The gizmos. Not the features them selves. Also there are plugins already to edit objects this way. Blender developers only want to make it an internal Blender mode.

Edit mode will stay. Where you can manually tweak shapes. But it will be extended with tweaking other data as well. Turning familiar Blender into something fresh. Modelling module can still focus on modelling. But it's going to be used differently.

Another way Blender is focused is by focusing on few things at a time. At least as a core team. For example for 2021 the plans are to focus on Asset Browser, the new asset system for linking and appending objects from other files. Library Overrides, the new linking, overrides feature that will replace the old proxy. Enabling you to edit single settings in linked object. Preserving the rest of the object linked. This will make working with assets so much easier. Also they are focusing on Geometry Nodes. The new system for building shapes using the Node Editor. Making Blender one step more procedural. Then they want to start moving to Vulkan from OpenGL, further develop Greece Pencil features and finally merge the CyclesX fork, that a few developers worked on for half a year already, trying to improve speed and reliability of Cycles. And a few other things too.

The idea here is, that they make projects like this, rather then focusing on the whole thing all the time. Otherwise it will not be productive. Also they are in the middle of their new open movie "Sprite Fright". They did a few of them already. Every time working closely with the artist to see what improvements they need. And every time it made Blender a lot better. Maybe they will not focus on the improvements *you* need. For this you have to start talking. There is Blender.Chat and Blender.Community. Websites where people can talk to developers and other Blender-heads about things they like and hate in Blender.

For hardcore people who do coding, there is Devtalk and Official Repository site. And for those who want to know what they are planning and doing, there is Developers Blog, outlining their current plans and achievements in an easy to digest manner.

For people on Odysee, every week there is an hour long video from the Official Blender Channel where a handsome dude from Argentina is talking about all the news in Blender for the last week.

Happy Hacking!

Adult vs Mature

Usually the age of a person is not nearly the way of knowing if this person is mature. Quite the opposite could be true. And it’s more often then you think it is.

lbry://@blenderdumbass:f/Adult-vs-Mature:8

By that point I actually don't know what channel I will post this article to. Since it's both philosophically interesting to talk about. As a kind of thing I do on Blender Dumbass. But in the same time I think I may post it on my other channel, where discussions like this are brought more often. You already know what channel it is on. I don't. Which is interesting to say the least.

A lot of people reach their 18 and do dumb shit. Since they are not yet mature. But the idea of being an adult is being mature. Or is it?

Why are there age restrictions?

To be considered an adult in most cultures, you have to get to 18 years of age. Which gives you the full set of privileges any other adult gets. Before that age you may not have sex, drive a car, drink and alcoholic drink or smoke cigarettes. In some places it will restrict your ability to have speech, own property and maybe a few other things.

Of course different cultures has different laws. And I'm specifically avoiding the word "countries". Since for example an "adult" for Israeli law is a separate concept from an "adult" in Jewish tradition. In Israel you can own property and have speech (age of consent) by 16. But alcohol and cigarettes are prohibited until 18. Which is when you considered a full adult. For Jewish tradition most of the adulthood happens by 13 for boys (Barmitzva) and 12 for girls (Batmitzva). But you have to listen to advice of your parents until 20.

If there was a country where Jewish tradition is the law, people would marry by 13 and have children by 14. This would include a ritual of Kidush every Saturday, which involves drinking a glass full of wine. And owning property. But Israel has it's own law prohibiting those until a higher age is reached.

This doesn't mean that everything is done according to the law. A lot of boys as young as 13 do Kidush. Some may substitute wine for a grape juice. If it's 100% natural, it's considered okay for the ritual. But most of them will still use real wine. Also in many cases, those boys are invited to a gathering called "Farbrengen" in which they participating in a different ritual called "Lehaim" (For live). Which involves drinking a shot of vodka, whisky or similar substance.

I'm speaking to you about this since I lived thought it myself. When I was 13 I was considered an "almost adult" by most people around me. I was excepted to adult conversations. I was participating in Lehaims and Kidushes. I drunk alcohol. My views were taken seriously. I became an adult. But only from the perspective of the ultra-orthodox Jewish people. But, from the perspective of the casual folk, since I wasn't 18 yet, I was just a child.

But why there are age restrictions to being with? What makes them so necessary, even when people can't even agree on what exact age it is supposed to be? The reason is quite obvious. Children are usually not mature, not prepared to live, the same way adults are. And thus some time to prepare, some time to mature, should be given to them. And it's usually about 18 years.

Age restrictions has a fundamental flaw

In order for people to be prepared for something, they need experience. And this experience is gathered by trying something. A good artist doesn't become good overnight. He tries to draw and fails. He tries one more time and becomes a bit better.

And with each try he tries, he becomes better and better.

With Maturity it's similar. In order to get prepared for live, living is necessary. In order to learn how to fix problems that come with living, get a job, pay taxes, raise a child, you have to try those things. And at first you most likely going to fail.

I believe that 18 years could be enough for most people to experience live enough, to learn it so they would be prepared for it. But in reality the laws are set in such a way that you can try things only after you are 18. You can start learning, start experiencing, start maturing at most things, only after you reach a legal limit, after which it is legal.

In other words, instead of learning, most kids are simply waiting to start learning. And when the opportunity arrives to learn, it expected from them to know everything already.

Alcohol

I don't drink alcohol now. And I'm very good at resisting alcohol. I have drunk last time about 4 years ago, when a very wealthy person was persuading me to simply taste a little bit of a liqueur he liked. I dipped my lips in it to get the taste. It was okay. I didn't get drunk. I didn't even feel much of an effect.

I could resist alcohol better than most people there. Even those that are older than me. But this wasn't always the case like that. When I was still a child I had a different story.

It was the last day of Passover. And the ritual was to drink 4 full cups of wine. Full to such an extend that the wine spills out. I went to one synagogue. I drunk 6 cups of wine, since I could sneak another 2, while the drunk adults around me are not noticing. They gave me the first 4 since I was over 13. Then I went to another synagogue, much bigger one, where there was much more opportunity to drink. I remember a guy from Mexico with his special, super strong wine. I remember sitting at multiple tables and drinking. I remember trying to go home.

I woke up, opened my eyes and saw multiple doctors above me. I was in a hospital. Non of them knew yet what was my name or where I lived. Or who to call about me. They just saw a teen nearly dead.

Poisoned by alcohol. They made me puke all the wine out. I still woke up drunk though. I was getting off the alcohol left in my blood for the next 6 hours.

When my Mother came and took me from there, we entered a store near our home, to buy some snacks. This was the kind of small store and the front part of it was full of alcohol drinks. I remember passing by that store, or entering it and imagining drinking all this stuff. Being this dude from the movies. Feeling euphoria. Feeling cool. But this time it was different. From a single glance on one of those bottles I wanted to puke. This was the worst kind of offering there is. I didn't want to drink any of that anymore. I had a bad experience with it.

If childhood is to prepare for live, this situation was my childhood in relation to alcohol. This was an experience from which I learned. An experience that matured me. And it could happen before or after I was 18. This one happened before. Making the 18 thing work. After I was 18 I was not drinking anymore. I was prepared. But most people start drinking at 18. While I started at 13.

Unfortunately not all people can develop a strength like that. I used to take small shots of vodka at Saturdays at 13. Not enough to register. I wasn't drunk from them. I could feel drunk later doing Kidush when I was 14. Drinking Friday nights a full glass of wine. I would drink much more at 15 and 16. And I nearly died at 17. I had a gradual enough alcoholic experience. Similar to how people at a gym lift more and more weight. I lifted more and more alcohol. That developed into a strong mental resistance.

Most people get very drunk early on at age 18. Usually at their birthday party. At the morning they have a hangover. A huge headache. And while some learn to live through it. Other realize, that by drinking more, they can remove the headache. And often it continues through the day until the next night. And so on, days, weeks, months. This is called alcoholism.

My Mother had this problem. She would drink at a holiday, for example. And will not stop for months. She was thrown into drinking immediately. And in large quantities. Making her an alcoholic. Not letting her defense to develop against it. She died this year from an alcohol poisoning. She was mature in so many other areas. But perhaps I had given a chance to outgrow her in regards to alcohol.

Conclusion

Being adult, as in 18+ has nothing to do with being actually capable. If the law was successful to restrict all alcohol for kids. People would die more from it. If the law was successful at restricting porn and sex from children, people would be traumatized from getting into it when adults. Everything in live should be gradual and 18 years is enough for it.

Think about telling a person who want to grow muscles to wait for a random amount of years before he can lift. And then call him a buff. In one millisecond changing his status from weak and thin, to a strong bodybuilder. And tell him that he need muscles in order to lift in the first place. This is why he needs to wait the random amount of years.

People need experience to grow. The more gradual it is, the better. If kids don't experience live until they are 18. The whole point of 18 breaks. This is why we see some countries move the drinking age to 21.

They misunderstand the point. What you have to do instead is, for example, allow 13 year olds to drink beer, 15 year olds to drink wine and 18 year olds to drink vodka. This would work better. Since this would give them actual, gradual experience to learn and grow.

Happy Hacking!

Locks vs Shackles

Prisons are usually very safe. But would you live your life in a prison?

lbry://@blenderdumbass:f/Locks-Vs-Shackles:a

On this channel, I'm talking about computers. So if you are really looking for a review of what's better, Lock or Shackles, you are in the wrong place. But I would advice you to still give it a read.

I'm not a very security focused person. I like tools to make artsy things with them, rather than tools to keep all my secrets safe. But for both tasks I would chose free software. Software that respects users freedom.

Prisons are kind of secure

If you want to be safe from most people out there, making a small crime to get to prison would be a way to go. Prison security was designed to not let you out, but it's unbreakable from both sides. So sitting in a prison cell will make you more capable to avoid things outside of prison. Good and bad things.

But from with in the prison, from guards and cellmates, you will not be safe. They are on the same side of the wall as you are. And if some of them has a problem with you, you have no place to run anymore. Since, well, you are in prison.

When I talk to people about Free Software I make a prison metaphor all the time. For example. Imagine you are given a chance to have the best devices, the fastest internet connection, the tastiest food and so on, while you have to stay in the same prison cell for the rest of your life. You will have the strictest schedule of when you are given these things. And you will have strict rules of what you are allowed to do with these things.

From the other side imagine being a homeless person, with not even a single penny left. Hungry, but free. Most people, when I tell them this, want to be the homeless person. They know that they will have a chance to climb up, clean themselves and become members of society. There are stories of homeless people becoming millionaires. Therefor, while it may be hard, it's possible to get all those devices, internet and food without sacrificing freedom.

But when it comes to choosing software, people are not aware that proprietary software, they tent to use, is a prison, designed to be a prison.

When Apple argues that their strict policy of allowing installing software only from the App-store is "More secure" for the user. They are making the same argument as I did earlier. "Prisons are kind of secure". They want people to believe that by giving up their freedom, they get something. But people can get those same things, perhaps a bit harder, but another way. A way that keeps them Free.

Also, if you install software only from Apple. Only approved by Apple. Like in the case of prison guards. You loose all security from Apple themselves. It's is a very well known fact by this point. With proprietary software, there is no security against it's proprietor.

What is the difference between a "virus" that only does malicious things and a program that does some malicious things, while giving also an ability to do something useful. The second example is a virus in disguise.

What is the difference between a cracker that only wants to do bad things to you and a big company that does bad things to you, while giving you some value in return. The second example is a cracker in disguise.

You have the keys from locks, not from shackles

When we use locks to keep something save, like for example, an unfinished project you don't want leaked yet, or a secret that may cause a scandal, or your bank account passwords. You want to lock it, so only you can open it later. When you lock your house, when going out, you do it with a key. A key that you hold yourself.

Imagine that you buy an item that is delivered in a secure, metal box with a lock on it. Separately you have been delivered a key for this box. Now you have the key to open your item. And then you are free to do with it what ever you want. This is fine. Some companies do that, like Purism, with they security USB thumbstick.

From the other side, imagine a box with, let's say, a book. That also has a lock on it. But you don't have the key. You have a device, that is also a box that is locked. When you want to read the book, you use that device. It calls the company that published the book. And receives conformation that you indeed bought it, with a one time key, but not for the lock, for the camera inside the book's box. It streams live feed from inside the box, outside to you. And observes you carefully. If you simply read the text of the book, it let's you do it. But if you take a picture, write something down or try to memorize the text of the book. It will shut the system down immediately.

Now, you may think, by that point, you will break the box and get the book out of there. I mean, you payed for it after all. But imagine that the publisher convinced the government to make a law, that makes it illegal to break this box. Unfortunately. This law exists in some countries. But because building metal boxes is harder than making software, they do this kind of abuse with software instead. Calling the system DRM. They interpret it as "Digital Rights Management" but really it's "Digital Restrictions Management".

They argue that they implement "locks". So only people who buy the copy can get the copy. But those are not locks. Those are shackles. The keys from those locks are not yours. They belong to somebody else. And you have a pointless, box that only teases you, with what it feels like, to own that copy. And you never owned that copy. You was merely licensed to read it.

Not so long ago I bought a book. Not a bad one, almost finished reading it. I bought a physical, printed book. That doesn't require internet connection. That doesn't need DRM systems to read it. That I own. That I can sell later to somebody. Or give away, if I so desire. I would not buy a book in a box with a license to read it. It's beyond abuse, from my point of view.

Conclusion

I was never a big security person. But I always wanted to be able to do things. And doing things requires freedom. I want to do what I want, when I want it. And for this I don't want things that put shackles on me. Locks on the other hand, well, as long as I have the keys, they are okay.

Happy Hacking!

My Thoughts on Windows 11

Short – Windows 11 is boring.

`lbry://@blenderdumbass:f/my-thoughts-on-windows-11:c`

Not so long ago I changed my keyboard and mouse, since the mouse started getting issues with clicking. I went to a store, and since I'm a writer and artist, and not a gamer, I asked for the cheapest pare of wired mouse and keyboard, with a keyboard being not silent when typing. I have to hear the clicks.

When they heard what I was looking for, they immediately showed me their catalogue of very cheap, loud keyboards. All the top pages were filled with cheap pares of wired keyboards and mice from Microsoft corporation. I told him "Give me the cheapest, that's not Microsoft or Sony". He looked at me weird, but later, he found a slightly more expensive pare of the HP bundle I'm using to type this article right now.

I hate Microsoft

On the page of Motives For Writing Free Software at GNU.ORG they mention one thing that might motivate a lot of people to write Free Software programs. **Hatred for Microsoft**. I think this one is about me. I avoid Microsoft. As well as Apple, Sony and Coca Cola because I feel like these companies do not deserve my money. About HP I have nothing against. Maybe I just didn't do my research and you may point me in the right direction. But they seem nice as of now. At least compared to Microsoft.

This is why I don't really care about the news of their Windows 11. This is something that has nothing to do with me.

I use KDE

I was hopping Desktops for the last year or so, liking Gnome a lot. But eventually setting (for now) on KDE. The screen shots and videos I've seen featuring Windows 11 look a lot like the KDE Desktop. Maybe Microsoft is trying to win over those people who use GNU / Linux for some stylistic choices.

They, as a huge corporation who wiretaps people and have researchers know, what arguments people may use to convince user to move away from Microsoft products to better products. People were moving to GNU / Linux for developing. They made WSL (Windows Sub-System For Linux) . People are moving for good looks and themes. Now they made this as well. People are moving for tiled window managers. Windows implemented it too.

Will I give Windows 11 a try?

No.

Will I recommend Windows 11?

No.

Conclusion

Stop talking about this worthless OS. It was dead before it was born.

Happy Hacking!

It's Not Only About Privacy!

There are other types of malware besides spying. Some of them are known, some are less known.

`lbry://@blenderdumbass:f/Its-not-only-about-privacy:c`

Malware is software that has malicious features. Free Software or Proprietary, it can be Malware. We know of Microsoft and Google software to be spying on us if we let ourselves run them. Unfortunately sometimes malware can be found in our beloved Free Software programs too. Like the infamous Amazon search on Ubuntu. But with Free Software it can be edited out.

In 2012 Edward Snowden made his revelations about constant surveillance that caused a huge outburst of privacy concerns. Some companies, like Apple, made it their marketing strategy. Trying to appeal to paranoid people, while simultaneously doing the same things, they were promising to prevent. Privacy became a huge topic. Privacy became the number one argument in promoting Free Software. But Free Software and privacy was out there way before Edward Snowden revelations. And surveillance wasn't the only thing. There are other nasty things found often in proprietary software, that people often don't talk about.

There is a page titled Proprietary Software Is Often Malware on GNU.ORG, listing various nasty things in software that could not be edited out. Since the software was proprietary. With Free Software, somebody would make a fixed version, without the nasty thing. And people could use this instead.

I'm not going to list all the examples, I gave you the link. But I want to get to your attention the categories of malware listed in the page. Because even though privacy is important, it's not only about privacy.

Addictions

https://www.gnu.org/proprietary/proprietary-addictions.html

A lot of proprietary software, especially proprietary games utilize a large amount of strategies to keep users using the software forever. Making the users very addicted to the software. Remember the YouTube algorithm. It's so addictive that people who wanted to get away from YouTube, ended up developing Invidious and FreeTube. Unofficial Free Software client applications to watch videos on the YouTube platform.

Game designers now a days are resembling psychological manipulators more than programmers.

If they want their game to compete with the current market, they have to "polish" the mechanics, animations and all the other things in the game just right, to give just the perfect amount of dopamine in the player, so he will never cancel the subscription.

And I'm not even talking about gambling. Game developers are making loot boxes and other strategies, successfully turning players into mindless zombies that spend thousands of dollars on pointless, randomized, virtual items. Some countries outright banned loot boxes, because of how nasty they are.

In Free Software it may be implemented. And some users might want to play the addictive games. But you can always fork it and remove the activeness. A lot of people prefer the LBRY Desktop app rather then Odysee.com. Since it's less addictive. With Free Software you can deal with addiction. With proprietary software, you stuck being manipulated.

Back Doors

https://www.gnu.org/proprietary/proprietary-back-doors.html

Proprietary software developers love their power. And sometimes they love their power a bit too much. Some programs contain features that let somebody else control your computer.

Technically a VNC remote control is a Back door. Since it's a program specifically designed to control one computer remotely from another computer. Sometimes it makes sense to use it. Like for example, you may need help from a friend. And it's faster if this friend will do the work himself, instead of explaining you everything. In this case, you may give him control. But you are in control of when you give him, and whether you give him, the control.

With the back door malware on the other hand, the proprietor is in the control, if the software is installed on your computer. If it's proprietary software, you can't remove the feature. Also it might not even be very noticeable that somebody is controlling your computer. They might not move the mouse and press the keys. But for example, edit files on the background. Without showing you that they are editing files. Reading them, deleting them. Installing software you didn't ask for. And so on.

Universal Back Door is when they can do anything at all with your computer. Usually it's achieved by having an automatic updater. Microsoft Windows has a Universal Back Door. It doesn't ask you when it's changes the software. It just does it. And the changes could have any other malware in it.

Censorship

https://www.gnu.org/proprietary/proprietary-censorship.html

Sometimes companies do not want things to be said. They want things to be only the way they envision them. They want political views, only those they hold. They want only their software to succeed.

Google blocked the LBRY app on the Google Play not so long ago. Finding any kind on excuse to do so. Since it's a direct competitor to their YouTube malware.

Deplatforming of people is another thing that happens quite often on non-free platforms. Canceling individuals because the proprietor doesn't share the views of the person.

With Free Software this cannot happen. The "Canceled" individual can always make his own server, his own fork. Under his control. LBRY and Federated Social Media proves that Free Speech can exist. Only one thing needs to be there to make it happen. Free Software.

Coverups

https://www.gnu.org/proprietary/proprietary-coverups.html

Go to any Free Software git page. There is probably an issue tracker. A bug tracker where you can read all the bugs that were reported. All the known issues of any Free Software are usually publicly available.

On the other hand, proprietary software developers keep their bugs secret. Some of the "bugs" are actually malicious features. But sometimes there is an actual bug. A mistake made by the developers. Such mistakes can lower the *perceived trustworthiness* of the company that makes the software. And thus lower the share value of the company. So proprietary software companies try to avoid publicly admitting any mistakes they made. Covering up most of the bugs.

A lot of terrible security bugs can be exploitable by crackers, if you are on a proprietary software operating system. While you don't even know that they exist. Since the company is burning the evidence down.

With Free Software it's not only easy to find what bugs are there. People with knowledge fix them way more often. So while it's not for certain that Free Software is more secure or less buggy, it tends to be so.

Deception

https://www.gnu.org/proprietary/proprietary-deception.html

Sometimes proprietary software developers give people Freedom that they later regret giving. For example they might make an operating system that is asking permission to install updates. And in such cases, if the user knows that the updates will be malicious, they will not install them.

The obvious choice for the proprietary software developer, is to lie about the update's intentions. And deceive the user into installing a downgrade.

Other times the interface of the program may show you things that are not correct. Which are not a result of a mistake or a bug. Sometimes the developers might not want you to know what exactly the software is doing. So they design interfaces that deceive users.

You press the Airplane mode button, it shows the airplane mode is on. But is it? What if it's still connected to the cellular network and tracks you, but only *shows* you that it doesn't? What if the interface deceives you?

With Free Software, people can independently read the source code to confirm that the program does what it's intended to do. And doesn't deceive users by showing them things that are not true.

Digital Restrictions Management

https://www.gnu.org/proprietary/proprietary-drm.html

Sometimes users obtain a copy of a work in a completely legal way. And they want to be able to use the work in the way that's permitted by the law. Copyright law is not as strict as people might want to believe. Just think about what you can do with a printed book you own.

Digital Restrictions are malicious features in software that prevent people from doing things with files they obtained legally. They may be prevented from choosing a preferred player to play a movie or listen a song. They could be prevented from copying a page in a book for a class. And other things that are completely legal under copyright law.

With Free Software you could edit these things out, until a law called DMCA struck the United States. And similar laws followed later in other places. These laws strictly prohibit breaking DRM systems.

Only DRM systems implemented under specific Free Software licenses like GNU GPLv3 have immunity against it. The license specifically removes all legal power of the developer of the software to file a DMCA request on the person who is editing the program.

But also, most Free Software just don't have DRM implemented anyway.

Fraud

https://www.gnu.org/proprietary/proprietary-fraud.html

Are you angry yet? Because this one will make you angry. Think about a deception tactic but that will make you spend more money than you want to spend. A tactic that can be built into software.

This is a widely used tactic. A lot of games and apps use this tactic to make people spend money. Usually targeting kids, making in-game purchases, not explaining what is it they are doing. Imagine a game specifically designed to make kids waste your money. But not even telling them, what it is they are doing.

Or imagine a simpler tactic. Every next level and every time you retry a level, you have to press in the specific spot. Training your muscle memory to simply press there, every time there is a game over. Then randomly the game put there a button to purchase something. Making you click that button without noticing it. Wasting your money, because you was conditioned to press a spot.

And I'm not even talking about simple stealing of money. Like saving your bank data to be used by malicious actors. Slowly or quickly drawing out all of the dollars out of your visa. Buying things with it, unrelated to you.

With Free Software all these things could be avoided. Since forks would be done, issues would be issued and clean, upgraded versions of the software would be released.

Incompatibility

https://www.gnu.org/proprietary/proprietary-incompatibility.html

Software that's made to process data is made to process data. But imagine software that's designed specifically not to let you process data. Not to allow you to migrate away from it. Software that implements artificial restrictions, that should not be there, that will keep you using only this software.

This is what's called Incompatibility or Vendor Lock-in. It's when a company specifically designs a format that will be read only by one program, forcing people to use only that one program to work with this format. This is also when a program doesn't want to enable you to edit files made with another program, while is capable to read them.

In Free Software there is a concept of Free Format. A format well documented and implemented in Free Software. Not under any software patent. Or under a license like GNU GPLv3 that gives away the patent rights. This is why OGG exists. It's a competitor to MP3 to avoid patents and Vendor Lock-ins. This is why OGV and WEBM exist. To avoid problems with MP4.

Libre-Office doesn't save to DOC by the default. Since it was developed by Microsoft to be incompatible with everything else. Hard work and tears went into reverse-engineering this format, to be able to open DOC documents in Free Software.

Insecurity

https://www.gnu.org/proprietary/proprietary-insecurity.html

Speaking about Privacy. Security is what makes privacy work. There is no reasonably sized program, free or proprietary that doesn't contain bugs. With each line of code, the chance of finding a bug grows exponentially. But with proprietary software there is no chance to be able to fix the bug yourself. You need to prey that a) the company will acknowledge the bug in the first place and b) they will fix it. With Free Software people can fix it immediately and have a patch the same day.

But it doesn't excuse some programs that have insecurities as a feature. Making sure there is no protection of Data, that user can implement to mitigate the abilities of the proprietor to read it and use it in malicious ways.

Some go so far as making the program a literal broadcasting station of your personal live. Just so it would not be difficult, to get the data about you. Sending to a server literally every keystroke and mouse movement you make. Recording all the conversations you have in the house. And more.

And if one company can access it, there is an access point. This access point can be reverse engineered. This access point is most likely used by more than just the proprietor. And using this kind of software will make your life available to who ever wants to risk getting in touch with a cracker on a dark web.

Interference

https://www.gnu.org/proprietary/proprietary-interference.html

But how about just being a nasty, dufus? Some programmers that made users already addicted and vendor locked, may do a bit of annoyance. Things that are not necessarily insecure by design. They could be. But things that are simply make you angry and annoyed.

This are YouTube's double ads and triple ads and mid-roll ads. YouTube demonetization for random reasons. And other things. This is when your phone doesn't let you remove an app. Or when it installs one automatically even though you've deleted it a millionth time already. It's when you have DRM in a battery. That makes it hard to repair your device. And other nasty stuff that makes my blood boil to hear. To hear that people put up with all that nonsense.

With Free Software, there would be a fix. A fork that people could use instead. But I doubt that any Free Software project will have such an anti-feature in the main branch for long.

Jails

https://www.gnu.org/proprietary/proprietary-jails.html

Jail is when you can only install software that the proprietor want you to be able to install. For example, in order to install software not via the iPhone's App-store you have to Jail-Break the phone. Acknowledging that this phone is a Jail.

A computer with install-able software is a computer where the user should be able to choose what exactly are the software installed. And if he wants to install something outside of the main repository, the user should be able to do it, in a more or less straight forward way.

It's called side-loading. Apple is trying to demonise this ability, claiming that it's "insecure". But what they are actually doing is fighting with the argument of Free Software. That people should be able to install what ever they want.

Some people claim that this is also happening to the FSF approved GNU / Linux distros. That they are "not allowing" installation of proprietary software. It's not true. You can still do it. It's just not going to be in the default repository. You will need to side-load it. The ability to do is still there.

Manipulation

https://www.gnu.org/proprietary/proprietary-manipulation.html

Think about clever software developers. They have a degree in psychology and they want users to do a thing. It could be anything. From paying a large sum of money, to buying pointless things, to believing an idea.

These developers will create software that deceive subtly enough to manipulate users psychologically to do things they others wouldn't do. This kind of software is what conspiracy theories are made from. And unfortunately a lot of them were caught doing this for real.

With proprietary software the manipulation will continue. Since the proprietor doesn't want to remove the malicious features. With Free Software, you are free to edit this feature out of the software.

Obsolescence

https://www.gnu.org/proprietary/proprietary-obsolescence.html

Some software is worse than others in some areas. Making your software obsolete can make for a worse software. Unfortunately, while Free Software, can always be continued, if a proprietor of a proprietary program doesn't see any penitential in it anymore, the program dies.

Sometimes the kill-switch is on a timer. Sometimes the developer knows exactly how long the software will be active in development. Or active at all.

This kind of obsolescence leads people to buy new electronic items instead of keeping using the old ones. Since the software of the old one is no longer wants to boot up. If it was Free Software, the bug could be fixed. And the development continued.

Sabotage

https://www.gnu.org/proprietary/proprietary-sabotage.html

There are many examples of Sabotage, but I want to illustrate you one. This will make your blood boil.

Imagine buying a safety gear. An airbag that should inflate when the danger comes. Like the one installed in cars and some motorcycles. And now imagine software inside of this "safety" gear that is tight to a subscription service. If you don't pay in time, you will die. If you get into an accident and you skipped payment, the software inside the airbag will not work. And the airbag will not inflate.

This is Sabotage. If you do something that the developer doesn't want. You face real life consequences. It could range from missing files to death. With Free Software it would be edited out immediately.

Subscriptions

https://www.gnu.org/proprietary/proprietary-subscriptions.html

I don't know. This ones seems self-explanatory. You have features implemented. But they refuse to work unless you pay for it every month. Just plain evil. And it's not like you are paying for an update. No. You are paying every month for the same exact peace of software. Or you are denied the ability to use it, if you don't pay.

Free Software definition's Freedom 0 is incompatible with it. To be free to run the software when ever, for what ever purpose. So all Free Software is automatically immune to this. And if somebody will try to implement it. We'll fork it.

Surveillance

https://www.gnu.org/proprietary/proprietary-surveillance.html

This is the privacy invading, all knowing corporation that targets you ads and sells your private data. Either received in a secure way or not.

Free Software can have this happen. But people are usually good at spotting it and deleting those anti-features.

Tethers

https://www.gnu.org/proprietary/proprietary-tethers.html

This is when a program or an entire device needs a constant communication with a server in order to work. Sometimes, like for example with communication software it's expected. But the problem here is that the server is unchangeable. And if the proprietor decided to shut it off, the program is no longer working.

Free Software communication services like Jitsi usually have their server code available. People can create their own servers as they wish. So if something happens to one of them, the software doesn't become obsolete.

For other Free Software that actually do that, like Telegram. You can always migrate it slowly, by implementing a similar protocol with a different server. Making the main branch die, but keeping the fork active.

Tyrants

https://www.gnu.org/proprietary/proprietary-tyrants.html

Tyrants are not software, they are devices. Devices that refuse you to install different software on them. On most computers you can wipe Windows and upgrade to GNU / Linux. Tyrants will only boot up when the original program is there. Wiping the operating, or installing anything else will stop it from running at all.

GNU GPLv3 was designed to fight against Tyrants too. But unfortunately Linus Torvalds didn't want to move Linux Kernel to GPLv3. And thus many Android phones (using Linux kernel) are tyrants. The hardware of which we cannot use to make a Free Phone.

Conclusion

Privacy is important. But it's only a part of the picture. Freedom of software fixes so much more. With Freedom, anything nasty can be edited out.

Happy Hacking!

Making a Movie is Hard!

A showcase of complexity in making a movie project. (The problem outlined in this article is already solved.)

lbry://@blenderdumbass:f/making-a-movie-is-hard:c

Last year or so I had bad thing after bad thing happening non stop. I already wrote and article about it. I can add to the list of sufferings, the recent cut I had on my thumb. That resulted in a very repulsing scar. Before I moved to LBRY (known as Odysee) I had a YouTube channel. And the last dozen videos or so on that channel (archived here) I was actively developing a movie project.

You may know me by my another name J.Y.Amihud. A director of infamous, to some people I'm Not Even Human. A 32 minute long, short film, made entirely in Blender. This movie was 3 long years in making. From 2015 to it's release in 2018. And I learned a lot during the production of that movie. I document it more in this article.

On Blender.Chat, some teenage Blenderers who resonate with the film, find courage to speak to me. Many users there, simply write me to the personal chat at random. And when I ask if they saw my film, trying to get them to watch it. They already know everything about it. And they want advice in their own films. When I talk to them more. More often than not, they end up being 13 to 16 year old teenagers.

But I can't give them advice since for the last year I didn't work on any film myself. I was working ones. But it turns out to be, that last things I did, was already a year ago. I feel like I'm not competent enough to give them any advice. Even though I finished a project before.

Inception of Moria's Race

In the beginning of 2020 when the virus still didn't hit the world much, I received a phone call from an old friend of mine. She was talking about an idea that she had. She wanted to make very cheap cartoon, similar to ones her daughter, Moria, is watching on YouTube.

They are all low quality CGI animations with very basic plots. They probably produce an episode every week. And they are on such a tight schedule, that the show looks like crap. She wanted my help, knowing that I can do that, with making something similar.

I explained her that I don't want to make a low quality show. I actually had a movie project back then. I was going to be a cheap, but interesting feature film. With a lot of work put into making a cheap to produce, interesting to follow, tense plot. Reading it still give me goosebumps. I still want to produce it one day.

By the time she called, I already didn't feel like much can happen with that movie. I already called a couple of studios in Israel. Others didn't answer the calls. And all of them either ignored me, or declined the offer. One studio would give me a chance if somebody else would direct it. But I wanted to directed so badly that there was no deal.

The script is about people talking, wearing police uniforms. By the time she called, I was already thinking about making the movie somehow myself. Trying to figure out how to make it. How to plan everything. But she called and sparked an interest.

I thought that maybe I can make a cartoon again. This time a bit better than I'm Not Even Human and maybe if I document it's development more, I can make a relative success with it. With a small following, developing another movie would be simpler. And making a cartoon is easier than trying to figure out a feature film with no budget.

Story, Concepts, Mood

She wanted to make children's entertainment. I wanted to make a film with a message. Even if it's a short film. I wanted it to be energetic, pretty and thought-provoking. Not the animation things, that those studios produce for YouTube every week.

Just before she called me I saw a movie that was surprisingly good. Best Picture nominated film Ford vs Ferrari. A movie about racing and the art of racing. A movie that made racing look so cool, I wanted to do racing next.

From another side, I'm Not Even Human is all about Respect Children. A fight against discrimination by age. If you read about my girlfriend, you probably know that she is still "underage". And while we are waiting and not doing anything prohibited by law, the amount of social pressure on us is still immense. And it's not like she is a stupid child. She is smarter than me. The problem here is for sure, the ageism.

So I wanted to make a racing movie with an anti-ageism message. This is a perfect recipe for tension and excitement. Since this insures kids driving cars. In the previous film, I also had kids driving cars. But the movie was trying to erase the 18+ law for the main child character completely. And some people saw it as too provocative. My father still doesn't give to see this film to my little brothers. This time, if the whole concept of the film is about kids racing. It solves the provocation problem. Giving me a possibility to make a "family friendly" respect children movie.

To give credit to the inspiration, the friend that called me. I wanted to credit her daughter, Moria. Making Moria the main character of the movie. A cool little girl who can really drive a car. And then I invented another character, her brother, Dani. Which by coincidence turned out to be a real name of the real life Moria's brother. The other child of that friend that called me. Moria was her first child. That she had at a very young age. Making this movie a kind of acknowledgement of this too. A double respect children.

Assets

I have an asset manager software for animated project that I developed for I'm Not Even Human. It's Free Software and could be downloaded here. So I was already pretty prepared when it comes to designing assets.

In about half a year since the initial call, I had all the assets finished. All of which is documented on my archived channel. This was roughly a year ago.

So what happened?

In the last year, I haven't done much with this movie. I did pose a few poses and made a few renders with the assets. But didn't make anything else. Not a single animation is finished. Not a single shot of the movie is done. I did try making a comics version of the film. I made a few panels using assets from the film. But that's about it.

Why? What happened? What suddenly stopped me? At the pace of the asset creation, if I didn't stop, I would've had a finished movie (according to the analytics in VCStudio) at roughly January of 2021. I wanted to release it in May. And it's already June and no scenes are finished. What happened?

I hit an issue I never was able to solve. This same issue occurred to me on I'm Not Even Human. But back then I figured out a creative workaround for it. This issue was casting voice actors. In I'm Not Even Human, originally I had a lot of dialogue for the main child character. And I was looking for a kid to make this movie with. But I wanted to make a film in Israel in English. And I needed an Israeli kid that sounds good in English. And all of it cost free since I have a budget of zero.

I ended up removing all the dialogue and making the main child character mute. Both because I couldn't figure out the actor and in the same time as a clever allegory. A child that has no voice. Furthering the anti-ageism themes.

This time, however this couldn't work. Finding a couple of kids to voice Moria and Dani was an easy task compared to casting real actors for a feature film project that I wanted to make prior. I thought it was trivial, so I made the story rely on their dialogue very heavily. And carried on with the assets not thinking about it too much.

This caused me a lot of problems. I think I could've hired someone, had I had any actual budget. Maybe even from outside Israel. But the budget is zero. I have no money. So this is a bit of a bigger problem.

Questions of where to look, how to look, who to look for, started arriving. Should it be a child, or a nice sounding woman is enough? Should they have a pure English or American accent, or they can sound Israeli? What should I do?

This together with a lot of headache from other sources caused me to doubt that I will even make the film ever. My brother is constantly asking me for animation shots. But how can I animate without the timing of the recording?

Plan

In any problem, the fist step of solving it, is to get a plan. A plan that might be changed if things go wrong. But an initial plan that will start the gears moving. If you don't have a plan. The plan could have only one step. And it's to figure out the rest of the plan. So I sat down with the headache and thought to figure out the plan. I have the first draft.

Remember I said that my girlfriend is underage? She could voice Moria. Her voice is quite good. I just made a reading with her. In Russian still. But she understands Moria's character a lot. So I think she is a good choice. Also it could be very cool to use her, as another kind of anti-ageism thing.

I know a decent English pronunciation. I can work with her to fix her pronunciation in English. And then having the Russian version of the script for the reference, I can get out of her the performance in English. I'm thinking about this like the Na'vi language in Avatar. The actors took a long course of perfecting the language and the accent of that language. So they could sound native Na'vi. I think similar course could be made to my girlfriend. And she will sound more or less native in English. After all, her name is Moria in the film. A bit of accent is acceptable.

For the recording, I will record with her both Moria and Dani. To be on the safe side. I will ask her to change the voice a bit for Dani. Maybe even a tiny bit of computer effect (pitch modulation) could be done to make Dani sound a bit younger. Then I will try figuring out recording my brother Pinchas (who is 7 years old) for the part of Dani.

If Pinchas will work, I will have two separate actors talking. If not, at least I will have my girlfriend's version of both. She wants to, later, help making the Russian version of the film. Which will not require learning an accent.

As soon as I will have the recordings, I will back them up and start doing the soundtrack of the first scene. I will need to figure out (after I know how Moria and Dani sound) what are their leitmotifs. So I could build the soundtrack accordingly. I already done that on I'm Not Even Human. And it's not too big of a deal.

I will assemble the sounds into a sound-only edit for each scene first. With talking and music flaw established first. Then I will find my cuts. And design the shots around that. Similarly to I'm Not Even Human.

And then slowly, but surely, I will have shot by shot done. Which after a scene or two will give me an estimate of how long it will take to finish the film. Few scenes in, I will release the first trailer. And toward the end I will release the second trailer, with more footage.

I'm planning to give up my freedom, to release the trailers on YouTube and Vimeo and other platforms. But with a disclaimer stating that I don't like these platforms. And I release it there only because I wish people would know about the movie's existence. The movie is planned to be only available on LBRY (Odysee) but with a Free license. Which will enable people to copy it around to other place, make reviews and do other things.

Conclusion

I had a terrible year of confusion. And it's expected. Similar things will happen to me again, when I will make my next films. But I think, I finally see the light in the end of the tunnel. And the progress will follow.

I'm planning to keep documenting this journey for people interested. Probably the next installment about it will not be very soon. Since I need to make a lot of teaching of my girlfriend. And she is very shy. So I think you may need to wait until I've got the recordings first.

Meanwhile I will write articles. And I may touch upon various themes. I already have a few in mind. So I will have something to write about tomorrow. It's an interesting feeling to finally have inspiration for that film again.

Happy Hacking!

Modern Film Look (Wetness)

It seems like dropping water on the movie set, will the film look more modern.

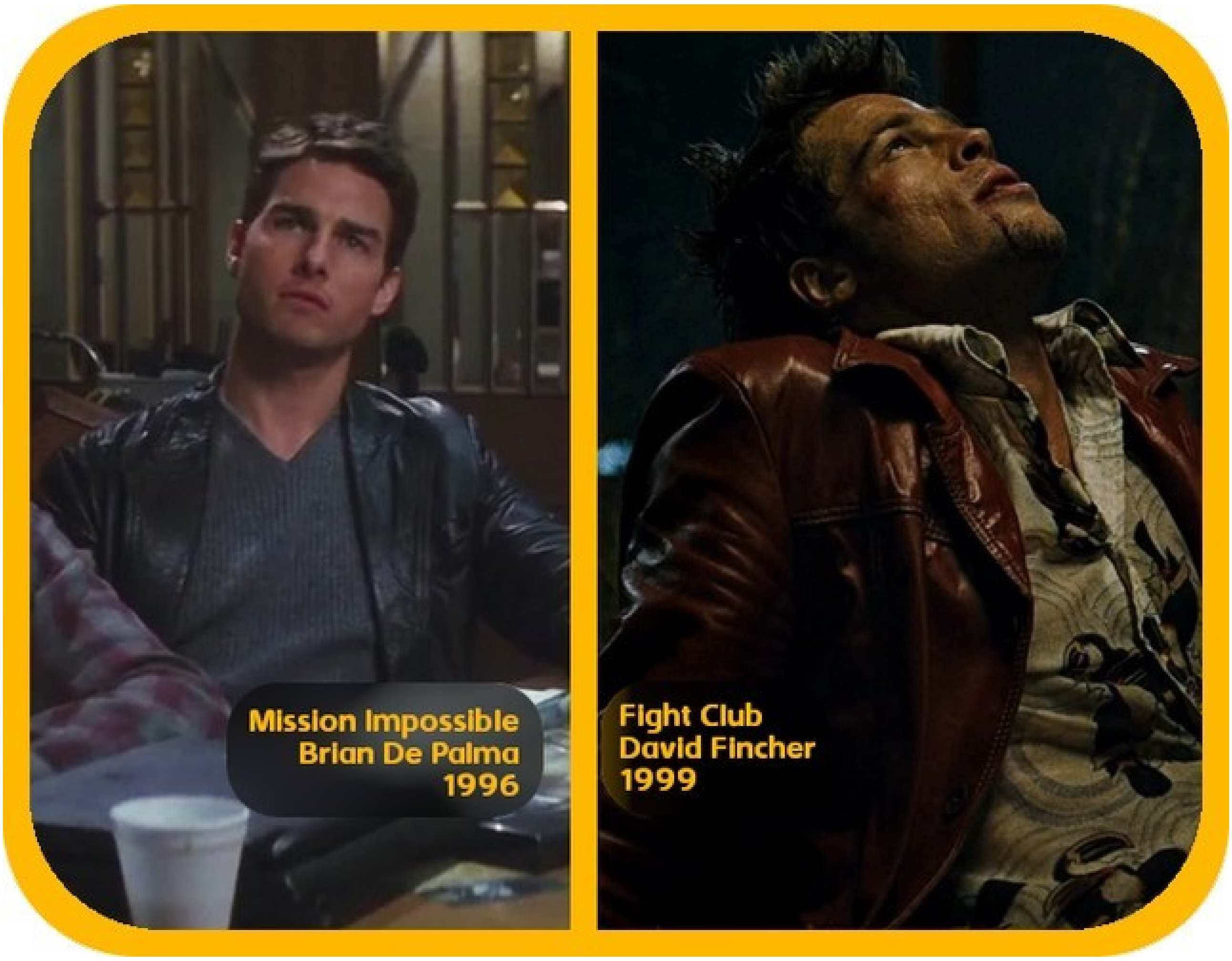
`lbry://@blenderdumbass:f/modern-film-look-wetness:b`

I was arguing with one guy, ones, about cinema. He was trying to prove to me that all people care about is the cinematography of the movie. 'The "modern look" is what's going to sell more tickets.' That what he used to say.

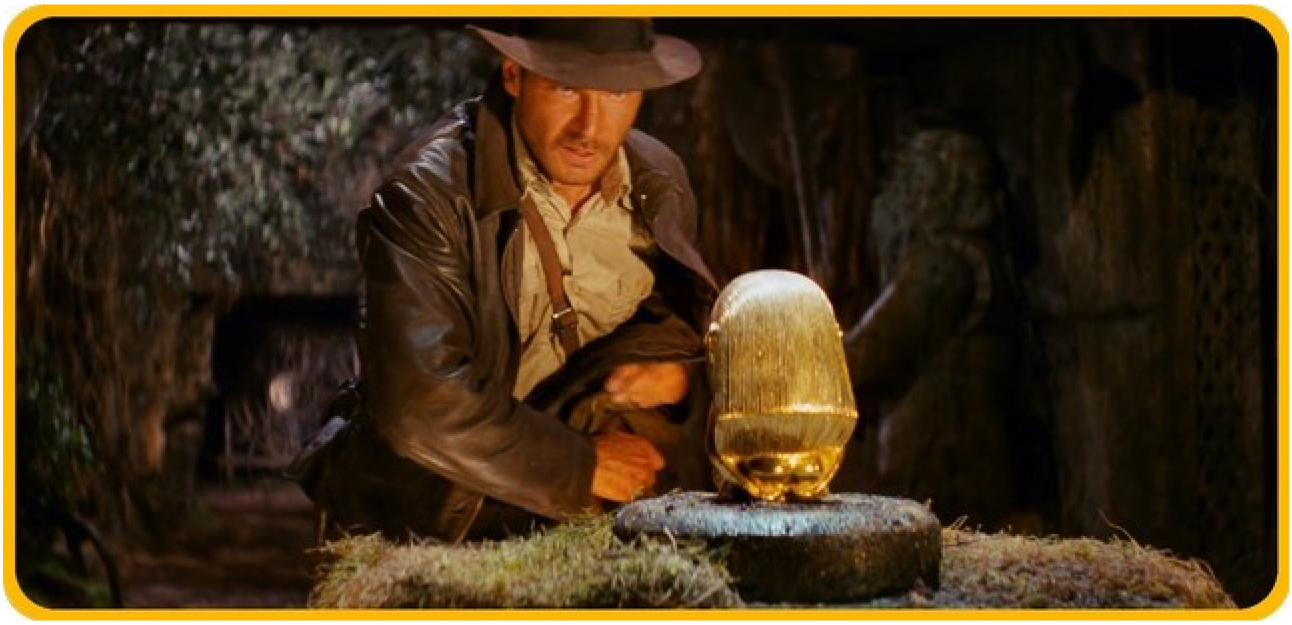
But then, when you actually think about it, yes you can distinguish between the two. Older films look older most of the time. But there are exceptions. Blade Runner for example doesn't look like it's shot in the early 80s. It looks more from a post-matrix, post-blade era. With dark shots, people shot with cool lighting techniques. And so on.

Dry vs Wet

I think I know the difference. And it has nothing to do with when the movie is shot. Let's look at two films shot roughly in the same time (late 90s). One, Mission Impossible by Brian De Palma that looks old. And the other one, Fight Club, still looks new. The difference in the timing is minuscule. The difference in style if huge. Bryan De Palma film, in my opinion looks too dry.



There is a technique that's used very often for films. And an exemplary film to look for this technique is fight club. It's an addition of wet, watery surfaces. Reflective things, puddles, polished jackets, sweat. Things that shine and make the image look wetter. This seems to have been not used quite a lot by Brian De Palma. And that's why his films look so dry. But I think it's more than that.



When I was a child I always couldn't believe that Steven Spielberg movies from the 80s were shot during the 80s. Since they looked all very modern. I remember watching ET and Raiders of the Lost Ark and wonder, how did he managed to be so ahead of his time. You may argue that he inspired the newer look. But what exactly is he doing here that so "modern" compared to other films in that era?

Raiders is 80% in a desert and the other 20% in doors. Being a dry film is a must, for such a movie. And Steven Spielberg doesn't try to wet the desert sets to make them wetter. But he uses some wetness. First is the infamous Jacket. That's quite reflective. Then he uses a lot of sweat. All the actors in all the scenes are constantly sweaty. Making their skin shine through out. But then he uses more things.

You can see in this shot, the bokeh (the de-focus circles) are not perfectly spherical. They are ellipses, which gives the image a kind of cool aesthetic. This is because during filming they used an anamorphic lens. A lens that distorts the image, so the wide screen aspect ratio image could be captured on a standard 16:9, 35 millimeter film.

This technique stayed popular to this day. A lot of modern movies still use anamorphic lenses. From Transformers to Super 8. From J.J.Abrams to Christopher Nolan. If they want to film in a wider aspect ratio, they most often choose an anamorphic lens.

Some aspects of the anamorphic look were criticised. For example until 2015 J.J.Abrams would overuse the lens flares. They are a gimmick for sure, but some of it probably contributes to the modern look of a film.

Modern cinematographers are more experimental in what they do with light physics. Older films were shot in such a way that any lens flare would be avoided at all cost. Lens flares were artifacts that were unprofessional and unacceptable in film. Today, on the other hand, they are more frequent and even expected to some degree.

In the late 70s Steven Spielberg drove in a city during the night, thinking about the climax of a film he was making, Close Encounters Of the Third Kind. While driving the night city, he found him self looking at all the neon lights. All this shine. And all the effects that it makes on the glass of the car. He saw the lens-flares.

Glass is a similar substance to water. They are both transparent. They both break light. And they both look wet. Glass has a tendency to look wet even while being completely dry. Keeping the glass in the picture of the film, keeping the lens-flares, so the audience will feel the glass of the lens, may make the movie wetter, and with it, more modern.

Steven ended up using the lens-flares to sell the effects of the alien ships in the film. Giving lens-flares to all their lights. And later used that same techniques in his other movies.

Anamorphic lens works that well, since people can feel, that a lens was used. Bokeh that draws attention to it, draws attention to the lens, to the glass, making the picture wetter. It could be ellipse bokeh, lens-flares, shallow dept of field. Anything that will draw attention to the lens, even perhaps on a subconscious level, will make the movie more modern.

Color and Contrast

If you come back to the first image on the top, of the Mission Impossible vs Fight Club. You can see that Mission Impossible has no contrast. The darker colors are greyish. No black. While Fight Club has a way deeper black. And higher contrast between the darker and brighter parts of the image.

In another term, this is a difference in dynamic range. How much dark can be the darkest and how much bright can be the brightest. How much light difference is there between the black and the white. If you ever seen an HDR (high dynamic range) monitor, you know what I'm talking about.

The more modern the film, the more colors it has. The more vibrant everything looks. But there is a problem. To make things look real you need to remove the colors. Usually directors specifically choose to shoot nearly black and white to give a movie this aesthetic of realism. Making a statement that the movie is about truth.

Sometimes though, like with Sin City, Black and White can have an opposite effect. It can make the movie feel more like it's printed in the comic-book and less like it's shot on an actual film.

So I think the question here, is between beauty of the image and realism of the image. And realism tends to look a bit older.

Every prop in modern movies with at least a bit of budget will be designed to accommodate a specific color scheme. Sometimes they may choose two or even one color for the whole film. And make the movie look prettier as a result. But by taking away some of it's realism.

For example, this is a frame from Moria's Race. And you can see that it has blue and orange color-scheme. All assets, cars, characters and sets are made with a strict use of the color-scheme. Some maybe are using shades of grey. And some can deviate from the orange or the blue a little bit, but still in the ballpark of the colors.

Conclusion?

I don't have a conclusion. I think I found some things that make movies look modern. But some movies may look "modern" without being defined as modern with my analysis. Many things could be added to the list of why certain movies look modern. Motion of camera, composition techniques, frequency of cutting, slow-motion, directorial style.

There are so many variables that giving a concrete answer is very hard. But certainly, making things a bit wetter will make it look cooler.

Happy Hacking!

Ads On Odysee?

Odysee recently implemented pre-roll ads for not signed users. This article was published just before that, speculating on various ideas for an ad system that could be implemented.

lbry://@blenderdumbass:f/Ads-On-Odysee:8

Odysee and Ads seems like two things that should never join together. But unfortunately it might happen. And even though I'm against ads, I think with a smart implementation, they are could be good.

Odysee is not YouTube. YouTube is a Cthulhu type monster, making people addicted and then abuses them for profit. Abuses them using ads. With Odysee I think an ad system could be done in a better way. In a respectful way. The way in which ads could become cool again.

Why are ads needed?

Monetization is a word people use a lot. On YouTube you upload a video and the YouTube system selects an ad to put in a pre-roll to your video. People will first watch the ad, then get to your video.

If you been in cinema, it's quite similar to cinema. They have ads and trailers right before the movie starts. I actually don't mind that. Some authors of videos and filmmakers even do product placements and sponsor segments. I don't think it's necessarily evil.

The evil starts when Google (the company that owns YouTube) wanted to be clever about the ads. If people click, the advertiser pays. So they need to make sure people click. And for this an algorithm that learns everything about you, from how you use your computer, phone, what you watch and who you communicate with, was developed. This algorithm helps categorize the ads. Personalizing them. Making them 100 times more effective. But as a side-effect, people using these platforms give away all of their privacy to a company that collects all of their data.

But Monetization on Odysee has nothing to do with ads? Right? Well, for now the company LBRY Inc. is giving people LBC coins if their publications are popular. It's like asking YouTube to give money for views. Without them getting anything in return.

LBRY Inc. has a lot of LBC and they may continue this generosity for some time, but then, ultimately, something will needed to be implemented instead. The current model is very unsustainable. They will run out of the LBC sooner or later. And they need to provide a different way for authors to monetize their publications.

There is the Support Button, hyper-chat and other thing like that. But how often do you give LBC to your favorite authors? I received some, but not nearly as I received from the automatic thing.

To give the author an ability to put a pre-roll ad can give the author a way to monetize the publication, without asking direct payment for it.

Ads are Opt-In

I heard a lot about ads being opt in. As in, you will need to go into the setting and enable them. And if you don't, you will stay on the Odysee without ads.

I don't know for sure, if this is what they are planning. But I know for sure that Odysee is Free Software and maximum you can edit the ads out. At least in the LBRY Desktop app. So I think they are going to address this obvious issue, by making the ads opt in.

The obvious question here is... **Who in his own mind will turn on ads?** And to this I will answer that a lot people will. I heard about a Patreon alternative based in France that is taking a different approach to author support. You can either tip the author with money. Or watch ads. And for each ad, the company will tip something to the author. I think unfortunately the service is non-free, so I would not recommend it. But the idea is interesting.

The authors may ask you to turn on the ads. Or perhaps the ads only on their channel, to give them support without giving direct LBC. If you want to contribute to the message, or help a project, you could always just watch some ads to tip.

Maybe even you could unlock payed publication by watching enough ads to unlock them.

Authors might be able to control ads better

How many times on YouTube you may watch an ad that contradicts the message of the author? Or an ad that the author didn't approve, or didn't want to approve? This is because ads on YouTube are based on the personality of the viewer.

Before that privacy nightmare was invented, the ads were based on contents of the video. Which was a bit better. But think about an ad that will be in total control of the author.

On YouTube they do sponsor segments. Think about if the sponsor segment is the skip-able ad on the beginning. It was edited by the author of the original video. It's payed with the rate the author set with the advertiser. And there was no middle man. Odysee wasn't there to make the deal. It just provided the software.

For example. A theoretical system could look something like this. An advertiser sets a channel on LBRY (Odysee). They either upload a publication, a video that they want people to see. Or they present a script for other authors to produce themselves. They add a maximum budget of LBC. They add a rate per click. And this will be added in a searchable directory of currently happening ads.

The author now wants to publish a video. And he looks into the directory of ads and finds the one he likes. With the good budget and click rate. Something that fits him. He can review the ad or the message. If it's a finished ad, he can post it. And it will be the pre-roll ad. If it's a script, a communication with the advertiser will start. To make them agree on the final version.

This could become a fair game advertising platform. Fair game, since the only middle man, is the directory that LBRY protocol may hold for currently happening ads. All of this middle man thing should be on the block chain. Meaning decentralized. Making a truly remarkable ads mechanism.

Conclusion

Ads don't have a good reputation today. But I think with a proper developer, an ad system can actually be kind of cool. I think we shouldn't just yell and scream because a word is said. But look at the details. Don't judge the book by it's cover.

Happy Hacking!

Odysee - Beyond Trans-Coding

More interesting ideas of the LBRY team to try an implement.

`lbry://@blenderdumbass:f/Odysee-Beyond-Trans-coding:9`

Odysee together with the LBRY protocol is the way of the future. And while it's in it's infancy. It can be very cool. One of the things that they are trying to make, that works half of the time, is trans-coding. Making a video available in multiple resolutions. So people could watch it smoothly on a connection that is not so great.

I think that this is a right step forward, but I also think that this system could be so much larger, than simply trans-coding. The concept of having multiple files representing the same publication could be extended much beyond that.

What is Trans-coding?

To do trans-coding, you are not making a server load up the full resolution version each time and downsize it for every person that has a slow connection. Those files are prepared ahead of the time. The 1080p, 720p 480p, 360p, 240p and so on, versions of the video are made and stored. And when a person requests a given version, he receives only one of them, so to keep the connection speed stable.

To choose which file to request from the servers (or in the case of LBRY, the LBRY protocol) you have a menu in the player. In this menu you have the list of resolutions available. And you can choose any of them.

The "trans-coding" it self, is the preparing of the video files. It can be done on the Odysee servers, on the LBRY Desktop application when uploading, or on a machine of one of the LBC miners. The point is, somebody has to put the processing power, to make those versions.

LBC miners trans-coding?

Mining is a game of who's computer solves a task faster. To get a chance to verify a block in the block-chain. But what if another rule would be added for the miners? What if trans-coding, and maybe other computational things could be a part of the mining algorithm for the LBC?

At the moment LBRY Inc. wants to implement ads to be able to pay authors on the LBRY network. And simultaneously to solve the problem with computational expenses of Odysee. Odysee is not the LBRY Desktop app. And all of it's computation are either happening in the browser, which is not very fast, or on a centralized server, somewhere. So if Odysee wants to implement trans-coding, it needs a server to make those files.

But if we could make the trans-coding itself decentralized, this could solve one of the issues with Odysee. I think everything LBRY protocol related that Odysee wants to implement, should be done in a decentralized fashion. Should it be trans-coding or anything else.

Beyond Trans-Coding!

If a resolution of a video is a drop-down menu that links to various files. Why not extend it a little. Make for example the automatic entries for trans-coded video. And give the user an editor for this menu. Where he could manually add or remove files, rename entries, or even nest menus inside menus.

One video could have only the resolution options. While another video, from a more advanced author, could have the same video in different languages, the trailer, the article version. And so many more things.

For example a movie could be a payed publication, with a gratis trailer, few poster files, subtitle files, various dubbed version, sound and image. This could be a very cool system. Why stop at only the resolution?

Conclusion

LBRY is a very cool idea. But we should not encourage the developers to implement things only on the most basic form. It's uninteresting. Hackers should hack. And make things that are exiting. Make things that empower users. Not trying to copy what's already there.

Happy Hacking!

3 Best Audacity Forks

Muse-Group, after buying the rights for Audacity, made an oopsie. And so they were forked. 1.5 thousand times. Let’s look at 3 best forks.

`lbry://@blenderdumbass:f/3-best-audacity-forks:9`

Proprietary software developers either do not understand Free Software, or intentionally trying to sabotage it. A company that bought Audacity, trademarked it's logo and name and asking kids not to use it, so they can spy on people, seems like a retarded company. But I tent, more, to believe that they are just cannot wrap their head around the concept of Free Software.

They "bought" Audacity and tried to treat it like any other proprietary software. There is a lot of money in data harvesting, so the first thing they did, as businessmen, is to add the data harvesting anti-feature.

Also, they are trying to get rid of the last thing that will make Audacity immune to their attacks. They are trying to get rid of the GNU GPL license, that is designed to protect peoples freedom. They are obligating contributes to give away their rights for the code. Without it, they couldn't change the license.

I'm glad to announce that people didn't agree with the company. More than a thousand forks appeared to save the project. To rescue it from the corporate, greedy bastards. They wasted their money. If you buy a Free Software project "upstream" branch rights, you better respect the people, or the people will fork away from you. And you will be left with nothing. It happened with CentOS. It happened with Open Office. It happened to countless other Free Software projects. It's happening now with Audacity. I guess they are retarded after all.

I'm going to list the entries by the amount of stars they have on Github. But I think the quality of those projects is unrelated to the stars. And we need to judge them, from now, by the ability to pull off good software in a respectful way.

Audacium

190 stars (by the time of writing) Github page Website

Audacium is a quiet project, slowly developed by some people. It's not a very much talked about fork. But people mentioned it more than ones.

I see this fork as a humble, bunch of nice folk, trying to do the right thing, without calling too much attention to themselves. And I admire it.

I think, in my opinion, this could be the right fork to focus on, since the two other ones are fighting with each other, too much, for no reason at all. While this fork stays calmly in the corner and keeps working.

Sneedacity

274 stars (by the time of writing) Github page

Sneedacity is a loud, animated, meme type, Audacity fork. They do know how to program. And it's a favorite of a lot of very vocal people. The fork is quite active. The development is steaming there.

But from the other side, Sneedacity looks like a bunch of clowns, trying to be loud, for the sake of being loud. I have no problem with that. I'm Blender Dumbass after all. But some people take issue with that. Also there is an entire newsworthy controversy.

Allegedly, developers of Sneedacity made hateful comments towards the developers of the other fork, Tenacity. And now, the developers of Tenacity want Github to "cancel" Sneedacity. Which is just funny. They are bunch of kids fighting for the title of the main fork. Laugh out loud.

Tenacity

3070 stars (by the time of writing) Github page
Website

Tenacity is by far the most starred fork. But people are claiming that this fork is done by incompetent developers. Most of the issues in their bug tracker is focusing of the drama between Tenacity and Sneedacity. And drama of Audacity it self. Rather than on bugs and fixes for those bugs.

For example, the now deleted issue #33 was a mess of yelling at each other swearwords and comments being deleted. And it was just another similar thing to my post, voting for the new name.

Conclusion

I'm glad that people doing the right thing. Leaving Muse Group, with their "upstream" fork, alone and sad. Teaching another proprietary software company the lesson of not to mess with Free Software.

I think Audacium is the best of the 3 forks. Since it's the most serious one. The most focused on making a good fork. And least focused on bragging about being a fork.

But people love Sneedacity and a lot of people starred Tenacity. I have no idea, which one will become the upstream fork soon.

Happy Hacking!

Questions People Ask From Richard Stallman

A lot of people have questions about the Free Software movement and what is okay and what is not okay. Some are fortunate enough to ask those questions from Richard Stallman him self.

`lbry://@blenderdumbass:f/Questions-People-Ask-From-Richard-Stallman:d`

A lot of people have misunderstanding about Free Software. A lot of people have questions about Software Freedom. And a lot of those people actually find fortune to ask them directly from Richard Stallman. Who is unbreakable in his answers. I want to present you with a couple of those.

All the questions will be taken from the lectures of Richard Stallman recorded on GNU Audio and Video website. So there you can listen, or watch, the lectures in full.

1:54:45 on Are We Facing Surveillance Like in China? (July 15, 2019 in Frankfurt, Germany) Also the video is available on LBRY.

Man with long hair:

> Okay, it is, I'm not very experienced as far as a data collecting, operation of data collecting is concerned. Ah... Try... eh... First is a question of maybe a minor importance... And that is, you said, that the mobile phones spy on us. So the simple question is, how can I prevent my, am, mobile phone, from spying on me?

Richard Stallman:

> You can't.

Man with long hair:

> Can't I just break it up and put certain parts on, maybe, the upper part of the kitchen, the battery...

Richard Stallman:

> Oh well, okay... If you are... want to do physical modification on it, maybe you could. But it wouldn't work as a mobile phone if it weren't spying on you.

Man with long hair:

> Alright, so when I'm sleeping, I might break it up and put away the batteries and then everything is alright?

Richard Stallman:

> You could. Or you could just, you know, you could just, throw it away, or give it to somebody else who wants to be spied on, or...

> I've never had one. I learned these things around 2003, because I was considering whether to get a mobile phone. And back then they were not "smart". They didn't have user-install-able software at all. All they did was the basic stuff. So that issue didn't arise for me, but I still had to consider whether they were mistreating me. And I found out that they were.

2:17:50 on Ethics and freedom in a digital age
(2019-05-14 in Zurich, Switzerland)

Person in the audience:

> Hi, am, is it a good thing that Microsoft is now adding Linux kernel to it's OS?

Richard Stallman:

> Ah... I don't know if it's good or bad. If you want Freedom, you must not have Windows in there. It may give... It may give some Windows users somewhat of an experience with having Freedom with some of their

> software. Unfortunately it wont tell them about the idea of Free Software. Because Torvalds doesn't support the idea of Free Software and he chose not to tell people about this. He only talks about Open Source.

2:14:05 on Copyright, freedom, and privacy (May 9, 2019, Copenhagen, Denmark)

Guy in Green Hat:

> Hi, so you mentioned that one of the 4 freedoms is a... the freedom to redistribute the software...

Richard Stallman:

> Well there are 2 of them. Freedom 2 is to redistribute without changes and Freedom 3 is to redistribute with changes.

Guy in Green Hat:

> Hm... How would you convince a business to allow their customers to redistribute the software?

Richard Stallman:

> Well, ah... First, a lot of businesses do it, so it's not that hard. But I'm am always going to present the issue of justice and freedom first. Because when you say "Oh this is the right thing to do, but of-course, since you are a business, you don't care about what's right or wrong, so I'm going to try to persuade you with some practical advantages of some kind." What you are effectively saying is "I'll go, I'll join your values." And I don't want it to appear that I agree with those a-moral values. I want it to be clear at every moment, that the main reason to make software free is because non-free software is an injustice. So I would want to convince the government... the... companies... by organizing enough people to say "We won't take your non free software!". So there... Ah... But, I mean, I'm not against using other arguments. And... the arguments of the Open Source supporters do achieve something when they convince a business. The place where I criticise is when they omit the deeper and more important arguments,

where they effectively endorse the idea that it's just a question of practical advantage.

Guy in Green Hat:

Thank you.

Richard Stallman:

So that requires care and attention if you want to present also arguments of practical advantage, while making it clear that you consider Freedom and Justice more important than that. You have to be careful how you express it and be alert to pitfalls, that might make it appear you enforcing the idea of deciding by advantage alone.

1:39:50 on What is Free Software? (May 27, 2015, Chania, Greece)

Guy in Green T-shirt:

The reason lot of software, which is ah... very special purpose. To give an example, in hardware, which is my field, FPGA software.

So you can do, place and route and ... specific technologies...

Richard Stallman:

Yes. That's actually, I don't know what you mean by "special purpose". That's exactly as special purpose as a C-compiler.

Guy in Green T-shirt:

Well, they make...

Richard Stallman:

A C-Compiler is... produces output to run on, computer. The FPGA tools produce output to run on FPGA. Ah... it's exactly, the special purpose-ness of both is equal.

Guy in Green T-shirt:

But, the scale is different. Because you have dozens...

Richard Stallman:

What is the question?

Guy in Green T-shirt:

Okay, can there be Free Software, ah... of the complexity, of ah...

Richard Stallman:

How complex is it? What makes you think it's complex? It's proprietary, we don't know. But people in the 80s would've said "Could there be Free Software at the complexity of the C-Compiler?" and I proved, the answer is yes. Ah... and now there is... now it's a lot more complex, that C-Compiler. Basically, we written very complicated Free Software. It's a matter of getting people to do it. And in some cases it's a matter of reverse engineering. In the case you mentioned, reverse engineering is needed.

Guy in Green T-shirt:

But, Reverse Engineering in this case would solve one specific instance. Part of the problem is that there is a product...

Richard Stallman:

Yes the same as, you have to... The C-Compiler has to know about each CPU architecture. It's exactly the same situation.

Guy in Green T-shirt:

Okay, if you make a back end, you can port it easily, right?

Richard Stallman:

Well, no, I had to work hard on making a back end that was portable. But the point is, you still need to know the specs of the hardware, you are trying to support.

Guy in Green T-shirt:

So you think it is feasible to have that kind of complexity with Free Software?

Richard Stallman:

It is feasible, but it's not easy to get it done. Because it's a substantial amount of work. This is one of the reasons why universities should teach reverse engineering.

Conclusion

Go to the GNU Audio and Video and get your self copies of videos. It's better then Netflix for sure. So many interesting arguments are thrown at Richard. He always knows how to answer them.

Happy Hacking!

How To Make Free Software The Default?

But how, how do we make Free Software the default choice for the most people on this planet?

lbry://@blenderdumbass:f/how-to-make-free-software-the-default:1

We all want Free Software to succeed. Even though Free Software is not about market dominance. It's about user's Freedom. It's still would be better, for the entire world, if there would be only Free Software.

Imagine a world, in which, a software company customer, that will see an unnecessary restriction in the license of the software, will freak out immediately. The same way people freak out at the cashiers for not receiving the discount, that was advertised to them.

But while almost everybody can look at the receipt and tell if they were scammed, not a lot of people know the difference of Free Software vs Proprietary. Nor they care about that difference.

I have a lot of colleagues at a real job place where I work. And while some of them know how to use a phone and are interested to try out new software. Others, on the other hand, do not know what the word "software" even means.

I want to discuss a strategy to move the current, largely proprietary world into a Free world.

Educate people on Hardware

Before there is Software, there is Hardware. In order to, adequately explain concepts like "source code" and "binary executable", you have to get people prepared. I started so many times, from assuming, that people know, what a CPU is and how it works, roughly speaking. And those times when they have no idea, they either try to look smart, nodding to my words, but later, not understanding a single thing I said, or they interrupt me with something like "I'm a simple person. I don't know what are you talking about."

Even I was a dumb child ones. When I was very young, I opened a computer. And saw the cooler on-top of the CPU. I thought that the cooler was the CPU. It has all kind of metal slots. It looked like, when you turn it on, it's filled with electricity and with some magic, this thing thinks. Of course the true nature of a CPU is much more boring. It's a little box underneath the cooler. That is just a huge circuit board, only done on a very tiny scale.

It's way simpler, for some cases to explain "source code" the way GPLv3 explains it. "A preferred form of the work to make modifications to it.". This is easier to grasp on some vague levels accepted in the court of law. When a question of software licensing is ultimately decided by a person that doesn't know what's a CPU.

I think an easy to illustrate idea could be done. Explaining the inner workings of a computer, without too much details. Just enough, that people could understand what software is. And how it's related to the hardware.

Educate people on Software

Next step would be do educate people, to understand software it self. And how it's made. A lot of people do not understand the production of software. When I was 11, I wanted to make video-games. I knew that programmers make games using keyboards. So I made a text file and wrote inside it "Make game". I was looking for this game for quite some time. Later I realized that programming languages exist.

We need to give people a very easy to understand example of a programming language. Something like

HTML, Python, JavaScript or even Basic could do that job. Showing people how a programmer thinks when making a program is also crucial. And interesting example of a video that does explaining in a fashion that I want to make, would be this one. It's a video from 1937, explaining, in a very easy to understand way, how a mechanical differential mechanism works in a car.

Maybe videos like this about Software, hardware and software freedom could be produced. But the approach should be that these videos should be clever and understandable. Not vague. Not a person angrily talking about software politics. It should be designed to educate people. People with various levels of prior knowledge.

A big part of educating about software, is to explain the difference between source code and a binary executable. To explain what it takes to reverse engineer something. And to explain technical things that might be very vague at some people's minds.

Educate people on Malware

Malware is software with a malicious feature. It may be Free Software or Proprietary. Only with Free Software you can edit the malicious feature out. (As shown lately by the Audacity controversy.)

But a lot of people are focusing on only a handful of malicious features. Usually focusing only on surveillance. When there are many others that are also malicious. And I think the coverage of this is not good enough. I have already written about them an article. But people seem, not too care about issues not related to privacy.

So if you are a publisher on Odysee and you are reading this article. I'm talking to you, @Nasikla, @Yal, @MattMadness @Mythologos, @polarhive, @tuxfoo, @OfficialZaney, @Pukima, @tech-pixels and others too. If you weren't mentioned, it doesn't exclude you. Please make publications about all kinds of malware. Not only surveillance.

The full list would include:

- Addictions
- Back Doors

- Censorship
- Coverups
- Deception
- Digital Restrictions Management
- Fraud
- Incompatibility
- Insecurity
- Interference
- Manipulation
- Obsolescence
- Jails
- Sabotage
- Subscriptions
- Surveillance
- Tethers
- Tyrants

People should know that malware means malicious feature. Any malicious feature. Not only spyware.

Educate people on Free Software

A lot of people do not know what are the 4 freedoms and why are they needed to begin with. Some people, including most of you, the readers of this article, know about the freedoms. But why are the chosen? What was the rationale behind them? Those are the kind of things people may ask when you tell them about the four freedoms. And what would you answer?

The two first freedoms are designed to give every person an individual control over their computing. Ability to run the software, study it and make changes to it. The other two are designed to help people who are not programmers. To give others copies of exact, or modified versions. So people could either hire somebody to make the changes. Or benefit from the changes, some random person had made.

Misconceptions about Free Software have to be debunked and addressed. Thinks like:

- **Software licenses**. What is the difference between copyleft and pushover licenses? What licenses are compatible with what licenses? And the fact, that Free Software licenses are just copyright licenses to begin with and not some end user license agreements.
- **Open Source vs Free Software**. People have to know that Open Source, while trying to stand for the

same values as Free Software, it failing at communicating those values. Since most of the Open Source supporters went the route of persuading others only by stating the practical advantages of releasing the source code. Not talking about it as an essential right.

- **What GNU is for.** People have to know that GNU, with it's Hurd Kernel, was the original Free Software operating system. That persuaded even the BSD to rewrite most of it's proprietary code, to create a Free Software operating system, so GNU could take peaces from them. Linux just became the major, working kernel for the GNU system. And that's why we call it GNU / Linux. And not just Linux.

Vote by your Wallet

Unfortunately we will be met with a very strong opposition. Proprietary software companies will not want to give away their power for our freedom. But there is a very easy way to make them stop being so nasty. If enough people will refuse proprietary software, just for the fact, that it's proprietary. It will, ultimately, make the companies give up. And make them release their software in the only way that will give them customers. It will persuade them to release all software as Free Software.

Instead of paying for a Windows license or an Adobe subscription. Why not Donate this same money to your favorite GNU/Linux distros and Free Software projects? If you will do that. The money will enable those developers to hire more stuff and ultimately make better software. If all users of Adobe will stop paying for the subscription. And instead transition to supporting the same amount to, let's say, GIMP. GIMP will outgrow Photo-shop in a matter of few years. And it will benefit not only GIMP, but all the other Free Software programs that, may use the new features implemented in GIMP.

You can start by supporting @Blender on Odysee with some LBC coins. And if you need a Blender Organizer. I may plug My Own Free Software here too. Also, you can help move the FSF train by Joining the FSF.

Conclusion

A lot of work needs to be done outside of making software. A lot of works needs to be done, by people good at explaining things. And a lot of work could be done individually by refusing to use something that treats you badly. And, ultimately, it can be done. Anything can be done. With a proper desire people change the world.

Happy Hacking!

Proprietary Software Companies Are NOT Better

A lot of people think about proprietary software firms as some kind of magical, all knowing factories, with millions of workers and best code out there. In reality they are not far from any other company.

`lbry://@blenderdumbass:f/Proprietary-Software-Companies-Are-Not-Better:1`

In an interview with Ton Roosendaal, the BDL of Blender, in a chapter called "Ton's thoughts on Autodesk" (46:53), he made an interesting calculation.

https://yewtu.be/watch?v=qJEWOTZnFeg

The part I'm referencing here stars on 1:01:35

Ton:

> I went to the public report from Autodesk. I looked at ... okay, what... how much money do they make? Okay, two billion per year. And then you get more information, you see, okay, the whole of the "media"... The media department of the Autodesk. That's where Mud-Box, Arnold, Maya, 3D Max and all those tools are. It's one hundred million dollars revenues per year. One hundred million, it sounds fantastic, but it's all of them. Imagine, half of that it's 3D max. It's fifty million. Max costs, 3-4 thousand per year? Divide that. 50 million divide by 4 thousand. That 12 thousand licenses.

Interviewer:

Blender gets more downloads then that in a months?

Ton:

Probably there is a million downloads every year, or so for Max or for Maya at the Autodesk website. And it's for training and for the students versions. But the paying user-base is not that big. And with that amount of money, they can't have a lot of people developing. So there is, maybe, 10-20 Maya developers. ...

Interviewer:

Wait? They have only 20 developers?

Ton:

10 or 20, something... For Maya, Max is bigger. But Maya is not that big.

Interviewer:

I would've thought that be like, hundreds.

Ton:

> And how would they pay them? This is not enough amount of money for it. Maya is maybe one tenth of the size of 3D max, or maybe 20%. Oh, come on, how big is it? You have one hundred million, you have to divide it, right? This stuff go to Mud-box, this stuff go to the compositing software... So what is, 10% is Maya? 20%? It's nothing. You can get 20 people or so... And of course you have to pay all the sales people and the marketing people and key-lock installators. And you have to pay the re-sellers. People that re-sell the software. They get a percentage. And then you have a fairly expensive booths at siggraph. And you add up all those things. That there is a little bit left for the developers.

The amount of contributors to Blender is way beyond 20 people. Blender of-course doesn't pay, most of it's developers. There is only a handful of people hired to write blender. Blender, from donations, makes just shy of 2 million dollars (at the time of writing) per year. It's less than Autodesk. But Blender has only one main peace of software (and a couple of websites, some of which have their own revenue streams, like Blender-Cloud). And they are not hiring sales people and other waste of money.

Free Software is not in any way, shape of form, worse then proprietary software. Both can be bad. Both can be good. It's matter of work being done. Let's discuss some strengths that Free Software has. And that can be exploited to get rid of the proprietary garbage.

Business is not about doing good products

A shareholder, who dictates the decision of a company, does so, based only on one factor. Will the share-value go up? The more the company brings per year. The more expensive is the share. The more the share-holder is getting from this. So all they care about is how much money they can make with this or that decision.

Even more than that. If a company has an opportunity to raise the shareholder value in some way, but doesn't act on this opportunity. The shareholder will most likely sue that company for sabotaging his shares. And will most likely win this case. This is why in a question of whether to collect user data or not, they could be good people and answer with no. But this would lead to a lawsuit. Making it illegal for a corporation to respect the customers.

There are other types of organizations from non-profits to Social Purpose Corporations, that do not have these same problems. But they suffer in a different way. Mainly, they struggle to interest investors.

But what does it mean in terms of software? Corporations only care, with software, about one thing. It's to sell you the program with the highest possible price, for which you are going to pay. Or if the software is gratis, to convince as many users to use it, then to collect as much data about them, as legally possible, then to sell that for as much as possible.

You are never a user of YouTube or Facebook. You are the used. The customer is the advertiser, data broker or other type of person, buying your information. You are a cow, that's giving the milk, for the dopamine. So if you see a non-free software project. YouTube, Facebook, Twitter, TikTok or anything else, that's gratis. Know that it's just a milking machine. Designed to make you comfortable with giving away everything about you. So some shareholder could profit.

They will develop it to animate pretty and to look nice. Not to do what you need. They will design it, to manipulate you, into thinking, you have needed it.

But what about software with price?

In the same interview about Blender, Ton Roosendaal also talked about implementing FBX format support for Blender (47:40).

Ton:

> I had a meeting with the [Autodesk] guys about the FBX. Because FBX it's the standard, right? ... For a lot of artists to move animation data from one application to another. Especially in the games industry everybody is using FBX. But that's a locked-in format. And Autodesk is frustrating everybody, who tries to reverse engineer it, by changing the format every year. Just for fun.

Interviewer:

Wait, who is?

Ton:

Autodesk. It's their strategy, yeah. They simply obscure things and they make things, to make sure that every body who has their own FBX thing, writer, reader, every year, you have to update things and fix it.

Interviewer:

Oh no...

Ton:

That's not fun, right?

Interviewer:

What did you tell them?

Ton:

I told them that. They "Mhra ma... arahraha... there's technical reasons... for that bla bla bla". They don't really admit this.

Interviewer:

Financial reasons?

Ton:

But you have to... What would be better, would be to say, we are going to do something for *open source*, or for our users. And make sure that interoperability between Autodesk products other products, that it's smoother. If the industry works together, it's a win-win situation. And they listened and they said that they will come back to that. And I never pushed this. And I never heard back. I think internally Autodesk is of course, ah... they have their own policy. It's a 2 billion dollar company. 2 billion every year. In revenues. So they are worth 10 or so... It's not a small business. It's big. They are on the stock market. So they are publicly traded. So they are not so, ah... ah... easy. Ah... I'm myself, I can decide everyday, what I want to do. But they, they have a job. And they have a boss and then a vice president and then the senior vice president and then the director, right? This is how big companies go. Those things are not so simple.

Interviewer:

Right

Ton:

And of course Autodesk is a company with, I think, with a philosophy that they want to lock in the users in their own world. And so what ever is possible with 3D, Autodesk, should give them the full package. And if they don't have something, they buy it. They buy it. Most of the tool, Maya tool, 3D Max tool, Arnold tool, flame I can't remember it, they are all bought. They didn't make it. They buy them. And then they stump it with Autodesk. And that's how they can create an infrastructure where everyone is happy and everything is beautiful and everything works together. But not if you use one of the competitor's tools. Of course that's how capitalism works, right? I mean you can't, ... you can't complaint about that.

The best argument that proprietary software has is that stuff works. An Odysee channel, and a friend of mine, Tyler Kelley, also known as @OfficialZaney recently tortured himself for a little bit, while giving Microsoft Windows a try. He found that the only good thing that he can say about the proprietary operating system is that "stuff just works". He is in a good condition. He suffered only a minor mental injury from giving Windows a try. And now he is healthy and back on GNU / Linux.

But they achieve the "stuff just works" thing, by the laziest and stupidest method. Making sure that every library they can get, will be available to the user. Not thinking much about how it all works together, or whether it even fits. You have this library, or we can make this deal, or do that. We gonna. Because we are "competing" here.

People will look at Blender with it's clean code and perfect, smart design of every feature. And will look at Autodesk with a bazillion features nobody uses. Bazillion formats naively supported. And with a little nudge from a sales person, the Autodesk wins. Microsoft Windows is installed even by a GNU / Linux user. And @Lunduke buys a smart phone.

The design of the software is there just to convince a person to buy. Just to make it easier to sell. Not to be a good product. Some may need a little more work done, to convince people. Z-Brush has put a lot of afford into making heavy scenes run faster. But in Blender, the developers are focusing on a different issue. On making everything right. Not on making everything.

And I know, that Blender is probably not the best example of it. It's a 3D modelling tool with a Video Editor build into it. Like, isn't it trying to just pack as many features as possible? Well, Blender is not a 3D modeler. Even though people think it is. Blender is a film creation tool. And editing is a big part of it. So it kind of makes sense.

Also, think about this. If you pay more for developing better, it will cut from the revenue. And thus will make the share-holders not very happy. So if they can find a trick that will convince people to buy for the same price, or higher, while making worse software, they are going to use that trick. Or the share-holders will sue them.

Conclusion

I didn't talk too much about the Software Freedom. I was discussing why proprietary software companies are not very good at making software. Free Software has a strength of anybody to contribute back. I know it's an argument of the Open Source supporters. But it's a very good, damn, argument.

It's true that both Free and Proprietary Software might be good. And both Free and Proprietary Software might be shitty. And even if their quality is equal. Free is still better. Since well. It's Free.

Assignment

If you know about a missing feature in a Free Software program. Think about this feature in a context of this Free Software program. How it might fit in. And how you would personally improve that feature forward.

Then, either develop it, in a plugin form, or as a pull request to the Free Software project. Or document this feature, in some way, accessible with Free

Software. (Like an article or a video). And publish it, so the developers of that Free Software project, would know about this idea.

In most Git repositories there is an issues page, or a discussion page, where you can post a feature request or a bug report. Do it. For Blender, there is a Right Click Select forum for Feature requests.

If we will act, we will outrun those proprietary bastards.

Happy Hacking!

GNU IceCat Web Browser - So Secure, It Blocks The Web

Have you ever though about how dangerous a regular web page is? This browser will show you.

lbry://@blenderdumbass:f/GNU-IceCat-Browser:3

A lot of people search for the best web browser. Something that will keep them safe from online attacks, tracking, cookies and other nasty stuff. Some people are stupid enough to simply believe a VPN ad, but we in the Free Software world will avoid any dis-respectful software. We know that trusting Microsoft Edge, Google Chrome or Apple's Safari is more than unreasonable. Even, beloved to so many people, Firefox, has fallen out of favor, with their latest scandals.

The Best candidates for a good web browser would be Librewolf, Brave and Tor Browser. Librewolf being a very good fork of Firefox. Brave having a tracking and ad blockers by default. And supporting Tor. And Tor Browser that blocks almost anything and routs through a special VPN-like encrypted, decentralized anonymization layer.

But what if I told that there was a browser that blocks so much nastiness, that it make the modern web unusable? What if I told you, that there is a browser that makes you think, that web doesn't need to exist? This browser is called GNU IceCat and it's a review of this browser.

Installation of GNU IceCat

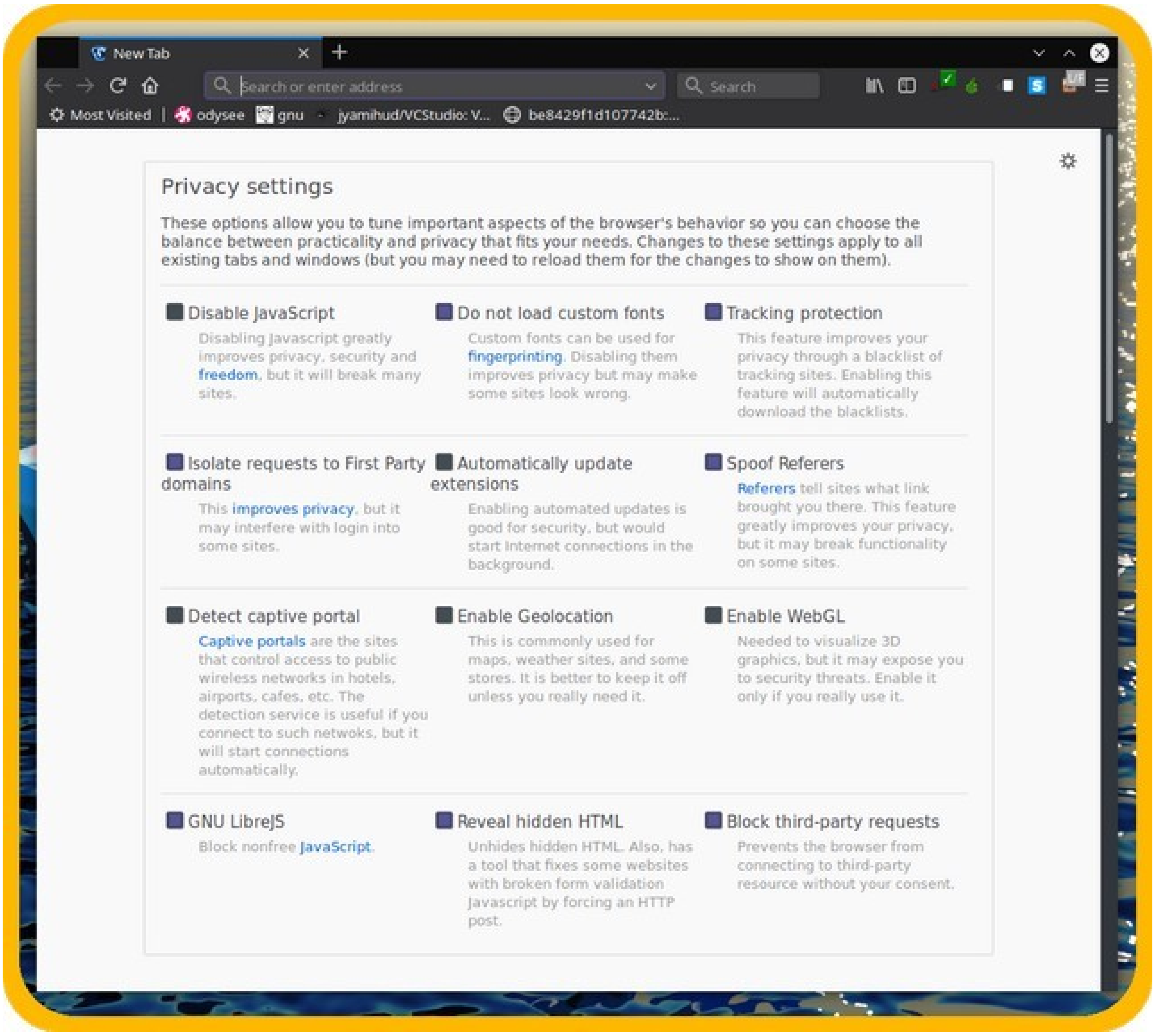
Most GNU / Linux distros will not provide you with GNU IceCat in the repository. It's because this piece of software is very scary. It's literally unusable for most people. Even though, that's my main browser. Most people will hate it. So if you go to your Ubuntu Software Center, you will most likely never find it there.

https://www.gnu.org/software/gnuzilla/

There is a GNUzilla page on the GNU website. That's where you can Download a binary build, or the source code of GNU IceCat. You have to, simply extract the package. And run the IceCat binary executable. It will launch the browser.

This browser is available only for GNU / Linux. May be you can build it from source on other platforms. I have no idea.

Welcome screen of settings



Privacy settings

These options allow you to tune important aspects of the browser's behavior so you can choose the balance between practicality and privacy that fits your needs. Changes to these settings apply to all existing tabs and windows (but you may need to reload them for the changes to show on them).

■ Disable JavaScript
Disabling Javascript greatly improves privacy, security and freedom, but it will break many sites.

■ Do not load custom fonts
Custom fonts can be used for fingerprinting. Disabling them improves privacy but may make some sites look wrong.

■ Tracking protection
This feature improves your privacy through a blacklist of tracking sites. Enabling this feature will automatically download the blacklists.

■ Isolate requests to First Party domains
This improves privacy, but it may interfere with login into some sites.

■ Automatically update extensions
Enabling automated updates is good for security, but would start Internet connections in the background.

■ Spoof Referers
Referers tell sites what link brought you there. This feature greatly improves your privacy, but it may break functionality on some sites.

■ Detect captive portal
Captive portals are the sites that control access to public wireless networks in hotels, airports, cafes, etc. The detection service is useful if you connect to such netwoks, but it will start connections automatically.

■ Enable Geolocation
This is commonly used for maps, weather sites, and some stores. It is better to keep it off unless you really need it.

■ Enable WebGL
Needed to visualize 3D graphics, but it may expose you to security threats. Enable it only if you really use it.

■ GNU LibreJS
Block nonfree JavaScript.

■ Reveal hidden HTML
Unhides hidden HTML. Also, has a tool that fixes some websites with broken form validation Javascript by forcing an HTTP post.

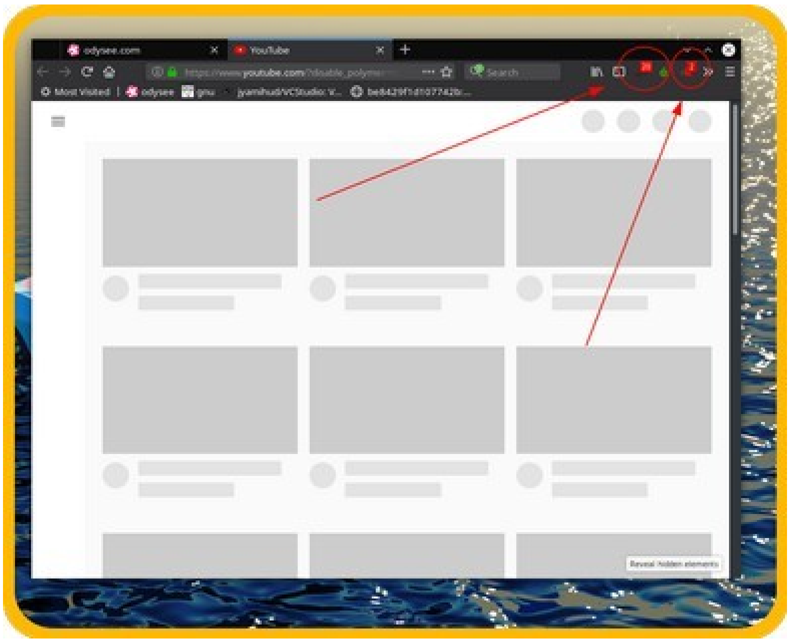
■ Block third-party requests
Prevents the browser from connecting to third-party resource without your consent.

Unlike other browsers that invite you to try and look up something on the web with a simple search bar. GNU IceCat instead is showing you options for what things to allow and what things to block on each run.

If you want it to work like a normal browser, you want to disable all of the "Disable" check-boxes and enable all of the "Enable" check boxes. The freedom is always yours. But of course, you want to enable the maximum protection. While disabling all of the stuff, you don't want. Or why the hell bother installing the IceCat browser anyway?

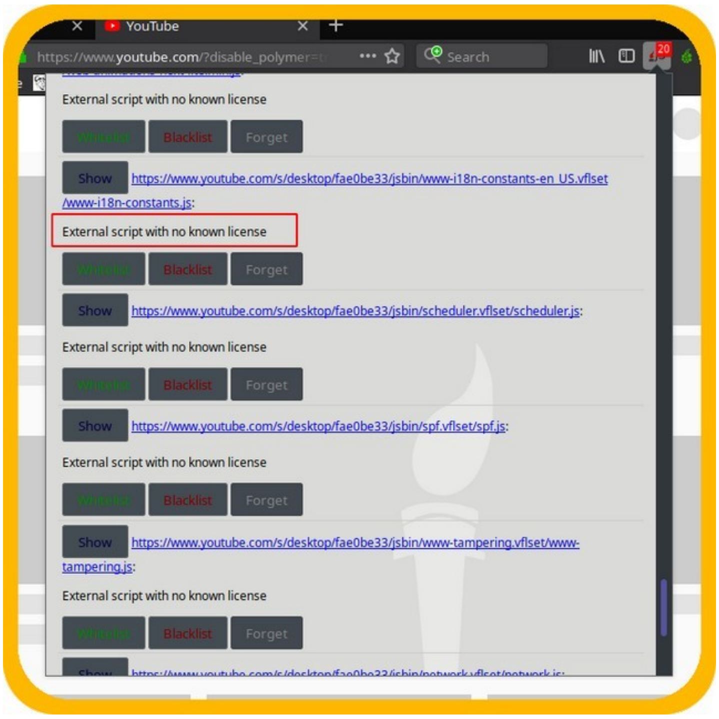
LibreJS

One of the most frustrating thing about the GNU IceCat is the LibreJS extension. I gonna demonstrate you what it does. Let's try and load our "beloved" YouTube.



Hm... It loaded only some grey squares. And nothing else. Where is the web-page that I'm looking for? If you are familiar with Brave and it's block counter in the top-right corner of the screen. Similar is happening on the GNU IceCat. You can see 20 was blocked by one thing and 2 was blocked by another thing. Most of the web-page from YouTube was blocked by the GNU IceCat browser. Leaving you only with some harmless, grey squares.

The first, 20 things, were blocked by LibreJS. The extension that blocks all, non-trivial, non-free JavaScript code. If you click that icon. You can see a full list of the scripts that the page was trying to load. And the reasons why some of those scripts were blocked. In this case, the script is merely having no recognizable Free Software license.



External script with no known license

https://www.youtube.com/s/desktop/fae0be33/jsbin/www-i18n-constants-en_US.vflset/www-i18n-constants.js:

External script with no known license

https://www.youtube.com/s/desktop/fae0be33/jsbin/scheduler.vflset/scheduler.js:

External script with no known license

https://www.youtube.com/s/desktop/fae0be33/jsbin/spf.vflset/spf.js:

External script with no known license

https://www.youtube.com/s/desktop/fae0be33/jsbin/www-tampering.vflset/www-tampering.js:

External script with no known license

https://www.youtube.com/s/desktop/fae0be33/jsbin/network.vflset/network.js:

It gives you an ability to read the script before, maybe Whitelisting it. So it could run. Or you can whitelist all the scripts from a given domain. For example, Odysee.com has problems to be recognized as MIT licensed with LibreJS. But I know personally that it's Free Software. So I can whitelist the whole domain of Odysee.com. And then it will work.

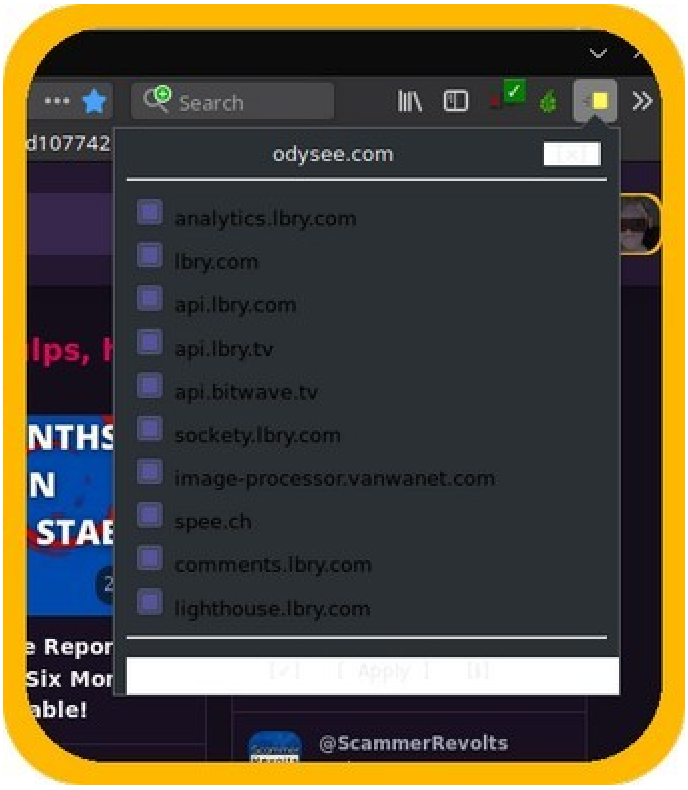
The reason for the LibreJS is documented in an article by Richard Stallman, titled The JavaScript Trap. Which explains the difference between readable JavaScript and the utter mess that big companies are loading into your browsers to obscure the intentions of the script. Don't believe me, try understanding this. Even though YouTube tried to fool LibreJS and added an "Apache License" to the code. It's still unreadable. It's useless. And it's probably malware.

And also, JavaScript is software. And software should be Free. So if you don't run Windows since it's non-free, if you don't run non-free applications on your free software OS, why would you run non-free applications in a free software browser?

Third Party Request Blocker

Many sites have tracking implemented, by making a very small request to a third party server. For example a small image, or a tiny script, could be loaded from googleanalytics.com which will make a recording of your IP address visiting that page. Where ever the page is.

So by default GNU IceCat blocks all third party requests. That's the other 2 in the YouTube blocking image. Let's look at the same extension when loading Odysee.com.



odysee.com

□ analytics.lbry.com
□ lbry.com
□ api.lbry.com
□ api.lbry.tv
□ api.bitwave.tv
□ sockety.lbry.com
□ image-processor.vanwanet.com
□ spee.ch
□ comments.lbry.com
□ lighthouse.lbry.com

I know the dark theme, I'm currently using, messes up withe the extension theme. But it's okay. At least it does the core job. The UI is fixable. Maybe I gonna patch it myself.

We can see that Odysee.com, to load all the stuff, all the scripts and all the images uses at least:

- analytics.lbry.com - handling view counts of publications and similar stuff
- lbry.com - the LBRY website
- api.lbry.com - some API for the LBRY protocol
- api.lbry.tv - some more API for the LBRY protocol
- api.bitwave.tv - API for the live-streaming protocol
- sockety.lbry.com - an API helping with connecting sockets
- spee.ch - an LBRY based, file downloading domain
- comments.lbry.com - handling comments
- lighthouse.lbry.com - search engine for the LBRY protocol

Some web-pages may contain stuff you will not like. Like googleanalytics.com or mc.yandex.ru that's found on Audacity website.

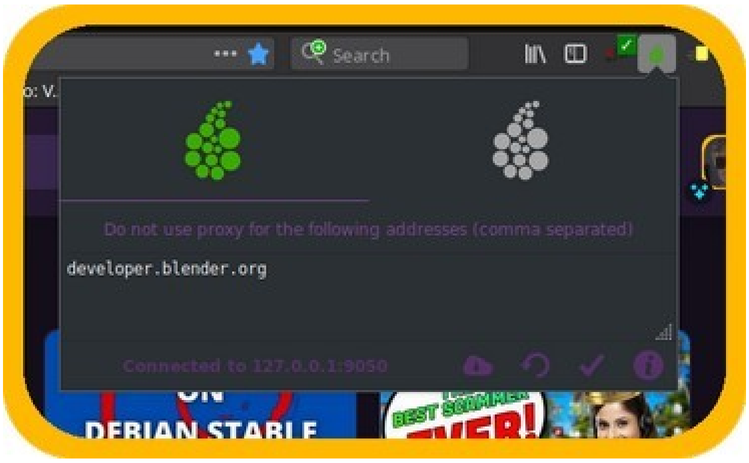
Of course, you have the freedom to enable any of those third party requests. But with some websites it becomes hard to do. Because of the shier amount of third party requests, each page is doing. This, together with LibreJS makes you think about a website, or a person / company running this website. If it's clean and amazing. And runs perfectly. You will feel good about them. If it requires you to load things from domains you never seen before. Or domains that are definitely malicious, like google. Then you feel bad about vising this site. And you will avoid it ever since.

Some good sites have problems with this feature. Like all the invidious instances. That are fetching the video files directly from YouTube. Which makes weird domains each time you request a file. This makes, a totally okay website, kind of unusable too. But the ones who did the bad implementation, were YouTube, not Invidious developers. And also, Invidious has a download button. Which links to the video file directly. Overcoming this problem.

I have Brave installed, just in-case some website with a good enough license and a nice functionality is blocked too much by my GNU IceCat. I love GNU IceCat. And most of the stuff I need, works just fine. But there are few websites that are kind of broken. Jitsi, Mastodon, Invidious and few others, just do not want to behave well when browsing with IceCat.

Tor

A lot of people make it a big deal that Brave has Tor in it. Well GNU IceCat also has a built in Tor.



Do not use proxy for the following addresses (comma separated)

developer.blender.org

Connected to 127.0.0.1:9050

You can enable Tor by clicking on it's extension in the tool bar. And choosing the filled in icon. Additionally you can add a list of domains to which, not to use the Tor network.

I run the Tor thing pretty much all the time. It's always on. Unless some dumbass site will refuse to load. And for some unfortunate circumstance I have to use it. Then I may turn the Tor thingy off. I never felt any drag or slowness because of Tor. It would happen more, a few years ago. But I think the Tor team managed to overcome this issue.

Conclusion

GNU IceCat will alienate most people. It's not some uncle Brave that's going to hold your hand, while you go into a store of unhealthy things and nudge you out of selecting a too sugared candy. GNU IceCat is not a warm dog. It's an Ice Cold and serious Tiger of a Cat. It will not let you into the candy store in the first place. Or if you will go there anyway, it will put on you a nuclear suit.

With the modern reliance of third party requests and JavaScript, GNU IceCat seems like the old, grumpy, boomer. That yells at kids for making TikTok videos. But with all that said, it's probably one of the most relevant browsers today. It's not only blocks bad things. But tells you exactly what it blocked and why. Giving you the full report. Being your personal secretary of defence.

Happy Hacking!

Game Modding - A Desire For Freedom

Most popular games are non-free. But players of those games still feel a strong desire for Freedom. Thus they mod games.

lbry://@blenderdumbass:f/Game-Modding-A-Desire-For-Freedom:5

Software Freedom is important. For me personally it's even more important than privacy. I mean, I wrote some very personal things on this channel. But I don't use non-free software. For me it not about being tracked. It's about controlling my computer.

Today, when I talked to a colleague of mine about this, he made an interesting point. As soon as it wasn't about privacy anymore, he started trying to understand Free Software in terms of game mods. And how popular those games are, that give the players some freedom.

I remember noticing similarities about Freedom and success early own. For example the James Cameron's highly successful franchise about the Terminator. I remember watching it as a kid and instead of focusing on the story. I was focusing on that dude, impenetrable by bullets. He could come into any place and do any thing without anybody stopping him. Total Freedom.

People understand that sometimes, some things are either too dangerous on their own, or there will be somebody willing to hurt you, if you do them. And watching a person that has no such restrictions is gratifying. Even if it's a cybernetic organism. Similarly, this may explain the current popularity of Super Hero movies. People that have abilities beyond law enforcement. A dream come true.

In the world of Software. Things that enable you to be a cyberspace Terminator. Things like Tor and BitCoin. Things that the most advanced government regulations cannot stop. Those things give you this sense of total control. Freedom and power. Make you feel, like you could enter a police station, during the night, wearing sunglasses and saying "I'll Be Back.". Just to destroy the entrance with a car crash.

In the world of gaming. Games that enable people to be psychopathic monsters, shooters and open world games. Especially GTA and GTA clones. Where you can go around the town, steal people's belongings and torture who ever you see. Like if you were the Terminator all of the sudden. Those games become popular.

People feel the need for freedom. Restrictions exist and people do not like them. Rules exist. And if just one of them will be removed, it would be gratifying. Unless of course, you designed the rule yourself. So even though people cannot stop using Windows and other proprietary operating systems, just yet. When approached from the right direction, they see why people want to go away from that garbage. Why people want to install GNU / Linux and use only Free Software. When approached from the side of Freedom. Not privacy, or quality of code. People tent to believe that it's important.

Maybe this is why a lot of hackers instead of hacking Free Software. Doing a hell of a hard work, reverse engineering and modifying proprietary games. Modding them.

Older Games

Most commercial games are proprietary. They are made only to make profit, either by selling copies, or lately by implementing all kinds of malicious features into them. But the older the games are, the more time hackers had to reverse engineer them. The more modifications to those games are available.

In the world of Free Software, the source code (a preferred form of the work, to make modifications to it) is always present. In proprietary software, including proprietary games. The only form you get, is the binary. Making it, not impossible to understand. But very complex. To such an extent that most people will not even try to do this.

But games reach such a connection to some people, that their restriction, becomes too annoying to some hackers, to suffer through. As I mentioned above, people are in need for freedom. And if you are using a piece of software often. You will feel in a need to modify it, to be yours. This need, together with a connection a game can do to a gamer, can lead to some hackers willing to suffer frustrating years, or even decades, of reverse engineering. Making it so they can finally modify the proprietary game, they are so much playing.

With some games it's easier than with others. Since, either the game is using standard file formats for the assets. Making it easy to modify. At least on in the visual aesthetic part of it. Or the game could have it's source code fully or partially available. With some very successful titles the source code of the engine is available. Making the reverse engineering of the rest of the game a substantially simpler job.

Maybe the most famous example of it is the original DOOM. The game it self is proprietary, but the DOOM engine is Free Software, released under the GNU GPL license. Enabling people to modify the game endlessly and port it onto many, many devices.

Now, you are probably asking "How can the game be proprietary, if the engine is under the GNU GPL? Isn't it forces the whole game to be under the GNU GPL too?". Well yes and no. The copyright of the Engine is on John Carmack. Who can do with it what ever he wants. Other people have the conditions of the GNU GPL. Which is a copyright license. A set of things a copyright holder, John, allows people to do with the code. And conditions with which it's allowed.

Of course if he accepts contributions from other people. He will not have the copyright over the contributions. And thus he also just has the GPL license. But the engine was released under the GPL after the game was already finished. So it didn't use any of the contributions from outside. Making it possible, only for certain people, to use this engine in a proprietary game.

Newer Games

Some engines are forced to release their source code, in some way, due to this, modding community. For example the infamous Unreal Engine 4. It's source is obtainable. But there is no Free Software license with it. The license is quite, non-free. The code is there to only look at. So game developers could understand the engine better. You can modify it slightly and give Epic Games the Pull Requests. But that's about it.

It's far from Free Software. But it's a step in the right direction. And I think, if people push just a bit harder. Epic Games might release the full source code of Unreal Engine 5 under an actually good license.

In some cases Game Developers do not use DRM and other nasty things. So the game modders could mod. But other game companies, that do not want to give up their power over people, just focus on different tactics at persuading the users into not having any freedom.

For example, a lot of game franchises, recently release only multiplayer games. There is no campaign. And since the main mode is the multiplayer, it's easy to use the cheaters and hackers, as kind of scapegoats, to force malware, such as anti-cheat software, to be installed as a kernel driver, monitoring everything on your computer. Turning your life into an Orwellian nightmare, just to prove that you are not a cheater.

Some companies are nicer than others. And it's easy to see who is nicer. Everybody agrees, to the point that it's kind of a meme, that EA Games is an evil corporation. Some of it's games are even banned by law in some countries, for shier amount of nasty. But, I think, developers are starting to understand modders and the need for Freedom. Some are just more evil than others.

Minecraft

There is a game that's having waves of popularity. It's a proprietary version of Minetest, called Minecraft. It's currently in the hands of an evil corporation. But originally was brainstormed by a hacker. The game's popularity is attributed to, you guessed it, Freedom.

Minecraft is not, by any stretch of imagination, a Free Software game. Minetest is. But that's besides the point. The whole concept of Minecraft is based on Freedom. You can literally craft what ever you want. Build worlds of your own. Modify worlds, other people built. It's Freedom. Only in a non-free package.

Minetest, I'm sorry, Minecraft, had allowed people to mod the game, without having modding knowledge. And even though, the graphics are simple, the game exploded through the roof.

There are source code versions of Minecraft, people leaked, or published, or what not. Non of them has a Free Software license. Except of Minetest. But it's more than enough ,for some people, to hack on the game and make mods. Making the game even more popular.

Conclusion

Levels of Freedom are there in the world of software. As Dr. Pito Sage said.

> I don't care if I have the four freedoms legally. Quite frankly, if I have the four freedoms even illegally, I'm happy.

And this statement makes perfect sense. You can modify and adapt source code released under the Epic Games EULA and never tell anybody. You can make mods, modify Minecraft, and live happily with DOOM on a calculator. Unless you are planning to release it, that's fine. I guess.

But in order to be able to share your experience with other. Which so many people want to do, a Free Software license, that's giving you a right to fork everything, is essential. I want to show you things I do with my computer. Not to keep them only to myself. Minecraft makes it hard. I'm not a Terminator. They will sue me. But of course there is Minetest.

Well, technically, I could use an alias name and publish the work anonymously through Tor. Or something in that nature. And then I would not be caught by the evil corporation that's trying to control me. But I want to be able to share things and say that I made them.

I love that modding exists. It spreads the hacking culture outside of Free Software. For me and so many other, Free Software is not only about privacy, or not even about the malware. Even though it's important. Free Software, ultimately is about Freedom. Freedom that we all so desire to have.

As shown by countless times, when companies tried to take our software into questionable directions. We always f**ked them. Free Software is the Terminator. It's an unstoppable, unkill-able machine. It will survive through anything. It will be Free and will Free us.

Happy Hacking!

LBRY vs Odysee

A lot of people seem to not realize a difference between a front end and an underlying protocol. This is my way of explaining it.

lbry://@blenderdumbass:f/LBRY-vs-ODYSEE:b

Guys at @Odysee have published a game that's going to enable people to win $500 worth of money, by making a list of strange videos. In the rules, they mentioned:

> Only rule: don't include any videos that would go against our community guidelines.

Which made a lot of people very angry. How is supposed to be a censorship resistant platform, if there are rules of what you can post? Aren't those rules, censorship?

LBRY the protocol

A few years ago, I was browsing the web and was thinking about a potential implementation of something like Tor and Torrent to make a platform. A platform on which everybody will be Free to post absolutely anything. And will keep their anonymity. Making sharing of music, full movies and extreme videos that are banned from all the other platforms, totally available and unstoppable.

I stopped thinking about this idea when I learned about LBRY. It's basically what I had in mind, but better. It's something that has a reward system. A search engine. And it's user-friendly. And it's Free Software. And they have it to such a point, that people are using it casually. To make things like this article, possible. It's not a dark web, full of drug stores, kill rooms and child pornography.

Technically, the LBRY protocol can be a place to host Dark Web stuff. But it's not build to be anonymized. It's built to be hard to censor. It has a different mission than the Tor. On Tor you are Free to publish anything, since nobody knows who you are and where you are. On LBRY, everybody knows. It's just hard to delete things.

You can think of the LBRY as a kind of Torrent 2.0. Basically, you download a file and while you have it, it's sent to more users. Also you can set up a server to host files and do similar things. But additionally, it supports a whole crypto-currency, it's built to be searchable, it's built to be a platform for publishing, rather than a file sharing system.

Also with LBRY finding files is way simpler. My channel is under lbry://@blenderdumbass or lbry://blenderdumbass:f. In which case the :f is a clarification code. Since there is more than one @blenderdumbass on the LBRY protocol. You can, not use the clarification code, in which case it will load the channel with more stacked LBC. My archived channel, with the videos I posted to YouTube a while back, has a clarification code of :c. Which makes the whole LBRY link be lbry://@blenderdumbass:c.

Any publication has an LBRY link like this. Which is easy to remember. I don't need to make people search for me. I know the full URL of the channel. I can tell people to look my articles at odysee.com/@blenderdumbass:f. Or for example, I know that most certainly, you can look of a buddy of mine's videos at odysee.com/@officialzaney.

Now you notices I used odysee.com and not lbry:// to give people my channel. This is because currently, as the LBRY protocol is still in it's infancy, it's not yet nativelly supported by web browsers. So a web API thing should be there to get channels and files.

SPEE.CH

When the guys a the LBRY were developing the protocol, a way of downloading the files and uploading them was needed. And so Spee.ch was born. Spee.ch is a very simple, downloading API for the LRBY protocol. It works in the simple web browser.

For example, my movie, I'm Not Even Human is on the LBRY url of lbry://@VCS:7/Imnotevenhumanshortfilm:3. In order to get the raw file, you have to change the lbry:// part to spee.ch/ part. Making the following link spee.ch/@VCS:7/Imnotevenhumanshortfilm:3.

If you clicked on the link. Your browser, had probably started playing the movie. It had no LBRY UI. No Odysee. Just the raw file it self.

LBRY.tv

Later they made two more things. It was the LBRY.tv website, that had UI for the player, the search, follow and support buttons, the account and all the good stuff. And in the same time, they made LBRY Desktop. Which is the same thing, but in an app.

This was supposed to be the way it is. You either use the LBRY Desktop, or the LBRY.tv website. Sign up, upload things, watch things. Tip authors LBC. All that stuff. But there were a few problems.

One was the confusion of LBRY with LBRY.tv. People thought that it's the same thing. And trying to explain the protocol, while the thing based on it, is named with the same name, is kind of hard. There is the protocol and there is the website based on the protocol.

Another problem, was spreading the LBRY in some countries, like Russia that had stricter rules on what's allowed. For example, on LBRY.tv all publications tagged as mature were available in a click of a button. Which was not very aligned with the Russian laws. So something had to be made.

Odysee.com

Odysee.com solved both of the problems. It's based on LBRY protocol. As it reads the same data, in the same way. But it has centralized features, that are not the part of the protocol. For example, censorship of some publications, tagged a certain way. On Odysee there is no porn. And things like notifications, likes and live-streams.

So for Odysee, there is community guidelines, they are related only to Odysee. Not to the LBRY Protocol. Since even if there was something like this to LBRY Protocol, it would not work. It's designed to be uncontrollable.

Multiverse

Something similar to LBRY would be ActivityPub. It's a network protocol similar, which makes sharing posts possible, but it's not a block-chain. Notable examples of the ActivityPub uses would be:

- Mastodon
- Nextcloud
- PeerTube

It's a way of making websites that all talk to each other, that use the same protocol to host data. But with different rules on each one of those sites and servers. For example, Mastodon alone has a lot of instances. PeerTube has a lot of instances. And so on. But since it's the same protocol underneath and the only changes are the UI implementations of the protocol. It allows for interesting things to be done.

Like following a channel on a video sharing platform with an account on a twitter clone.

LBRY as a protocol wants to achieve similar things. But the problem is, as far as I know, there are only 2 websites using it. And it's soon to be closed LBRY.tv and Odysee.com.

Actually LBRY guys encourage people to open new instances and making they own implementations. There is this LBRYgram project. Which is trying to make an LBRY based, Instagram clone.

Assignment

You don't like the Community Guidelines? Fine. Write your own. Maybe make it so only porn is allowed. Hell, we need a good porn site in the Free World too.

https://lbry.tech/resources/web-instance

Go through this tutorial and install your own LBRY server. By default, your site will not even host the files, so don't be afraid trying it out. You can always enable it to host files as well. And it would help the protocol. But while you are playing around. It's not going to do any harm, not to host files.

Happy Hacking !

Valve Steam Deck vs Free Software

Valve had released a gaming console of their own. And the decided to put a Free Software operating system on it. But there is a problem.

lbry://@blenderdumbass:f/Steam-Deck-vs-Free-Software:c

It's not a secret anymore that VALVe, the company behind Steam, and things like Proton, had announces, soon to be available, Steam Deck. A portable, gaming console, that's not a Jail, not a Tyrant and runs GNU / Linux. They announced that they are working on supporting things like anti-cheat malware and making gaming proprietary games on GNU / Linux, kind of good.

This is wonderful news for some people. For other people it's another Android, type device. Trying to appeal to GNU / Linux users, while being a proprietary thing. Others simply trying to understand, whether it's even good to begin with, to promote proprietary software availability on GNU / Linux.

Pros

Sharing a list of pros, like this, are hinting at a potential list of cons. And if you scroll down, you will see cons too. I'm not going to turn an essay, type article into a boring list, though. It's not my intention. I want this thing to flow naturally from one topic to another.

Obviously, the better the support for games, the better the argument for moving people over to our side is. We had a very tough time, when all you could do is basic Wine, some 10 years ago. Anybody who plays anything, would be, not ready to leave, their system of unjust power (Windows), since things usually didn't work.

Now, with VALVe's help in making Proton and even the anti-cheat malware, run on GNU / Linux, even if it's only for their console to work, the argument of "Games do not work", doesn't make sense anymore. We just show them the Proton thing, or the steam deck. And all prior misconceptions about inability of Free Software is gone. But is the software really Free?

Cons

One potential con I heard was that, with Proton being advertised as this all in one solution, for porting games from Windows to GNU / Linux, developers will not see a need to make a GNU / Linux native games ever. While I agree with this observation and argument. It's only a part of the problem.

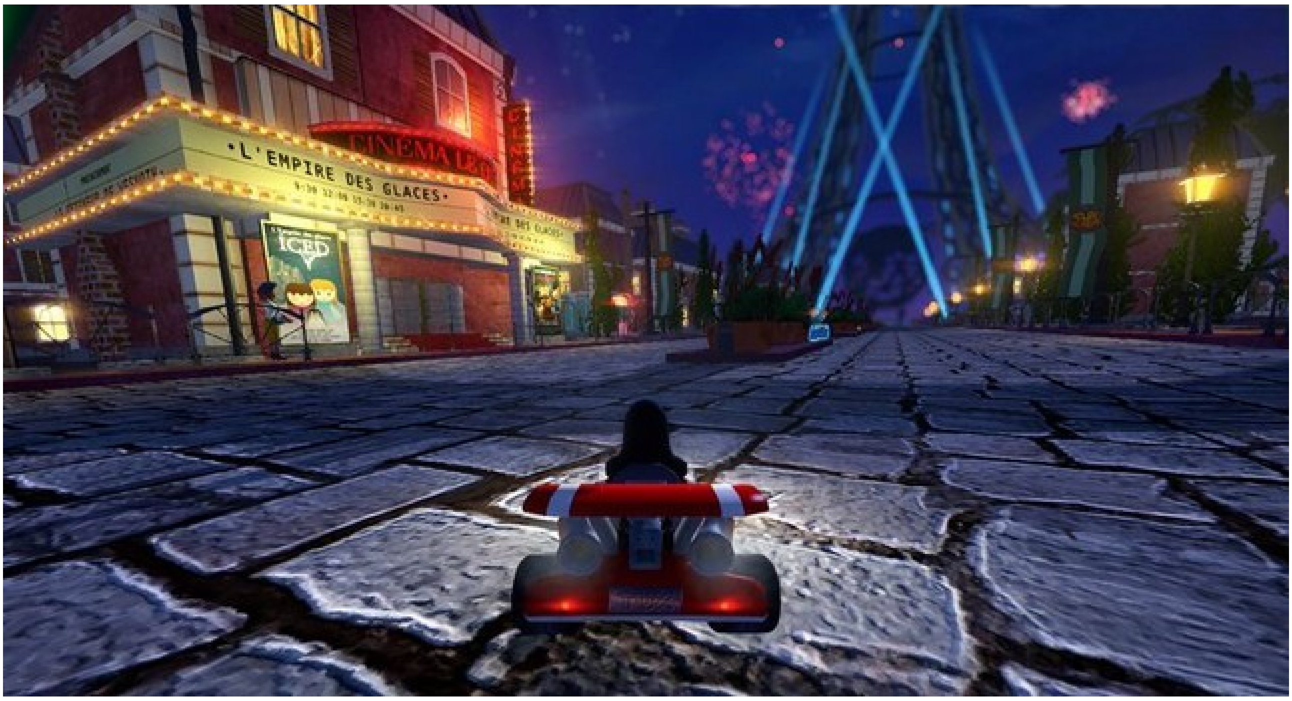
The 4 essential Freedoms of Free Software are getting violated by any proprietary software, either on Windows or not. The game it self is the problem too. Yes, we fix it a little bit, by moving people away from Windows. Solving one problematic software abuse, with a working alternative. But it's not enough.

Think about having 20 death traps in your house. Obviously, you change the house, to avoid all the death traps. But when moving, you insist on bringing the death traps with you. Because, you are familiar with those death traps. And if they do not work in the new house, the new house is "worthless".

This is the kind of cringe I have hearing about Proprietary Software or Proprietary Games support on GNU / Linux. I don't want Adobe here. I don't want Autodesk here. I do not want EA Games here.

But... We need Games...

Developing games in a Free Software manner, as in making them as such, that they keep all the 4 essential freedoms intact, is possible. And it's even possible to make a very good looking games like this. An example would be SuperTuxCart.



It looks amazing, and it plays nice. It has nice sound, nice textures, nice animations. Nice everything. It's basically a good game. Yes it's more for kids. But non the less. It's a nice game under a nice license (GPLv3).

SuperTuxCart is a fork of TuxCart. The initial fork from TuxCart happened in 2006. TuxCart was in development since 2000. It's 21 years ago. When, if you played the game, back then, you would vomit, from how horrible it was. It took 21 years of people contributing models and contributing textures and contributing other stuff, to make it where it is today.

Linux the kernel was a toy of a guy, who didn't think it was something serious. But in 20+ years of constant development, it's the most famous kernel.

Why, though? Why we just can't make a game good first, then release it? Well... It's because it's hard. Making a program on GTK that does a thing, is relatively easy. I've coded the VCStudio in about, only two months, till a working release. It misses features and it's not polished still. And I think, in a few years. It will, most certainly, have the polish and the features.

The problem is, when people do not have a reason to do something, they are usually sitting and procrastinating. For proprietary game developers, the reason is simple. They want to eat red caviar and blue cheese. They want to drive fast cars and snore cocaine from a body of an expensive prostitute.

Just kidding. It's not the developers. It's the share holders. Developers are usually getting a nice salary at a studio. Their job is to come and contribute things. And then go back home. They do it because they are getting payed. Regardless of how the game is released afterwards.

With Free Software games, it's either some people established a good enough foundation, to support paying developers like this. As with Blender, Linux the kernel and other things. Or the game is surviving on one guy, doing it in his spare time. And a few people passing by to help occasionally.

People do not feel like there is an incentive. So they are not in a hurry to do anything at all.

Making an incentive

It's a known fact, to some people, that Free Software, doesn't mean software Gratis. There are ways to make profit. It's just not very intuitive and requires a bit of cleverness. But I think, people are usually clever, if they can develop software.

There is a project that was, recently announced, going to be released as Free Software, presumably on the GNU GPL license. It's a game called Never Alone.

For a very long time, I avoided talking about this game, since it was developed using a proprietary game engine. But today, in this video (scroll to 13:50), the developer had announced that he is moving the game into Godot, in order to make it a Free Software game.

His incentive will be to put the game in various places, where people can get the finished, binary executable for a fee. And also he is going to publish the source files, to modify, and build using Godot. Basically having almost the same business model as Ardour.

BUT... I think, because he can still reverse his decision, we need to provide more motivation for him. And with this I'm going to test, my new section in this article.

Your Turn To Help Free Software!

https://odysee.com/@OfficialZaney:8?view=discussion

I just went to the Community Discussion of his channel on Odysee. And I just sent him a comment, with 100 LBC reward attached to it. The comment simply said "FOR DEVELOPING A FREE SOFTWARE GAME (NEVER ALONE)".

You don't believe me? Go and look it up yourself.

I hope you will do the same. Perhaps not with such an outrageous amount. You can put as much as you like. Give him 0.01 LBC. It doesn't matter. The rule is, the more the better. But if you have only little to give. It's okay too.

The idea is to make him know that we need this game. And we need it as Free Software. And if it's going to be a success, it's probably going to make the bigger companies, consider doing what he did and releasing their AAA games as Free Software too. Shareholders smell money. Well. Let's use it against them.

Happy Hacking!

"But Windows Is More Familiar"... Proprietary Software in Education

Imagine teachers would give students in school cigarettes and tobacco. This is what’s happening in school as they teach them proprietary software.

lbry://@blenderdumbass:f/proprietary-software-in-education:3

When people are thinking about moving to Free Software, they are usually, very anxious about loosing stuff, they are familiar with. They are happy to erase the Microsoft Windows operating system. But they are not very happy to move into the Free World fully.

Microsoft, them selves and Windows lovers, even use this as their argument to keep using Windows. "People are already familiar with it.", they say. Well it's true, but only partially.

I've participated in an interview, where I was a guest, on a podcast. I don't want to spoil it yet. I will re-post it when it goes live on Odysee. One of the topics we covered, was my brief experience with Windows 7. And how utterly unfamiliar I was with anything Windows.

My Windows Experience

I was about 15 years old and my mom bought me a birthday present. She wanted it to be a surprise, so I didn't know what it was till the day I received it. It was a brand new laptop with Microsoft Windows 7 on it. I was a GNU / Linux user since 2009. I was 12 when I got introduced to it. Before that, we didn't have a computer at all.

So I open the computer and see an operating system I hate so much. But an operating system that made me curious a little bit. So I made the dumbest decision in my life. And let myself use this garbage OS for some time.

The first I tried to do, was to change the theme. Well. I knew that there was no options for Desktop Environments, but I didn't know that there was no choice for custom themes. All you could do it set a color for the transparent parts of the windows. It was frustratingly restrictive.

I did install Blender and made a couple of scenes in it. And after some time, I finally decided to ditch Windows and installed Ubuntu 14.04. Which was way faster. To a degree that on Windows a Blend file was working with about 10 to 15 FPS. While on GNU / Linux this same file, on this same hardware, was giving me a full 60 FPS.

Educational facilities

Most people are familiar with proprietary software because educational facilities are teaching proprietary software to students. Most schools have a computer class with Windows. And they teach kids to use subscription based, malicious software, like Microsoft Office.

You could argue with schools (like I did when I was a kid) that GNU / Linux and other Free Software would be at least cheaper to install. Since, even though it's not the point, most of Free Software is also Software Gratis. But Microsoft and other proprietary software companies already had thought about it. They have campaigns and special offers to schools, to get their software licenses gratis, without paying a single dollar.

Why would Microsoft give up on an ability to make money? They could make a very large revenue, charging for a pro Windows license and an Office 365 subscription, from all of the schools. Well... They know a very sinister truth about it. If schools will teach their software, students will grow up, being dependant on it.

When the student goes out of the school he knows how to use Microsoft products. And perhaps he has an idea of how to use them. Let's say write a book in the office program. Now he faces the reality. While during the school, he had access to it, gratis. Now he needs to pay a subscription to use one, on a operating system, he needs to pay a professional license for. He was conditioned to be screwed by Microsoft.

And thus, for this student, the argument of "It's more familiar" works.

How to become a Drug addict?

DISCLAIMER: This chapter is only to illustrate a point. DO NOT TRY DOING IT!

So you don't want to become a drug addict. Who wants? You go to a party with some friends of yours. And you meet this guy. And he has dope. And you are looking away, avoid contact with this guy. Well. You don't want to become a drug addict.

But he comes to you and offers you to try the drugs without paying anything. Without any, so called, "consequence". He convinces you, that if you will try it, nobody will demand anything from you in return. You will just be able to walk free.

Curiosity kicks in. It's an interesting deal. A lot of people pay for this shit quite a sum. And now, he says, that for one opportunity, you can get this gratis. He convinces you that from one shot, no addiction happens. And you agree to take the shot.

For an unknown amount of time, you are in a very euphoric, narcotic, high state. You liked every single second of this stuff. Your brain changed some of it's chemistry. And when you come out of it and become sober. You suddenly want to try it again. Your brain demands more. Your system needs it.

You come back to that same guy. Asking to it try one more time. Believing that he is this, nice dude, sharing his dope, freely. But this is where he reveals you the sinister secret. He sells dope. And from now own, you gonna buy it. For what ever price, he puts.

This is exactly the same business model, proprietary companies do, when putting their malware into schools. They lure people into trying their dope, gratis. When later exploiting those people, arguing that they will not know how to use the alternatives.

Free Software is Familiar

Most people, realizing it or not, using Free Software casually, every day. From Wikipedia, the Free Encyclopedia, to VLC media player and things like Telegram. It's easy to find at least one program installed on somebodies machine that's a Free Software program.

And even when people are talking about some UI differences like perhaps, Gnome doesn't look quite similar to the Windows layout. Those same people, will, most likely install new games, with different UI layouts, every time. Or buy new hardware, that's not a computer, that has different UI, they need to learn every time from scratch. Not talking about the websites. With their constantly changing layouts.

People adopt to UI changes quickly. People learn how to use a different UI quickly. I've installed GNU / Linux to one couple, that had a first ever experience with it. I installed Gnome. They weren't complaining, that it's different. They were exited, that it's different. They were, ones again, happy to use the computer.

This whole "Users are more familiar" thing, is just a propaganda. Together with the business strategy to teach young kids to be dependant on a product, instead of being an independent adults. It's just few more nasty things, in a list of nasty things, proprietary software companies do.

Your Turn To Help Free Software!

Normally, I would ask you to sign a petition of a kind. To vote for the removal of Proprietary Software from education. But I think it's not going to do anything. I don't necessarily believe in petitions. They are just numbers. And they are far from an executive dude, making decisions.

I know a school here in Israel that decided to put Ubuntu on all of the computers. Because, some brave individual talked to them, convincingly, about all of the topics, I just discussed.

What I will do next, is, I will call a few schools and I will try to convince them to move to GNU / Linux and Free Software. I will offer my help in installing the software, and giving the teachers the understanding they need. Even if non of them decide, it's still and idea worth spreading.

I would like you to do the same. It's probably not very hard to find a phone number online, of an educational facility near by. If enough of us will talk, we will end this shenanigans.

To make it a little more interesting. Let's start a Tag on both Odysee and the Fediverse. I would use #FreeSoftwareSchools (or tag FreeSoftwareSchools on Odysee). And share your adventures with us. Even if you didn't convince anyone, trying is still a step forward.

Happy Hacking!

Let's Help Pablo Vazquez with the GNU / Linux Problem!

A fella GNU / Linux user had a problem with his microphone. Using this post I was trying to figure out how to help.

`lbry://@blenderdumbass:f/Help-Pablo-With-GNU-Linux:f`

Usually I'm talking about broader topics on this channel. Covering things like the Philosophy of Software Freedom. Or making huge observation about some topic related to it. But sometimes, little things rub me the wrong way. And this is one of those topics.

I'm a subscriber of Pablo Vazquez, a Blender employee and commentator of updates, which he does every week on the @Blender channel. There is already a problem that Pablo, despite Blender's policy on using Free Software where ever possible, is still using YouTube as a streaming platform. And on the Odysee Channel, we get only the replay, that's the LBRY protocol syncs.

For a very long time, more then 10 years, Pablo was a GNU / Linux user. But not so long ago, he installed the problematic, Microsoft Windows 10. And a couple of latest streams he did from that.

I want to discuss with you the problems he ran into, that lead him into installing a terrible OS. I want to discuss a possible solution, that could make Pablo, fix the issues, without switching the OS. And finally, I want to do something about it all. And encourage you to help me.

The Problems

On May 3rd, 2021 (two months ago) Pablo made this stream. Where he talked about a technical problem, he has, with the USB microphone. Scroll to 11:10 into the video.

> I don't have mate today, because I was dealing with this audio problem. Sorry for people that were watching this live. I look at the amount of tests that I had to do. For, eh... for ya... for... trying to get the sound right. Or I don't know what. Sound and Linux never got along. I don't know why there are so many, ALSA, Pulse, Jack... There must be good reasons, but ah, it's a bit ah, annoying.
>
> However if I move to Windows for streaming, then I have other issues. So that's more of a problem.
>
> Hm... Maybe somebody that is familiar with Linux, the problem I have is that pitch sometimes goes down. When I start recording,

> it's like ohrohrohroh, I speak, like this. So I could actually stream, but my voice would be even worse. So if you have, eh... any, help... you can help. If it's a known issue, leave it in the comments.

Two months later, at July 5th, 2021, he made this stream, using Windows as the operating system. During the video, he confirmed that the reason he uses Windows, was that same, audio problem, mentioned above. On Windows, everything is apparently works fine.

This is a known issue

For making the streams, he uses a USB microphone, with larger bit-rate, then your regular, cheap, mic. Those are usually made with sound quality in mind. And are not very understood by most GNU / Linux operating systems.

I had this same issue too. With a BeyerDymanic Fox. Sometimes it either recorded me at lower pitch, or played the sound back with weird artifacts, clicking noises and other nasty stuff. The problem, as I understand it, is that the microphone is trying to give too much data, than the current sound buffer allows. Making it exceed the buffer, which creates weird sound artefacts.

There was a tutorial, I found, almost two years ago, that showed a configuration file, where you could setup your ALSA to stream at the rate that the microphone supports natively. Alternatively I found a hack. If I run music constantly on the background, with a very low volume. Enough for the mic's sound system to pick up, but not enough for me to hear. It will keep the bit-rate stable and the artefacts will disappear.

Though, how this hack works exactly, is not yet clear to me. What I would do is to run random songs with Lollypop. And make the volume of them almost zero. And it would fix things.

Your Turn To Help Free Software!

I like to encourage people to help Software Freedom in these articles. I would ask you for two types of favours. You choose which are more suitable for you.

If any. You can do both, if you have enough spare time.

Technical. First, there is the technical side of things. People can search online. Use what ever search engine you like. Some are better than others. If you have knowledge, let's gather it. If you know people who know things, ask them. If you can read the code, to find what the root of the issue is, let's do that.

I want to know exactly what is this problem. I want to know what causing the lower pitch. I want to have a full report on the issue. And with this issue, properly researched, I want to do, two things.

- Make a detailed, step by step, explanation, so we could provide Pablo and other people with a temporary fix.
- Make a bug report, so this issue would be fixed. Contributing fixed code, would be an extra step, that I would also encourage.

Social. On the other hand. I want to persuade Pablo to keep using GNU / Linux. He is a front face of Blender for a very large part of the community. And people should not be encouraged to use malware, to fix technical issues. They should be encourage to fix the issues directly.

Pablo Vazquez is available on multiple social media. From Blender.Chat to Odysee. To Blender.Community and other places. He uses Twitter, which is non-free. But there are Free Software programs that talk to Twitter. That may allow you to communicate with Pablo that way.

Please word, your argument to stay on GNU / Linux wisely. And to make it more interesting, let's use a tag. #PabloVazquezBackToGnuLinux or PabloVazquezBackToGnuLinux on Odysee. Let's convince a 10+ year long, member of a GNU / Linux community, that we are not going leave him behind. And we are more than interested in helping him. His problem should be eliminated from the face of the earth.

Happy Hacking!

Proprietary Software Companies are Digging Their Own Graves

You may think that they are unstoppable with their money and "premium" status. But in my opinion, they are driving fast on the highway to hell.

`lbry://@blenderdumbass:f/proprietary-software-companies-are-digging-their-own-grave:c`

Yesterday I've got mixed feelings of fear and anger, when I watched the announcement of the `#TakeBackOurTech`. In the video, Ramiro stated that there is a patent by Microsoft, about a proposed system of getting an employee work-ability score. This system will use a webcam, to read facial expressions, from the people participating in a business meeting. And based on their AI system, it will give each participant a score. Using which, a boss, could decide, who to raise and who to fire.

With in the same video, he stated that one of the requirements of the Windows 11, is a high quality web-camera. It's assumed that this technology will be implemented in all Windows 11 installations. Making it an ultimate spying system.

Think about being called for a job interview and the boss is quickly typing your ID and Name into a computer. He bought the access to the Microsoft database. And he can see your employee work-ability score. This score will decide whether you will even start the interview, or you will be fired immediately. And think about, what would the boss say, if your data is not in the list. If you used a Free

Software operating system like GNU / Linux, that doesn't constantly spies on you.

I think, while it sounds grim, the situation is not that bad. Yes, what they are trying to do is bad. But it's not all lost. And I'm about to explain you why.

Shier amount of insanity

When the Free Software movement had started, back in the 80s, most proprietary software was more or less good software. It would not spy on you. It would not restrict you, more than, by not giving you means to edit it. It would not be malware. It could be. And there was no way to verify it. But programmers just weren't as outrageous.

There a quote:

> Power corrupts; the proprietary program's developer is tempted to design the program to mistreat its users.

It's from a page of Malware on the GNU website. They have a large catalogue of instances, when proprietary software was malicious.

When software was proprietary in the 80s it was only as a mean of capitalism. Nobody would know how it works, so nobody could re-implement it. So one company could control a function of a computer. This is still true, but developers started realizing something.

They have power to put into the software anything the like. And if the software is proprietary, the user has no way of dealing with it. First, they might not even have a way of knowing about it. But even if there is a way to analyse the changes that the software makes, and the signals it sends. Ultimately, the user has no control. He knows that he is running malware. But he has no choice. Since with proprietary software, he cannot edit it.

Some companies have tried sneaking this kind of control into Free Software as well. There was a whole case about one program released as Free Software, that was signing the files with a crypto-graphic key generated from the binary of itself. Making it able to read files only if the software it self is not changed. This is lead to an additional clause in the GNU General Public License version 3. Which states that the software should be designed in such a way that a simple act of modification will not result in it being useless.

During the 80s, software was mostly okay. And now, we have malware in every big tech program. And it's going to be even more apparent. Even more outrageous. Even more insane. There are so many examples of it, that a whole website could be made to document them all. And it's good.

The good

People are now very familiar with the terribleness of Facebook. And it lost a lot of it's useds, when people started realizing what Facebook is doing. YouTube's censorship and shier insanity, made Odysee (and a few others) a thing. Windows 10 was the reason, so many people switched to GNU / Linux. And Windows 11 is even more of a reason to do so.

Those proprietary software companies, by disrespecting people systematically, are digging their own graves. Since people will just all go away from them. Even with a situation like Audacity. People just went to the forks. Tenacity, even though not the most liked fork, and not having binary executables yet, still managed to attract 80% of the Audacity contributors on Github.

Those business executives are so far from the real humans, using the software, that they believe the charts too much. Yes, Windows has the most market-share in the world. This is why they feel like it's unstoppable. But with enough shit, they will loose the market-share. Since people will move.

Computer users are like water in a steam machine. You can either keep it cold. And the water (users) will stay. Or you could heat it up a little and some steam will escape. Proprietary software companies steam machines are about to explode from heat, letting all the water escape.

Imagine how hard it would be to talk to people about Free Software if proprietary software companies respected people. If their software was actually good. It would be way harder. And with each addition of new malware, new exploitation, new bullshit. They are helping us, with examples, to argue against them.

So, while, it's true that the stuff they are doing is not good. And I would not tolerate it. But ultimately. They are helping us quite a bit.

One proprietary program I would develop

I was walking outside the other day and thinking about a software project that would be proprietary, but that would make sense to be proprietary.

It would be an app that is advertised to do something useful. But ultimately will be a little, annoying bully, on your device. It will be designed to harass and annoy the user as much as possible. And every now and then, it would show a message that says something like "Are you annoyed yet? With Free Software, you could edit all the annoying functions out. But, sorry, I'm a proprietary program. I won't let you change me."

This would be a perfect kind of educational tool, to show people what we, in the Free World, think about proprietary software. But I think, this program is not needed after all. Since, well... Proprietary Software companies are doing this job for us. And soon, people will just snap. And either stop using computers. Or move to Free Software.

Your Turn To Help Free Software!

In this section, I usually encourage people to do something good to help Software Freedom. Two articles ago, I asked you to donate LBC to a guy developing a Free Software game. And it was a success. You can still participate and donate more to him. But I'm afraid he is going to get 5 starts on Odysee before I gonna get 5 stars. I had 4 and he had 3 when it all started. I still have 4. And he has now 4 as well. Do something about it. IDK.

One article later, I asked you to talk to schools about teaching Free Software instead of proprietary software. And it turned out to be a bigger problem. Scroll to the comments to see the issues. It seems like it's easier to persuade them to teach both. Which is okay. Some schools do dual booting. And it also seems like a problem is usually higher. At the government level. Since, Microsoft was lobbying quite hard.

Yesterday, I asked you to help me fix the microphone issue that Pablo Vazquez had. It spawned quite a reaction from the Blender Community. I hear about it being discussed outside of the Free World. In Discord and other places. On Blender.Chat Pablo joined a special chat. He didn't say anything so far. But I'm guessing, it sparked his interest as well.

With a couple of people, we found out that we have not enough data. And we would like to be able to recreate the bug somehow. Pablo is busy. It would be great if he could volunteer his time. But it seems like, it's not an option.

If you, by any chance have a USB mic, running GNU / Linux and want to help. Send us a message in the special chat. So we could troubleshoot everything. Try out solutions. And ultimately come up with an easy tutorial to fix Pablo's problem.

I don't have a special request today. But I would link you the help page of the GNU project. So you could decide yourself what you want to help with.

Happy Hacking!

How To Argue About Free Software

A lot of us want to explain our friends and family our passion with Freedom in software. But sometimes you are caught ahead of the time. Do you want to know how to argue better about this topic?

`lbry://@blenderdumbass:f/How-to-argue-for-Free-Software:0`

A lot of us are trying to convince our friends and other people to move to a Freedom respecting Software alternatives (Free Software). You, most likely, argued with people in defence of GNU / Linux. And you had either persuaded somebody, or not.

In this article I want to look at some techniques you may use in order to persuade people to use Free Software. Keep in mind that we are not trying to force anybody. We want people to free themselves from the proprietary garbage. It's not gonna be Freedom if you point a gun at them and threaten them with a bullet if they keep using Windows.

This is a guideline, not a rule. It's an advice, not an order. We all have our own minds. And we all can use our minds to come up with arguments on the fly. Reading this article may give you and easy way out.

Messengers

Recently I bought a book, printed in paper, to avoid DRM. It's about the art of persuasion, but written from a point of view of scientific research. It argues that the message it self has less to do with it's success, than how it's being communicated and by whom. The book is called "8 Ways to Get Heard - Messengers".

The book is arguing heavily about the persuasion factor of a person talking. And it breaks apart this observation, with links to research, into 8 different categories. I gonna summarise them, but in order to get the full picture, I suggest you to read the book.

- **Socio-Economic Position**. A lot of science agrees that people listen way too much to the opinions of the elite. It's true that celebrities endorsing a product may increase it's sales dramatically. This is why big tech companies want famous people to use their hardware and software. And this is why I urge you to help me with the Pablo Vazquez situation.
- **Competence**. People listen to experts. If you say that Richard Stallman had suggested something. Or that Linus Torvalds had suggested something. Pointing out that they are very good at the field, ahead of the argument. It may increase your chances at persuading people. Research shown, that even if people ask to be introduced as competent. It's still increases their persuasion factor. This is why on talks, there is a person who lists the achievements of the speaker. You may ask your

friend to introduce you as the guy that knows a lot about software. Which will make your chances at convincing people higher.

- **Dominance**. It's is very true that Dominant speakers get heard. It's not very good to use it all the time. But if an argument is made, by a speaker, to which you disagree, saying your counter argument louder, may silence the other speaker. Giving you an advantage. But you have to know exactly what are you saying. Because if you don't, you will be perceived as incompetent.
- **Attractiveness**. It may sound unrelated. But research shown that people that look nice, usually have more persuading power. So dress nicely. Comb your hair. And get a shower. There is only one exception to this category. And it's Richard Stallman.
- **Warmth**. If you are talking with a non-hostile opponent. Perhaps a friend, who's not getting, why he should use Free Software all the time. Being nice to him. As in, not forcing him. Will go a long way.
- **Vulnerability**. Being open about mistakes you make, or bad situations you are getting yourself into, might cause other people to want to help you. So do not mask your weaknesses. I don't know how to

relate it to Free Software, but I guess, when the arguments comes, you may use it.

- **Trustworthiness**. It's not very convincing, if you try to persuade people to use Free Software and then, a second later, you use proprietary software yourself. So if you post things to, say, YouTube or Twitter. Make people understand that you don't want to do that. But, you just don't see any other way to reach them. (Using a, non official, Free Software client might be even better)
- **Charisma**. It is obvious that Charisma is a very good tool at persuading. But what is charisma? Well... it seems like nobody knows. The book describes it as "Charisma involves a constellation of characteristics: self-confidence; expressiveness; energy; optimism about the future; rhetorical ability; an ease with risk taking; challenging the status quo; and creativity (to name a few)." Another few things, the book is pointing out about charisma is that charismatic people like everything to be good. Despite fighting with evil. Think of Richard Stallman talking about the good Free Software more then about the bad proprietary software. Also, charismatic people tent to reward people, they agree with. Like when we got together and donated LBC to a Free Software game

developer. And another little thing. Charismatic people usually think fast and answer immediately. A lot of people want to get the answers right. So they think too long. It may loose momentum and be a disadvantage. You have to know your stuff, so it's slips of your tongue.

Know your stuff

In order to persuade people, you have to know what to answer, if there is a question that's unexpected. For this you have to know the subject on an intuitive level. I think, in order to learn about Free Software, you have to use the source.

There is the GNU.ORG website, that I would recommend you to dig into. Especially in the next four areas:

- GNU.ORG/PHILOSOPHY. It's a collection of explanations of what and why of things Free Software.
- GNU.ORG/LICENSES. This is a more hands own, implementation level of Free Software. It's the reasons why to use a certain license. And

information of what is a Free Software license to begin with. And why it's needed.

- GNU.ORG/MALWARE. It's a handy resource of instances, when proprietary software developers included something nasty into their software.
- AUDIO-VIDEO.GNU.ORG. It's a collection of lectures and interviews by Richard Stallman and other people related to the Free Software movement. I urge you to listen closely to the Q&A sections in the end of the lectures. It's usually, both fun and informative. People ask Richard all kinds of interesting and challenging questions. And he always has something to answer.

Alternatively, you may use any other resource of information. The more you know the better. You can start by reading all my articles. And clicking on accounts in the comment section. They might have their own, good stuff.

Careful wording

Free Speech is important. This is why I would remind you, that it's all just a list of suggestions. And if you see it doesn't fit, you may not use any of those suggestion. People are often misunderstand the FSF for even having guidelines of what to say. And how to call things. But it might get you out of hot waters. When people are trying to catch you on the word.

In this part I would suggest you to use the following things:

- GNU / Linux instead of just Linux. Since there is almost nobody who uses the Kernel directly. People use software. And on GNU / Linux the software (even the bash terminal) are parts of GNU. Not parts of Linux. Alternatively, you may call the operating system by the name of it's distribution. Ubuntu, Arch, PopOS... and so on.
- Free Software, or Libre Software instead of Open Source. People are usually have strong misunderstandings about the Open Source. It gives an impression that Open Source, simply means, that the source code is available. Free / Libre Software, on the other hand, is all about the 4 essential freedoms. You may need to explain people (in English) the clarification for the word "Free". I often use "Open Source" as a kind of strategy to get them quickly to the ballpark of the idea. And then switch to Free Software and continue from there.

There is a whole list of those on GNU.ORG in an article called Words To Avoid.

Again. Remember. I'm not saying that it's has to be done this way. Or that you can't use the words, you like. It's just a list of suggestions. Since they may save your ass in a hot argument situation.

For example, with the words "Intellectual Property" listed to be avoided on GNU.ORG. You may end up in a situation where the person arguing with you, uses a similar phrase:

> You are talking about Intellectual Property?

And you may answer:

> I'm talking about the copyright law specifically. I never use the words "Intellectual Property". It sounds absurd.

Ways to start

There is almost always a computer near by. A phone. A laptop. A TV. Something that runs software. You may start by asking people about the device. I usually talk to them about Telegram. I know it's not the best example of Free Software. Since it has tons of issues. But it's a good example of a successful Free Software. That's probably already installed on most people's phones.

You can start pointing out the similarities between Telegram and Whats-App. And pointing out the differences.

This is only one suggestion. Ultimately, you need to use your chances, when they are presented. And do not to be shy to talk about it.

Your Turn To Help Free Software!

Tomorrow I will be working, my regular job, at a different place. They asked me to help there for one day. And I yet to meet all of the employees there, with whom I will be working. What I'm going to do, is I will try persuading them to use Free Software. Or least educating them into knowing what it is and where to find more information.

I urge you to try doing the same. Talk to, at least, one person. And then, return to this article and write a comment about the experience. Let's see, who will get the most interesting comment.

Happy Hacking!

"It's All Lost!" - Wrong!

People tent to think that we are lost already. No.

lbry://@blenderdumbass:f/Its-all-lost-wrong:a

A lot of people seems to poses a certain believe. A believe that we're all doomed. That the "moral battle" is lost. That Facebook and Google will take over the world, because people don't mind. They talk about all those same issues, that we are talking about, but from a perspective of an apocalypse. They talk about it as if we have nothing, we can do about the issues. I strongly disagree with those people. Because there is a way. It's just requires doing something.

Everything is possible. Some things are just hard.

A lot of people like to procrastinate and think about doomsday scenarios, as a kind of rational for their procrastination. I've seen a very interesting quote by a businessman. It seems unrelated, but if you give it a thought, it is related. I, unfortunately, don't remember the name of the businessman, or the exact words he said. But the meaning of it, would be something like this:

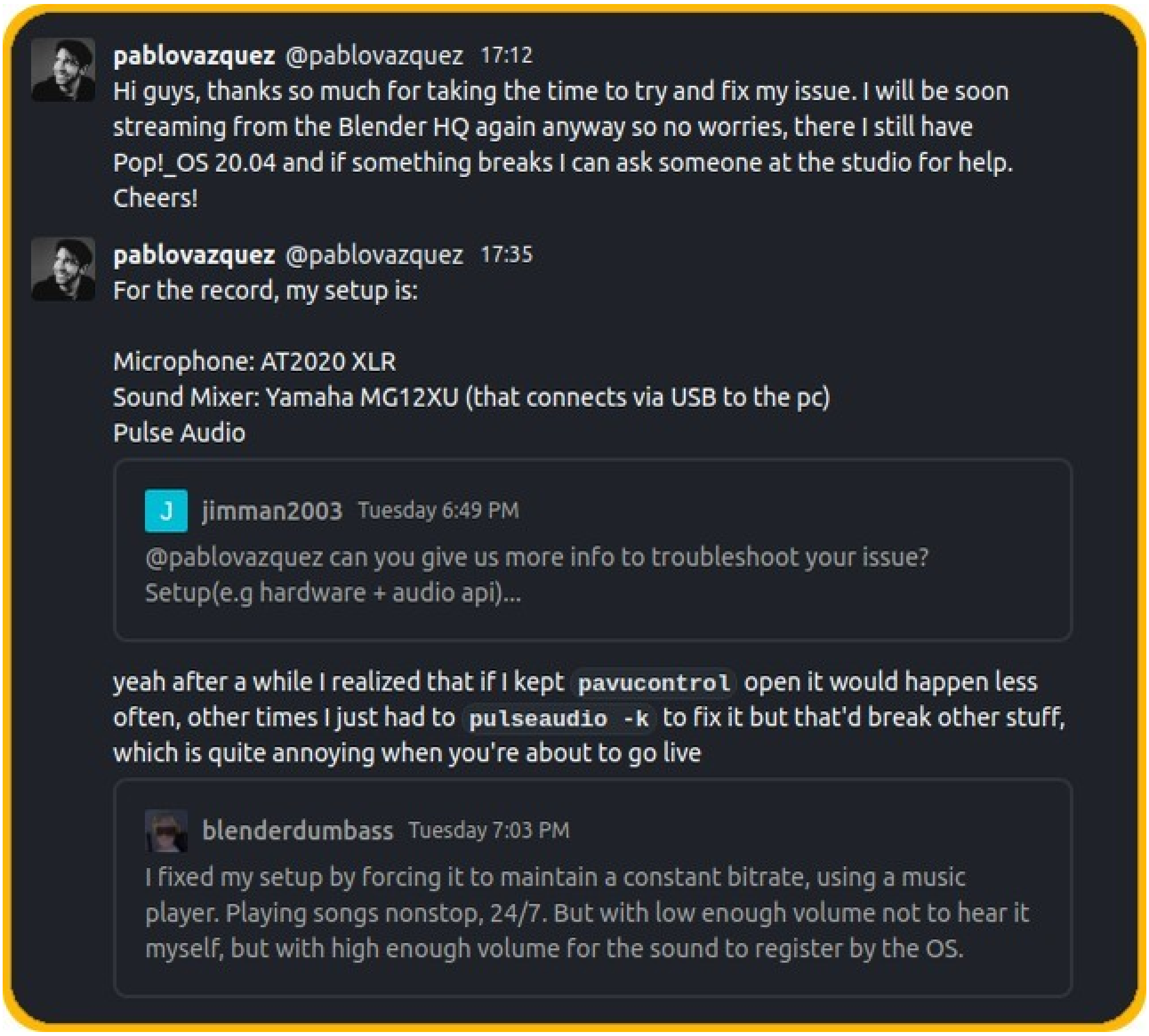
> A lot of people say "I don't know how to do it.", when they have a device in their pocket, that can access the information. And it's not

> even about being scared to go to the second page of the search. It's about actively looking for an excuse to procrastinate.

People are lazy bastards that will do anything they can, so they could keep sitting firmly in their chairs and receiving dopamine without doing a thing. But this is possible to overcome. Since there are people that achieve things. There are people that, go and find information and then do a thing.

Think about James Cameron. He designed a ship, in which he gone to the bottom of the Mariana Trench. He shot some of the most ambitious films in the last few decades. Think about Richard Stallman. He didn't like the direction of software development in the 80s, to such a degree, that he literally made the Free Software movement.

Or think about us. Yes. This channel and the followers. You. We donated to Free Software Game development. We returned a fella GNU / Linux user (Pablo Vazquez) into continuing using GNU / Linux, by finding information about his microphone problem. Pablo had finally answered. This is from the Blender.Chat thread:



pablovazquez @pablovazquez 17:12
Hi guys, thanks so much for taking the time to try and fix my issue. I will be soon streaming from the Blender HQ again anyway so no worries, there I still have Pop!_OS 20.04 and if something breaks I can ask someone at the studio for help. Cheers!

pablovazquez @pablovazquez 17:35
For the record, my setup is:

Microphone: AT2020 XLR
Sound Mixer: Yamaha MG12XU (that connects via USB to the pc)
Pulse Audio

> J jimman2003 Tuesday 6:49 PM
> @pablovazquez can you give us more info to troubleshoot your issue? Setup(e.g hardware + audio api)...

yeah after a while I realized that if I kept `pavucontrol` open it would happen less often, other times I just had to `pulseaudio -k` to fix it but that'd break other stuff, which is quite annoying when you're about to go live

> blenderdumbass Tuesday 7:03 PM
> I fixed my setup by forcing it to maintain a constant bitrate, using a music player. Playing songs nonstop, 24/7. But with low enough volume not to hear it myself, but with high enough volume for the sound to register by the OS.

It seems like he is coming back to GNU / Linux. Though, it's unsure, yet about his personal computer. Since the problem is not fixed quite yet. But at least the following streams will be done from within GNU / Linux. Which is a fantastic news. BTW. He also gave us the info about the hardware he is using. So, how about looking for the solutions even further? I think, this silly microphone problem has to end.

The idea is, people think that it's their nature to be lazy and that's why they will not achieve anything. But it's not very true. I mean, yes, people are animals. And all animals, when not hungry nor scared, are sleeping. But sometimes, with in the smarter animals, another thing pops in. Curiosity.

Curiosity

Think about this again. A dude wanted to see, how it is, on the bottom of the Mariana Trench. So, what he did? He worked hard for multiple years, to design a ship and conduct a research mission, that could potentially end his life. And all, just to know about a thing he was wondering about. "How is it, on the bottom of the Mariana Trench?"

People can go very far to answer a question. And not even a question that makes sense to answer. Question like "What if X?" are very interesting. This is what, the whole science fiction genre is built upon.

With the Free Software, "what if X" is way more interesting to think about, than with proprietary software. Since you don't have to build a whole peace of software, usually. It's easy to just modify something, to satisfy your curiosity. What if there was a Download button on Odysee too? Well, now there is one. What if there was a game, or a web-browser, or an Email client inside your favorite text editor? Well, there's Emacs.

People think, it's all doomed.

I know a lot of people, who share many of views with me. People who even talk at Free Software conferences. But people with no believe in a way to fix things. They see everything as going constantly towards hell. And while it's understandable why they think this way, it's wrong.

It's a scientific fact that the overall trend of goodness, freedom and things like that in the world increase. It's just not very noticeable, since it takes hundreds of years to feel a noticeable effect. You can think about Proprietary Software as a recent experiment that goes wrong, but a lot of people are still blind to it going wrong.

Let's look at a Free Software adoption over the years. And I'm not talking about the Linux Kernel. It's not my metric. I'm talking about new startup project that go Free Software. They have increased dramatically. Some of them see Free Software as a way to make development easier. Some see it as a moral duty to do Free Software. But it's, non the less, happening.

Think about this. Software existed for less then a hundred years. While humans existed for millions of years. Software is very recent. And as soon as it appeared, Free Software movement appeared. Humanity, even though has a few bad cells, ultimately has a very strong immune system.

People that are afraid of Free Software

There are people that avoid Free Software, for the reasons, other people use Free Software. They are scared from the big brother, the constant surveillance and such. But since they, usually don't understand software much, they have a kind of interesting train of thought. Let me demonstrate:

> Big brother knows everything about me, so he knows if I use Free Software, or he knows that

> I don't use the software, that he wants me to use, that's why I should not use Free Software. Since it will make the big brother angry. And I gonna suffer from that.

Other people, the most paranoid of which, believe that Free Software is developed by the big brother. And it's advertised as being secure, so they will start using it to store secrets. And the big brother will know those secrets this way.

Let me give you a short tutorial of how the computer works, so you could debunk all of those, crazy claims.

How the computer works?

A transistor is a very basic electrical switch. It usually has two input cables and one output cable. If both input cables have voltage. The output cable will output an electrical current too. Making it a basic AND gate.

An AND gate is a logical operation in computer science. If you have two inputs that can either be True or False. And you need to produce one output. If it's an AND gate. It means both inputs have to be True, it order for the output to be True also.

But since you can wire the output of one gate into an input of another gate. Wiring one transistor into another and have endless designs of how everything is connected. You can start building gates that are more complex.

There are gates like OR that returns True if either of the inputs are True. NOT that returns True if the input is False. And so on.

With this, wiring one transistor into another, making an electrical design, carefully constructing a flow of electricity, you can make a logic board. And logic boards could be small, as in making something simple, like turning on the lights in your house. Or it could be something complex. Like the CPU of a modern computer.

A CPU, or a Central Processing Unit, is just a very large, and complex logic board, but only with microscopic transistors and millions of wires. It's wired to read a specific type instruction, sent as impulses of electricity. And it has a very basic list of things it can do.

The things are so basic, that it's literally phenomenal that it's even working to begin with. So for example, modern CPUs have pins connected to the Memory (RAM). And an instruction could be "find a bite in location X on the memory". And the next instruction could be, "execute from this byte". And this bite, could have further instructions like this.

If you have access to the instructions that the computer is doing, you may not even be close to understanding what program it is. Since it's nearly impossible to know. But today, we have compilers. A compiler is a program, that is translating something written in text, to a set of bytes, readable by the CPU. A compiler would have a predefined set of rules, a code, in which a programmer, could implement features, so other programmers, would understand the features.

In order for the big brother to access the information on your computer. First, the computer has to be connected to the same network, as the big brother. Let's say, the internet. Also the computer should have an instruction, in software form, to understand the internet. Okay. Most operating systems can do that. A program should have a feature, to respond to a specific command from the internet. So called, Back Door. And this command should be sent.

If you don't have internet, you are safe. There is nothing to worry about. If you don't run software that you cannot verify for back door, you have nothing to worry about too. And this is why Free Software is the most secure software. Since you can verify that nothing malicious is implemented in it. And if you are using only Free Software. And you are careful. You are way safer, than with the malicious software, the "big brother" (proprietary software companies) want you to use.

So if you are one of the people that are afraid of Free Software, since you will be, looked at, as a criminal, stop it. All Free Software does, is giving you ways to verify things. And also other essential freedoms, like you can use the software, or give a copy to somebody. There is nothing criminal in using Free Software.

We are not doomed!

Everything is possible. If you refuse things that look good on the surface, but disrespect you inside, you will have less problems. Alternatively, you can always stop using computers entirely. But there is no reason. Free Software exists. Alternatives exist always. Sometimes, they are not advertised, or not as obvious. But they do exist.

People should start using the ability to look for information, to look for information. If you are frustrated with something. You can find a solution. It's not lost on you. Especially with Free Software. It's software that you can fix, if it's broken. A software with a bug report page. A software with community of people familiar with it's source code. You just have to find them sometimes. Maybe talk to them. Yeah, I know, talking to people? What a shame....

Stop being a procrastinating fat-ass, writing angry comments in the comment section. Come help yourself. And come help us. Open your mind and think.

Your Turn To Help Free Software!

I've talked about Curiosity for a reason. We have good software in the Free World. But I think people can always add to it a bit of their own flavor.

What I would encourage you to do is this. Find a thing, you are interested about. It should be not very serious. It should be related to software. Implementable in some, already existing Free Software. It could be a new feature. It could be an interesting UI design. It could be literally anything. Any type of modification to an existing project.

Find the project contributor. Either on Github, or somewhere else. And start a thread about your modification. And explain a possible way to implement it. Send to the comment section the link with your thread. And let's see who has the best one. (An extra step, would be to develop it and send a link to a pull request)

Happy Hacking!

Which Distro Is The Best?

Forks, branches, distros...

lbry://@blenderdumbass:f/Which-Distro-Is-The-Best:0

Free Software with the freedoms 2 and 3, let us create forks. This makes software decentralized. But it creates a problem. Do you like Snaps? Most people don't. But there was a reason, why it was made.

I personally do not agree with Snaps. It's one of those, Telegram-like software. Where all you get to mess around, is the client program. The server is proprietary for one simple reason. They wanted to create a centralized software hub for all GNU / Linux distros. Alienating all of the distros in the process. Making it so, it's *pretty much* only Ubuntu, that uses Snaps right now.

But they did, tried to address a real issue. An issue that was brought to me by this article, earlier today. It's talking about the *elitists* in GNU / Linux user forums, that yell at each other, with their personal software choices, instead of trying to help people, fix their issues. It's true even with the Pablo Vazquez situation. One of the first answered to his mic problem, were asking to switch to a different audio handling software. From pulse-audio to pipewire, as an example.

I don't think it's necessary a bad thing, to have so many choices. But there is a problem, that I want to address. And a potential solution, that I would like to try.

Monopolies vs Free Software

Despite attempts at restricting monopolic growths of companies, they still grow quite monopolic. Monopoly is when one entity, a corporation, or a single person, controls one market, fully. For example, if there was only one manufacturer of telephones. This one manufacturer would hold a monopoly on telephones.

It is also true that having a system that let's monopolies to exist, doesn't promote competition. The same competition that makes capitalism work to begin with. In order to give you the full picture let's talk about capitalism. And how it differs from let's say communism.

In a communism, money either don't exist. Or has very little value. And resources are spread more or less equally, between the people. The entire resource economy is controlled by the government.

And makes the government, also the single, big, monopolic corporation. It can choose who gets what. And in what amount. People get goods if they work, or in other ways useful for the government.

In capitalism, money is everything. The more money you make, the more stuff you can do with it. The more things you can buy. Nobody is forced to work. But also, you get money by working. And only if you have money, you can get goods. But with capitalism, the arguments is: Since anybody can sell anything to obtain money. The competition for making the best product, will force innovation. Of course it's true, only in one case. Where there are no monopolies.

While there is more than one company producing telephones. There is, for example, only one company producing an iOS operating system. Also Microsoft has *a near* monopoly in a space of personal computer's operating systems. But not a monopoly, quite yet. They have a complete monopoly on Windows. Since there is no other Windows out there. You can switch. But if you install GNU / Linux, like so many of us did, you will not get the same level of compatibility.

Free Software is where monopolies break. Anybody has freedoms to modify the software. And everybody has freedoms to re-distribute the software. Free Software sounds against capitalism. But it is, after all, pro-capitalism. Since it supports competition.

Coming back to the cow I milked to death already. The Audacity scandal. With Free Software, since anybody can give people the same service as you are, you have to respect your users, or they gonna switch. This is how capitalism was forcing the innovation. People will bye (use) the stuff, that's better. You can't lock down a Free Software program and make any malicious feature, since you don't have a monopoly on it.

Dark side of constant forking

A fork is a term used in the Free Software world to describe a split in direction of a particular software vision. One group of people decided to develop the program in this direction, implementing these features. Another group decided to do something else. It's a fork.

With forks of things like a whole operating system, comes a problem. Since anybody can put anything he wants into his fork. And you can fork the systems endlessly, you end up with a confusing set of wires. Instead of a coherent whole.

On Windows, the user experiencing it on one computer, can be confident that all the same things will work in exactly the same way on the other computer. The same is true for macs. But on GNU / Linux. Each user has his own system. And it's a weird concept to adopt to.

There is though, a way of illustrating it to a new user. And I would use a metaphor for a house. Each person has a different layout of rooms. Different sizes. Different amounts of rooms. Different things put into those rooms. And placed in different locations. If your house, would be designed by a company like Microsoft or Apple, everybody would get the same exact thing.

With distros, you can turn one into another

While with houses, in order to make modification to an existing layout of rooms, you will need to destroy walls nor build walls. With software, especially Free Software, modifications are way simpler. Anything a distro does, is an inclusion of default, preinstalled software and settings for them. So if you delete enough software, install enough other software and change enough settings. You can turn one distro into another.

For example there is one or two commands, one needs to type into the terminal, to turn Ubuntu into PopOS. And from that point onward, the system is officially PopOS. It will use the PopOS repositories. It will use the PopOS theme. And it will receive the PopOS updates.

Some systems are father away from one another. Like Ubuntu vs Arch. And to turn one into another, will require doing more setups. Sometimes, installing a system from scratch, will be less of hustle, than doing it, like that, manually, after the installation.

Also, since the Linux kernel, could be modified, changes and installed differently, on a running system (perhaps with a need to reboot). It is probably possible to even move this way between Linux based, HURD based and BSD based systems.

But it's probably a stretch. Since there are way more things that going to break from such attempts.

Also there is a logistical problem of having dependencies in a correct version will all the software installed. Since most software do not include libraries that could be included elsewhere. Like the PNG library to draw images. Or the GTK library to draw the user interface.

Snap, Appimage, Flatpack

With the dependencies, comes one more problem. Sometimes, a person needs to use two different peaces of software, simultaneously. While both are dependant of the same dependency. Only with a different version number.

All three, Snaps, Appimages and Flatpacks solve this issue by making a containerized software package with all the correct dependencies. While Appimages and Flatpacks made their design quite decentralized. Enabling further forking of formats. Canonical, the company behind Snaps, decided to make it a monopoly. Making only the client software Free and making the Server side completely proprietary. So while Flatpacks and Appimages could be distributed by whoever, Snaps are distributed only by Canonical.

This has pissed off a bunch of people. And a bunch of distro maintainers. So even distros based on Ubuntu (Canonical) are not using Snaps. For example *GNU /* Linux Mint uses Flatpacks.

Your Turn To Help Free Software!

In this part of the article, I usually encourage people to do something good to help software freedom. I would encourage something this time too. We have a problem of communicating the choices. Distros, Desktop Environments. Software distribution models. Forks. All those things need some way of getting to the people, that try out our Free Software.

I would split the problem in two:

- **Communicating Choice.** We have to find ways to communicate to people that come from outside, that our software is ultimately, personizable. @Nasikla, in his article (mentioned above) suggested Memes and humour as a way to spread this idea forward.

Maybe simple communication. Like explaining what's Free Software first and only then showing the GNU / Linux distros, could be a easy road to take.

- **Documenting Choice.** Finally, we have to be able to set the new user on his own journey. Letting him go. Letting him figure out the settings, the forks and the desktop environments that he might like. I think, we need to either find, or design a website, or a peace of software. Something like Wikipedia in accessibility. But ultimately a Free Software catalogue of sorts. Documenting different distros, different desktop environments. Having a tree of forks. With dates and explanations. Having a search bar, where a new user can type what ever he has a problem with and the website will give him a suggestion of software or a setting, that might be the closest possible fit for his request.

#BlenderDumbassChat:matrix.org

I've just opened a little Chat room on the Matrix protocol, where we can discuss, how it could be all done. Go grab your favourite Matrix Client. And...

Happy Hacking!

Who Controls Your Computer?

Software – Instructions to a computer. Who ever controls the software, controls the computer. Do you control the software?

lbry://@blenderdumbass:f/who-controls-your-computer:c

When Richard Stallman does his lectures about Free Software, he starts usually from a question to the audience. Who controls your computer? The answer seems to be obvious. The user controls the computer. Right? Well... No. Computer does not understand the user. It understand only one thing. A program. And who ever has control over this program, ultimately has the control over the computer. The question is, whether you have control over the program.

Free Software with the 4 essential freedoms insures that users will have control over their programs. But only if all the software on the device is either Free or made by the user. The user controls that device. Case closed. Use Free Software and, now the device is yours. Not so fast... People who want to take away your freedom are not stupid.

Right To Repair

I'm not very interested in Apple. It a company that does hardware with proprietary software on board. But I am a movie lover. And I enjoy a good film by a competent director. One such director is Danny Boyle, director of such classics as Trainspotting, The

Beach, 28 Days Later, Sunshine, 127 Hours and Yesterday.

There is an interesting film, I saw from Danny Boyle. It's has 3 acts. Each one is set in it's own decade. One is about the 80s. Shot on 16mm film, for the grainy, old look. One is set in the 90s and shot on much crisper 35mm film. And the last act is set in 2000s and is shot on digital.

Also each act is a scene of the same, main character preparing to give a public speech about his new product. Each time, the tension is built using the simple, ticking clock device. He has to finish arguing with all the people and get ready before the show begins.

The movie is called Steve Jobs. And I'm not talking about the 2013 Jobs. I'm talking about the 2015 Steve Jobs. With Michael Fassbender and not Ashton Kutcher. It's important. One is a good film. While the other is questionable.

While watching this movie, I learned quite a bit about Apple. And I learned to hate them even more. As I understand it there were two people behind the company. Steve Jobs and Steve Wozniak. And early on Jobs wanted to lock all of the technology down. While Wozniak wanted to give people some level of freedom. With the Apple II people could still open it up and fix it. Jobs had different ideas in mind.

Creating a monopoly on fixing a product is an evil move non the less. But some companies care about their image more than they care about being nice. I heard about some kind of apple credit card which asked the users kindly, not to put the card in leather or jeans. They wanted the card to look new always. This would be a kind of advertisement for the company.

Think about restricting modification to such a degree that all people will have the same exact device. And if somebody buys another of those devices. They will get the same exact features. I think I understand the marketing concept that Jobs was envisioning. But I think the ends, do not justify the means.

Apple started by implementing strange screw designers in the 80s. And now, they have software DRM encryption keys in each component of a device.

That if somebody wants to change a part on their own, the part will refuse to run.

Apple is not the only company that has such a terrible design practices. For example, the infamous John Deere tractors. They created a DRM system on a tractor's computer, preventing people from fixing it. Abusing the DMCA laws (that make it illegal to break DRM) to create a monopoly on fixing their tractors. I would not buy a tractor like this.

"Secure Boot"

With legacy BIOS becoming more and more obsolete and more manufacturers shipping UEFI only boot-loaders, a new problem has arrived to the scene of computers. This time from the side of the Microsoft Corporation.

Microsoft made it's 80% market share not by making good software, but my cleverly doing bureaucracy. They have contracts with a large amount of manufacturers to ship computers with Windows preinstalled. You can probably do something about it, by installing an actually Good operating system, such a variant of GNU / Linux. But manufacturers are trying to make this harder.

In the modern BIOS you can find a setting for a "Secure Boot". It's advertised as a mean of preventing malicious operating systems from running on the hardware. But in reality it's a way to stop people modifying the software. Making it hard to move from what ever is preinstalled.

Few articles back I explained that malware manufacturers (proprietary software companies) do not care whether you know that you are running malware or not. They care about you not being able to stop the malware. Not being able to change the software. So if it will help you to buy the computer, since it has a GNU GPL license on it's operating system. They might force the secure boot, so even if you have the ability to edit the source code, you would not be able to install the modifications.

Unfortunately it's already happening with mobile phones running android. Android is based on the Linux kernel, with a completely new user-space. But the Linux kernel is under the GNU GPL making all android phones run a Free Software kernel. But most of the manufacturers use digital signatures and DRM systems, similar to secure boot, to stop people from changing the software on the device.

This is why the GPLv3 includes another, additional clause to prevent such misconduct. Unfortunately Linus Torvalds doesn't want to cooperate. He wants to keep the Linux kernel on the phones. Since Freedom is not his concern. From the other side, people like Bradley M. Kuhn from Software Freedom Conservancy argues that GPLv2 has enough legal code to prevent this kind of abuse. And people just need to file lawsuits on the phone manufacturers if they don't provide a tutorial of how to sign your own, modified versions of the software.

As with computers and laptops. We can wipe the Windows away and install GNU / Linux. With other hardware, we should be able to do the same. We should be able to by the new Galaxy phone and put Ubuntu Touch, or Mobian, or PureOS on it. It should be possible on the launch day.

Your Turn To Help Free Software!

WE CAN MAKE IT HAPPEN!!! There is a campaign of the Software Freedom Conservancy called Copyleft Compliance Projects. A part of this campaign is to **Report GPL Violations** in devices. If you have an Android phone and you can't find instructions to swap it's operating system, please report those phones to the Software Freedom Conservancy. They provide an email address compliance@sfconservancy.org. Please do not bother them unless you've researched the device ahead of the time. It could be, that to get the instructions, you will need to contact the manufacturer. Only if they do not provide the instructions, we can sue them.

Alternatively, you can look for the desired device in Ubuntu Touch Supported Devices. If it's not there yet, you have a device that you can hunt for GPL violations. The more SFC will know about these device, the more lawsuits they can file, the more freedom we can get back. Who controls your computer? Let's make it so we do. **Happy Hacking!**

Presentation Of Free Software

I think the biggest hurtle for Free Software adoption is the presentation that it's given. There are almost no pretty ads and almost no promotional campaigns. Maybe we can do something about it?

`lbry://@blenderdumbass:f/presentation-of-free-software:8`

Successful business, software or not. Successful software, Free or Proprietary, usually have a good, proper presentation. They have a nice looking website, the software it self is nice looking too. It's easy to install. Easy to get started. Easy to try out.

If you go to Blender.org a website for a Free Software 3D modeling and animation suit. You will see what I'm taking about. Blender has more than 14 million users. About 3.5 million downloads from the main website, every release. It's not counting the other ways of getting Blender. Like Flatpacks, Snaps, Steam, Microsoft Store and more.

When you launch the Blender's website, you will see a big banner image, showcasing work done with Blender. A title, usually outlining the latest news. And a large, blue Download Blender button. If you scroll a bit down, there will be a section of articles about all kinds of Blender related news. And if you scroll even more, the website will show you interactively the features of Blender. And then they will ask you to participate in donations.

All of it is pretty and calls for action. And it's important that it does, so. You don't want your website to say something like "If you would like to try out this software, you can get it on Ubuntu via the following command sudo apt-get install blender". You want it to do this instead "Download Blender NOW!".

COPY-WRITING

There is a term that I hate pronouncing. It's way worse on the confusion ladder, than even Free Software. Copy-writing is this term. The problem with this term, that it seems to be talking about Copyright. The law that restricts people from making copies of certain files. But instead Copy-writing is about writing things that people will understand. Do you copy?

It's an art in promotion using text. If you ever seen a poster, a billboard, anything with text, and designed to sell you something. It's copy-writing. It's usually nicely formatted, gets people hooked right away and has a very easy to follow, call for action. A website with a button to download something. A billboard with "Buy now in *these* stores". Or a movie poster that says "In cinema from *this date*".

You can learn about what works and what not, by observing it yourself. What billboards grab your attention and make you want to buy the things? What movie posters convince you to see the movie? And then, you can digest the piece at hand and reverse engineer how it made you feel the way you feel.

It's important to look at bad examples too. To understand what to avoid.

Free Software Has Presentation Problems

While a lot of good Free Software is introduced to us using a nicely built website. Blender, Ubuntu, LBRY (Odysee) and so on. Unfortunately for the majority of the free software, what you are going to look at is a github repository page. It has a very rudimentary, text explanation about what this software does. It usually has a small step by step tutorial of how to build the software from source. And it's written in such a way that it will most certainly scare away most users.

Other times, we are introduced to Free Software through a Distro specific application installer.

Sometimes it's good enough to feel nice about installing something. Like KDE's Discover. Or the Ubuntu Software Center. But even though they did try to make it look good, it still lacks the panache of some other software centers.

Ultimate GNU/Linux Game Installer

Compare it, perhaps to the proprietary Steam. Or other things like it. It will have a very prettily placed, composed UI. When you click on a game, you get a video preview of it's game-play. A pretty poster, designed by Copy-writers. Everything screams GET IT NOW!!! *And yes, I'm currently reverse engineering Steam*.

Imagine a Free Software Game installer app. Where you could both publish your games and install them. Where you could get payed for your games. Even though they are Free Software.

Imagine, you open the app and it animates with dopamine inducing beauty. Perfectly composed. Pixel perfect. Animated. With all the glory and charm one can expect from a good looking UI. It has a BIG BANNER of the random game of the day. With a video of it's gameplay running in a rectangle. A BIG BUTTON saying something like "GAME OF THE DAY... *name of the game*".

When you scroll down, it animates tiles of other random games. You have a search, somewhere in a corner. But it's not required. It should present you with enough things, to never use a keyboard.

You click on a game and it's rectangle animates, filling up the whole screen. A video of it's gameplay stars. A HUGE install button is there. If you click it, it will install the game. Unless it's payed. Then the button will say "BUY", or something.

Beside the INSTALL / BUY button will be a MESS AROUND button. That will get the source code of the game for you to hack upon. The source code will always be gratis. But will be significantly harder to install.

It will have a Freedom Score. Similar to F-Droid. And based also, partly on the License. A game which is copylefted will get more points. A game with it's assets on a non-commercial license will get less points. I want to implement it in such a way that anybody could post any game they like. And if the game is proprietary. It will have a HUGE RED ALERT. Also it's good to have proprietary games under a setting. Similar to mature tag on the LBRY Desktop.

Possible Implementation

I want to utilize the LBRY protocol for storage. Making a set of conditions. Like special tags. And things like that, that will be recognized by the software as a proper game.

I think a game, should be a whole LBRY channel. With it's binary build as one publication. Source code package as another publication. And so on. Probably even updates could be done similarly. Using a separate publication. Or updating the current LBRY publication.

It will also store a gameplay video. A poster. A logo. A manual, if needed. And other things.

This is no where near final stages of the concept. I WANT TO MAKE THIS APP. With your help. And this is where the next part of this article comes in.

Your Turn To Help Free Software!

On the Matrix chat for Blender Dumbass (#BlenderDumbassChat:matrix.org), we already started discussing, with the members, the potential implementation. But we need to figure out everything. And for this a collaboration of multiple people is needed.

I'm asking you to JOIN THE CHAT now and help us figure that out. Make mocaps. Figure out the best ways to build it. To ultimately have the best Free Software game installer possible. And with it help the Free Software.

Happy Hacking!

Free / Libre Alternative to Steam

How about starting the quest of making better presentation with a Game Launcher?

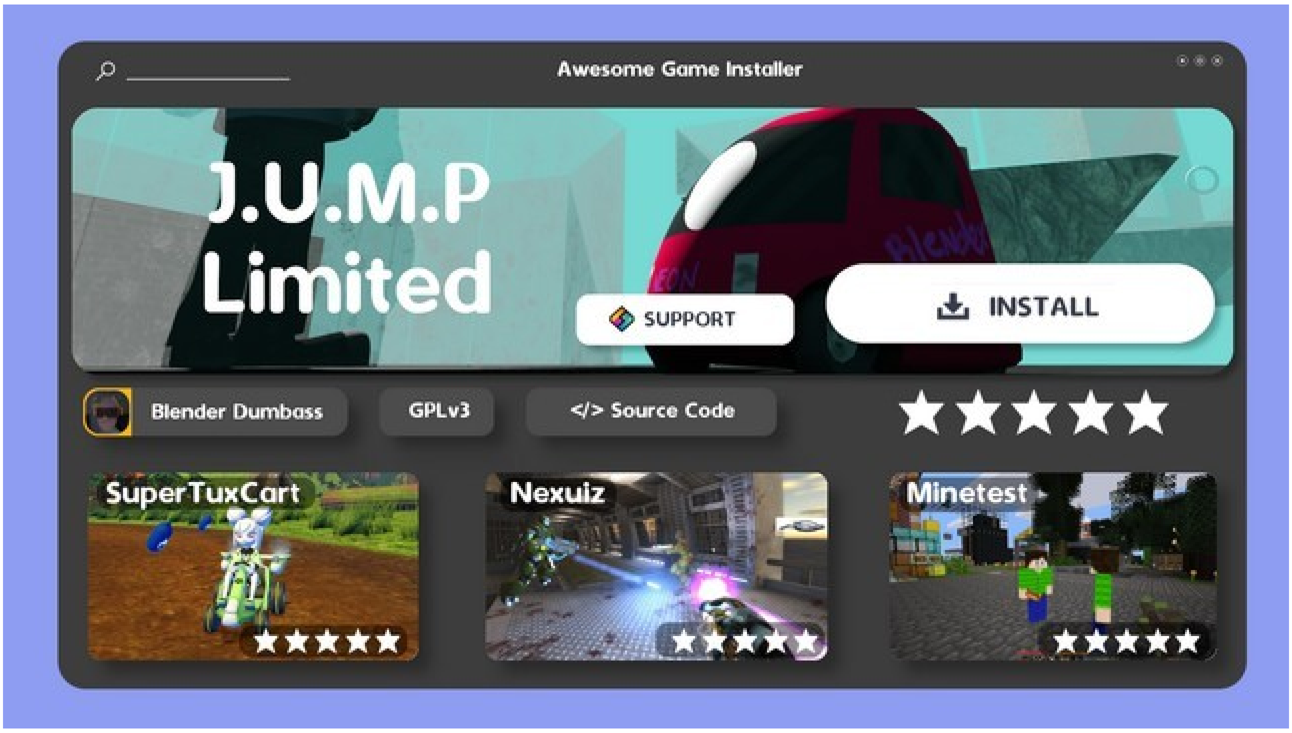
lbry://@blenderdumbass:f/Free-Libre-Alternative-To-Steam:f

In a previous article titled "Presentation Of Free Software" I described a problem that a lot of Free Software projects have. I outlined a possible idea to solve it. And while some people criticized me, others came to the Matrix chat with me (link in the end of the article) and brainstormed the piece of software.

Just to be clear. It's not built yet. Nothing of a working prototype is finished. It's just an outline of a design.

Actually I feel weird. Since I have two projects running simultaneously right now. One of them is this software project. And the other one is my movie, Moria's Race. I feel like when dealing with one, I'm procrastinating from another. It's a strange feeling.

The UI



This is the UI of the game app that we are making. It's not a functional app yet. So don't get too excited. It's just a preview of how I imagine it.

When you launch it, it will select a game... Perhaps randomly, or using an algorithm running locally. It will present the game in the top banner thingy. Basically a selected game is going to render as the top banner thingy. And all the other ones are just suggestions.

As you scroll down, the suggestions will load. And populate the scene. I want all of it to be animated.

So nothing will look boring. I want this app to be dopamine inducing AF, so to speak.

Maybe even it could do satisfying sounds. This would be even cooler.

Protocol

In the previous article I described using an LBRY protocol as the hosting platform. I changed the plan to a little bit more polished.

So each Game will be a little text publication in the LBRY protocol. It means, you can publish a game to this app using Odysee. If you know what you are doing.

The text publication, probably will be a small JSON database. It will include things like title and description. And will have links to things like logos, gameplay video, installation scripts and source code repositories.

For something like my game, J.U.M.P Limited It might do a file like this:

```
{
    "title":"J.U.M.P Limited",
    "description":"A game where jumps are limited.",
    "binary":"lbry://@blenderdumbass:f/jumplimitedthegame:0",
    "install":{
        "ubuntu":"https://example.com/install-script-ubunutu.sh",
        "arch":"https://example.com/install-script-arch.sh",
        "windows":"https://example.com/install-script-windows.bat"
              },
    "repository":"https://example.com/git/jumplimited",
    "gameplay":"lbry://@OfficialZaney:8/jump-limited-playing-a-game-made-in:b",
    "logo":"lbry://@example_channel:0/example_logo:0",
    "license-software":"gpl3-or-later",
    "license-art":"cc-by-sa"
}
```

Look at the example very closely. Since it's important. We gonna give a person a way to provide a script to install the program. It may only be one line of bash script, as in sudo apt-get install game_name. It could be something complex as running a very elaborate set of commands. And downloading files and what not. We will give a user an ability to review the bash scrips before they run them.

The "binary" data point will be used to link to an installable package, which will be downloaded first. It might as well have a tree, like the scripts. To include binaries for different operating systems. It could be used to make the game payed. Using a payed LBRY publication to store the binary.

Also, since it's a game, we will include a dual license. A lot of Free Software games use one license for the code and another license for the artwork. Games with a non-commercial license on the artwork, will still be recommended, but will have a little yellow warning sign beside the art license.

About the software license. If it's copylefted, it's gonna have a green sign of approval. If it's not copylefted, using MIT or Apache 2.0 license. It gonna be a neutral color. And if it's an Open Source license that is not a Free Software license. Like the Ethical licenses. That prevent people from doing certain things. Or something like the license of the Unreal Engine. It will have a blinking yellow light. Signifying a huge problem.

Also these games will not be recommended by default. And installing them will require pressing something else first. Making it more of a hassle. To make those games even appear in the search or the recommendation, you will have to go into the setting and enable "Non-Free Readable Source Games".

Proprietary Games

In the same settings, there will be a button that you can press to "Remove, protection from proprietary games." Which will stop filtering the results based on license. And will present you with everything.

Proprietary games will show up with a red alert. It will be blinking and making a scary noise. So technically, you could publish a proprietary game on this platform. But it will be not recommended.

Hosting?

If you want to publish a game and have no hosting servers. You can use the LBRY protocol. I think, if we make a publishing feature. It could upload every file directly to LBRY.

I want to support as many file protocols as possible. But we can start with LBRY and HTTPS. Since they are the easiest to implement. So you could link both HTTPS links and LBRY links.

Your Turn To Help Free Software!

Games are very crucial. Since most people play games. And that's what they buy a computer for.

While Valve is bringing some people to a little more freedom. By making proprietary games run on GNU / Linux smoother and smoother. This is not fixing the issue of the proprietariness.

I want to make this app, because I believe it could actually be something that will fix the issue. Free Software could be sold. And the easiest way to sell it is, to have your binary payed.

For payed games, we can make a payed publication on the LBRY protocol. It will be a pack of the game that will be unpacked and installed. As you can see in the "binary" section of the protocol piece.

It's gonna be significantly simpler to install a game by sending some LBC and hitting an install button, than by hitting the source code button, cloning the repository and compiling something from source. Even more than that. If we are not using dollars, but using LBC. It will be even easier to press the buy button. It's proven by research that people are more likely to spend a token, than to spend the "real money".

If this app will succeed in making people buy Free Software games, it might open a real market. And hopefully, will convince larger and larger studios to publish their games to this app. Of course, since the app will prefer Free Software games more than proprietary. It will ultimately force them to release their big games as Free Software too. To be discovered and recommended.

In this design, there are many flaws. I still don't know how to implement it. In what language? Is it going to be Electron? Or something else? Is it going to be built using a game engine? What are exact specifications for the protocol system? How exactly it's going to look?

All those things are important to figure out ahead of the time. To set the foundation. And this is what we are doing, right now, at the Blender Dumbass Chat on Matrix.

So JOIN THE CHAT NOW and help us make it real! (link in the bottom of the article)

Happy Hacking!

You Should Not Be Okay With Annoyances!

A lot of things could be fixable if two things were there. One the strength to make a change and two, the ability to make a change. With Free Software you have this ability.

lbry://@blenderdumbass:f/You-should-not-be-okay-with-annoyances:9

When it comes to Software Freedom, a lot of people use arguments like "It doesn't spy on you" and similar things to promote Free Software. And while it's a valid argument, most of the time. The real reason is quite deeper. Not so long ago, a Matrix Chat user by the nick of Troler installed my software VCStudio, to hack on it. He didn't like the speed of the scrolling in the text-view window. So he opened the source code and edited the speed to be precisely as he wished it to be. He was annoyed by it at first. And was not okay with being annoyed by it. So he fixed it.

Stockholm Syndrome

In 1973 bank robbers took hostage 3 people. They kept them hostage for 6 days in a vault of the bank. When released, those 3 people didn't want to testify against the robbers. Instead, they began raising money to defend the robbers. It happened in Stockholm, Sweden. Coining the term Stockholm Syndrome.

Stockholm Syndrome is a condition in which hostages develop a psychological bond with their captors during captivity. And it could go beyond such situations. It's arguable that people develop similar bonds with Proprietary Software. While it keeps them in captivity, taking away their freedoms. It gives them enough surface level dopamine, to develop a bond like this.

People that defend malicious software, that clearly makes everybody uncomfortable, arguably have Stockholm Syndrome. In the Matrix Chat for Blender Dumbass (link at the bottom of the article) one of the users shared with us a comment, of one such victim.

He was answering to an argument of not installing Windows to the Steam deck with words:

> I agree with your reasoning, but all the more reason that Windows Central should not promote this product. It's intended to run Windows Games (some have said, "They're not Windows Games they're PC games!", but according to Valve and Proton, they're Windows Games: "Proton is a new tool released by Valve Software that has been integrated with Steam Play to make playing Windows games on Linux as simple as hitting

> the Play button within Steam." from https://www.protondb.com/). But it doesn't pay a license fee to Microsoft. It provides a broken experience (though I agree with you that via Linux they can more easily customize it for the hardware). This is technically legal, but ethically an anti-MS pirate device. It harvests Microsoft's extensive investments in WINDOWS gaming and steals that value from Steam users. At a time when MS has recently been trying so hard to play nice with everyone, including Valve by putting MS Studio games on Steam, this is a product based on Gabe's personal spite against MS. I hope MS removes its games from Steam and adopts a war footing. Whether they do it or not, Valve is treating this as a war and is on the offensive.

I can go on forever deconstructing and debunking claims in this comment. From that I don't see Microsoft trying to play nice, as proven by this page, this page and the recent Copilot drama. That equating people who want to copy files to people that attack ships (pirates) is more then unreasonable. That the 4 Freedoms are Essential and Windows doesn't give them to users. That nobody should be paying a license fee for merely using a file format or a software API as proven by the Google vs Oracle case.

But the more shocking to me was the fact that this person was defending Microsoft to begin with. While almost everybody unanimously agree that they hate Windows and they hate Microsoft. To a point that it's an official motivation to write Free Software. People use it only for one reason. And it's because it works. They are okay with suffering this operating system for a convenience of being able to run some software.

Then comes this person and argues about Microsoft as if it's his own child. At first I assumed that he was a Microsoft employee. That would've explained a lot. But later, the chat confirmed that he wasn't. And the next best theory was, that he had developed Stockholm Syndrome toward Microsoft.

Switch Now!

For the last week or so, as I was communicating via the Matrix Chats. I was looking for the best Matrix client. I stopped at Spectral. It looks and feels like Telegram. Reminding me of this old article. But I looked through a lot of clients. From Quaternion, Fractal, FluffyChat and NeoChat to even Element. Non of them makes me feel as nice as with Spectral. And while Spectral has it's own little problems that I might get into fixing. It's nice.

I like that in Free Software you can always find a different thing. Different app for the same service. Different Desktop Environment. Different fork of the software. Different Disto. If you are annoyed with one application, you most likely can switch to a different, less annoying application.

With Proprietary Software, it's not as easy. There could be alternatives. But they are substitutes. Moving from WhatsApp to Matrix will require way harder work, than moving from Spectral to Fractal. But... Non the less, it is possible.

And people can switch from their proprietary software, to software that they can edit themselves.

To software that respects them as users. To software that is often not malicious.

You can switch from Photoshop to Gimp or Krita, from MS Office to Libre Office, from Maya to Blender, from Minecraft to Minetest, from Windows to GNU / Linux. You can do that. So how about doing it?

File a Bug Report!

I use Odysee a lot. I write those article. And Odysee is Free Software. I encounter bugs a lot with Odysee. Other people will be fine with having bugs. Saying that "It's a startup." or "I don't mind it". But I'm usually not fine with having bugs.

Most Free Software projects, for the sake of convenience host their source code at a Git Repository. Usually using something like GitHub, that has a lot of problems. But non the less a repository where you can file a bug report. *(If you develop Software I would recommend instead, using notabug.org)*

This is a bug tracker for Odysee and LBRY Dektop UI. You can see that people are actively posting issues that they find with the software. From bug reports to

Feature requests. Anything that annoys them, they speak out about. And the developers / contributors have access to those issues, to know what to fix / implement next.

So if you have a broken feature, a bug, a missing feature. An issue of any kind with the software you are using, file a bug report. Go to a Git Repository for that project. Sign up, it takes a few seconds. And write a little text article, explaining what you are annoyed with.

Fork it!

Coming back to the story that I outlined in the beginning of this article. About a user editing my program, to make his user-experience smoother. He found the file where the feature was implemented. He figured out the syntax. And he added a value that speeds up the scrolling in the text window. It took him less then 5 minutes to do so.

Coding is not particularly hard. You just have to remember that the computer likes things to be precise. Sometimes finding the feature, is harder then editing it. But with good software code, the developer provides enough comments, for you to make a simple search. To find things you are looking for.

In the Documentation of VCStudio (here is an example) I have a Source Code section. Asking people not to be afraid to customize the software. And explain roughly for what files to look and what variables to change, to do various modifications.

With other software, you may need to ask for help. Or dig a bit longer through the code yourself. But it is doable. And with enough stubbornness, you can remove any annoyance what so ever from your computer.

Your Turn To Help Free Software!

This is a part where I ask people to help Free Software. I want you to do a couple of things.

- Switch to a Free Software program, from proprietary. It may be only one program. You can go one step a time. Chose one program that you are fine with switching. I recommend it to be a chat application, since it will force your friends to switch as well.

- File a bug report or an issue for a Free Software program that you are using. Let the developers know that you have an issue.
- Edit one thing in a Free Software program. Try something that doesn't need compiling. Something built on Python or similar languages. You can hack on my program VCStudio. If you make a modification that is actually good. Don't hesitate to make a pull request.

You should not be okay with annoyances. And with software they are easiest to fix.

Happy Hacking!

Simple People Like Nice Things
August 2nd, 2021

Lot of us use complex reasons for why we use Free Software, but as it turns out, most people don't care about privacy, malware and other nasty stuff. Instead they may be only interested in the good stuff we have to offer.

lbry://@blenderdumbass:f/Simple-People-Like-Nice-Things:1

For the last month or so I had drawn a Free Software promotion poster everyday. I used a piece of paper and a marker and glued that to the wall at a store where I work during the day. It's not a geeky store where customers are computer nerds and hackers. It's not a place full of reporters and free speech activists. It's a place with mostly old ladies and their husbands.

When I draw a poster with the words "Free Yourself", they see it as nothing at all. When I draw a poster talking about malware, they either do not notice, or think it's some kind of a conspiracy theory, that has nothing to do with reality. I could go on and explain things. And this works, if you have time. But I notice one pattern that keeps them reading the whole poster and even asking question. It's a poster where I tell them about the 4 Essential Freedoms.

A good looking re-design of such a poster, would looks something like this:

4 ESSENTIAL FREEDOMS OF SOFTWARE

0 To run the software when ever you wish & for what ever purpose.

1 To study the source code & make modifications to the software.

2 To give or sell copies of the software to other people.

3 To give or sell copies of your modified versions of the software.

You have the 4 essential freedoms with other useful items that belong to you. Clothing, Food, Simple Electrical Devices. But most software companies do not want you to have these essential freedoms with software, running on your various devices. Taking away your control over your own devices.

SWITCH INSTEAD TO FREE SOFTWARE!

www.GNU.org

It's only a correlation, but it seems like in Free Software, it's easier to argue about it's benefits than to argue about the hazards of proprietary software. At least to start the conversation.

It does sound a lot like a conspiracy theory.

To prove that a propriety program doesn't do something malicious is almost as hard as to prove that it does. Experts that know where to look and what signals to expect, could point us in the right direction. Sometimes this might be such a good prove, that it's hard to overlook. But most of the time, there is still doubt.

With Free Software, the source code is available to read. So people could verify everything precisely. Removing any doubts about any feature anywhere in the program.

But when you are talking about this to a person to whom software is literally magic, you sound like one of those tin foil hat conspiracy theorists. You have the same exact trope of claiming that something bad is happening in secret, and there is no way to know for sure.

This is one of the reasons why most people do not care about their privacy. Especially grannies, that might believe Facebook Messenger to be more secure than Matrix. Since Facebook is a bigger company, which means they know better.

For them, trying to explain the Free Software movement from the side of surveillance, sabotage and other malware. Is not going to be productive. They will counter-claim everything. Say that you know nothing. Call Free Software a sect. Do anything, but avoid caring about this "surveillance" thing.

4 Essential Freedoms sell Free Software better.

When a simple person that knows what a computer is and what applications are, but doesn't know how it's made or what is malware, reads about the 4 essential freedoms, he gets curious. Most people experience a form of annoyance with the software they use. Either a button is in the wrong place. A restriction is too ridiculous. Or other annoyance that might make them mad. When they read about the 4 essential freedoms, they get exited. Since this will solve all their issues.

When I introduce them to the problem of having control over the computer they use, rather then being private, it tends to fire their imaginations. I can ask them to imagine what modification would they do to the software. And tell them that with Free Software this is all possible.

This is a dream come true. Instead of talking about preventing an Apocalypse. I tell them about building a Utopia. A Free World where they decide what goes into the functions of their software. Where they decide how the software they use will look and feel.

Your Turn To Help Free Software!

The poster above, with the four essential freedoms illustrated easily, is available to download. In this part of the article I encourage readers to help Free Software. And we have done some good stuff already.

Let's break up my today's request into stages. Stage one: Get the poster in a format that will be preferred to you. In this folder you can get the following files.

https://notabug.org/jyamihud/FreeSoftwareActivism/src/master/colorful-4-freedoms

- English poster PNG image.
- English poster PDF document.
- English poster Source. ODG (Editable with LibreOffice)
- Russian poster PNG image.
- Russian poster PDF document.
- Russian poster Source. ODG (Editable with LibreOffice)

I provided the sources for a very simple reason. So you could translate it to the language of your country. This is going to be the step two: Translate it to the language of your country. If you want to help. Upload your version, and link it down below. So people could use your translation.

If you don't have a color printer, there are services in many cities that print in color for a very small fee. I used those myself. It's usually an office appliances store, where they sell printers as well. This is the step 3: Print the papers, in what ever size. In what ever amount.

And the step 4 is: Spread them. Glue them in a visible place on your work. Put them in any place that you can find, where people can read it. Hell, make a sticker with it and glue it to your own laptop. Be creative and don't be afraid.

Remember, you have the Freedom of Speech. And thus you can spread these posters around. Anybody who tells you otherwise, don't listen to them. *Be careful in non-democratic countries.*

Happy Hacking!

I've Got an Issue With Odysee! (let's fix it)

Odysee is great and all, but sometimes comes a point when you find an annoyance that the main developers are not welcome to help you fix. This article will show you how fixes are actually done if Free Software.

`lbry://@blenderdumbass:f/Ive-Got-Issues-With-Odysee:6`

Any Free Software project is worth it. It adds value to the side of Freedom. But sometimes people do questionable things. And this is why Software being Free is important. So you don't have to agree with the questionable things. And use the software in some other way.

Odysee lately done some questionable things. But it's Free Software. The code that draws Odysee is under the MIT license. And there is an alternative LBRY Desktop app that I'm using to publish this article. It doesn't have notifications though. Which is weird. So I might not be able to read your comments.

The main issue with WEB-APPLICATIONS

A lot of people want things to be up to date all the time. It's usually a good advice, since security is important and crackers find new ways to crack things all the time. And only the newest patches can prevent some of the recent cracks. But on the other hand a security advice would be ***not*** to update Audacity. Since the main branch of Audacity had recently implemented malware.

Think about how much more of an issue would this be if Audacity was a Web-Application. Or a website. If every time you run it, it's automatically up to date. And where you are never able to load the previous version if you so desire.

Websites used to be documents with information. Linking to other documents with information. Software was something you would install on a computer locally. It might have some ability to talk to the internet. Like if this software is a web-browser or a torrent tracker.

Something similar you can observe today with telephone applications. Even though I dislike Facebook or Twitter, on the phone it's a program installed locally. Not a website that's loading new code every time it runs. Unless of-course you are using the progressive web-apps. Which is a big mistake in my opinion. It doesn't excuse Facebook or Twitter since their apps are proprietary and are filled with malware anyway. But the idea of installing an application that you may or may not update is good.

There is a whole issue about back-doors with software. And ability for somebody to execute remotely commands on your computer. It's a real thing and it's easy to implement. Every time an app like this runs, it connects to a server online. And if the person on the other side of this connection sends a message to the program, the program is probably already configured to do something.

An application that you are using for instant messaging, for example, has many such back doors. Every time anybody send you a message, your device without your command to do so, automatically sends you a notification. This is quite harmless in this particular example. But think about all the malicious ways, a programmer can exploit this ability.

A universal back-door is even worse. It's an ability for a programmer to impose any change to the software remotely. Making it so next time you run the program, it does something different. Think about an automatic update feature. But where the programmer could sneak anything at all in the, so called "Update".

A web-application is an application with the universal back door. Even if it's Free Software, you cannot ask the server to give you the previous version of the website. It's not present on the server. Unless of-course you are using the Wayback Machine. And even then, most of the functions will break.

What happened with me and Odysee?

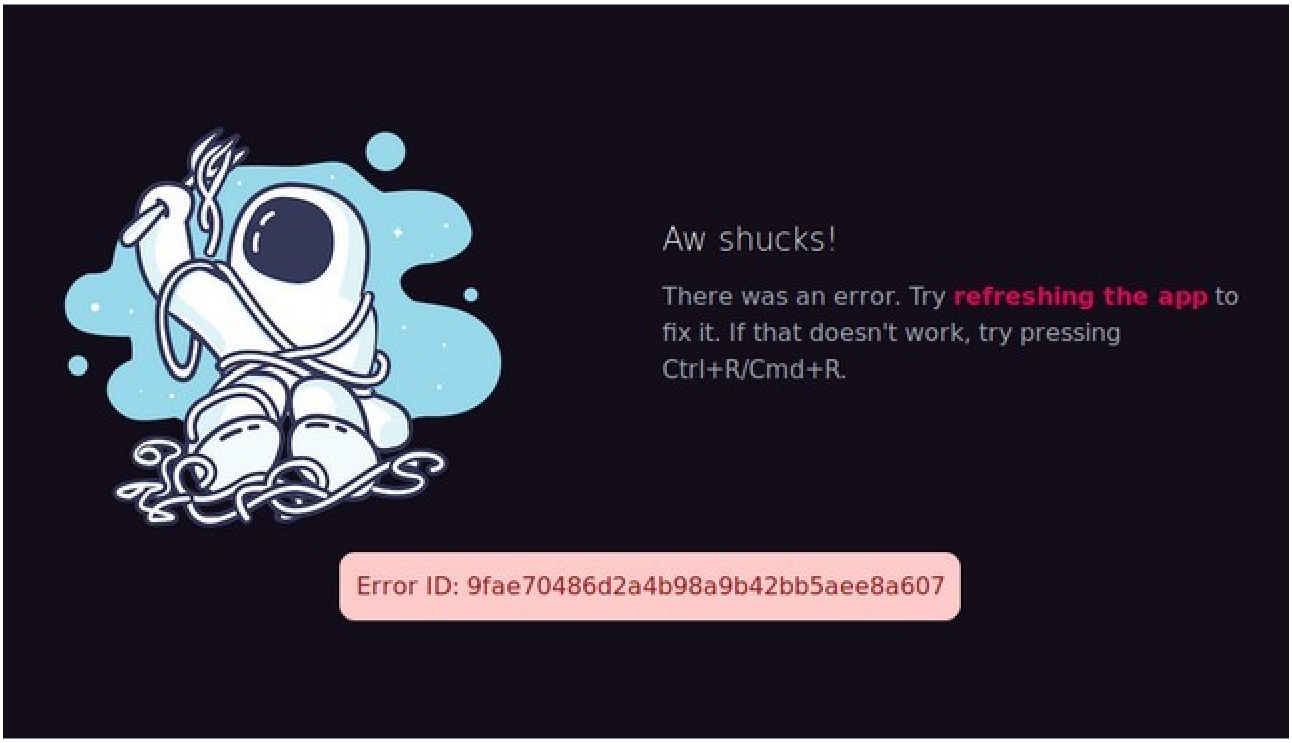
I'm using the GNU IceCat browser as the main browser. And I was using Odysee.com to upload these articles. As you may know, IceCat is weird. It takes your Freedom seriously. And basically blocks the web, so to speak. So this is why I use GNU IceCat.

Odysee.com uses a lot of very poorly designed libraries to draw it's UI. And they were trying to optimize the responsiveness of the UI, while breaking some support for browsers like GNU IceCat.

I do still have Brave installed. So it's not like I'm out of options. It's just doesn't feel nice to be forced to use something else. I want my freedom. And I want to use what I want to use.

I started noticing bugs with the UI on IceCat when they added the playlists support. I reported the bug.. Nothing changed. The bug is still there. Though I did receive a 100 LBC support from one of the developers. Probably they liked my deep analysis of the situation.

Later I noticed a similar bug when pressing the "Copy RSS Feed" for various channels. Again it happens only in IceCat. And finally this bug started happening when I want to load a page of any publication. I can browse the home page. I can browse the channel's pages. But I cannot load any of the publications. It just gives me a bug screen and that's it.



Aw shucks!

There was an error. Try **refreshing the app** to fix it. If that doesn't work, try pressing Ctrl+R/Cmd+R.

Error ID: 9fae70486d2a4b98a9b42bb5aee8a607

I reported this as well. and the answer was as follows:

> Please send us an email at hello@lbry.com with a screenshot of the console tab (press ctrl-shift-I, then console tab).

Okay, so I did the Ctrl-Shift-I thing and mailed it to the mentioned email. This is the error message in the console:

```
columnNumber: 63175
fileName: "https://odysee.com/public/ui-4bcb1f34-4ed9-48f7-af02-55b735e23668.js"
lineNumber: 1
message: "Object.fromEntries is not a function"
```

You can see from this error it self, that the developers of Odysee are lazy asses. Column 63 thousand on the line first? Are you kidding me? The JavaScript source of the page is insane. If you actually load the url of the script, you will see what I mean. It's a compiled mess of characters. There is nothing remotely resembling sanity. Basically, it wasn't written by humans.

The email that I got back from them was as follows:

> Hello, sorry about that, have you tried refreshing the page? If you did and it didn't work, please try with a different browser.

I already did try with a different browser. I wrote it in a bug report earlier. But maybe this person is not familiar with the bug report. So I sent them another email clarifying the situation. They responded with:

> We addes some optimized javascript functions that may not be supported on all browsers, especially older ones. We'll see if we can work around it, reported to the team.

So okay. They are now even more familiar with that I have an issue. This issue is not fixed yet. The code base is still unreadable. The source is still confusing as hell. The problems are not gone.

What I can do?

Currently I'm using the LBRY Desktop. But unfortunately Odysee and LBRY desktop UI is built from the same source. Meaning that I'm technically running a tiny browser with a bloated website on it. I don't like it.

One person on the Blender Dumbass Matrix Chat suggested using an LBT application. Which is a kind of, terminal based client for the LBRY network. But it's too simple in my opinion. And it doesn't have an option to upload.

Alternatively I can try hacking on the LBRY SDK which does have all the functions. I can manage everything from the SDK, but I will need to develop an application to do so.

Your Turn To Help Free Software!

What I gonna do is this. I gonna go and read the SDK manual. Then I gonna build 3 basic clients for LBRY.

The first client will be terminal based. It will support uploading, downloading, commenting and replying. As well as reading comments. And seeing the Wallet History. It's not supposed to take too long to make it.

The second one will have a graphical user interface. But will not resemble youtube in anyway. I want it to resemble a torrent tracker with an upload and download. I want to use a very basic GTK to built it, so not to introduce any bloat.

Third one will be a simple web-client installable as a server. That will use HTML5 and as little JavaScript as possible. Even better than that. If I can make it not use any JavaScript at all, it will be dope. It will be light to load. Will pass the LibreJS test. And will work with JavaScript disabled.

I just opened a Matrix Chat (#FastLBRY:matrix.org) for this project. Who ever wants to help, can join the group. I will setup repositories when I will see how it could be achieved. Maybe I will load it onto somebody else. Like the development of FreeGILE is now on Troler. Since I have my movie project and other stuff.

Anyways this has to be done by somebody. So please join and offer your help.

Happy Hacking!

Malware that's Advertised To Help You

Sometimes there are things that are designed to make you harm. But they are advertised to you in such a way, that you will think that it’s design for your good.

lbry://@blenderdumbass:f/Malware-thats-advertised-to-help-you:6

There is an abundance of hardware and software that's meant to do various nasty things to you, but is advertised as something good, not calling to attention it's nastiness. A good example of this, are those spy apps parents install on their kid's phones. They literally bypass any law that prohibits collection of personal data from kids. Since parents themselves install it and agree to the malicious terms.

Then the program collects anything it can from the phone of the poor child. I've seen some advertisements claiming to give parents the whole browsing history and all the messages. Not counting the GPS location, ability to take a picture from the kid's phone's camera and collect other very sensitive personal information.

All this data is sent to the data brokers at the company who developed the program first. They sell the data as usual. Making profit off of the child. Then depending on the tier that the parent selected, they will send a fraction of this data back to the parent.

It's wrong to begin with if parents are doing it to their children. Teaching those children about software freedom confuses them. Since parents ultimately have control over their own lives. And they think it's fine. Those parents have a strong stalking tendency. A desire to control the child. They are usually afraid of talking about software freedom, since it will teach the child to remove the malicious app.

But the presentation of the app doesn't go into showcasing the malware too. The developer is not required to put on the main poster, or a description to the app, all the negative things it does. They are only required to put it into a license agreement that nobody reads anyway. They advertise the power that parents will have over their kids. And happy monster-parents write positive reviews to this app. Making the problem even bigger.

When privacy is trending, companies like privacy.

There was a huge case of Apple fighting with Facebook's surveillance to "protect it's users". But if you look at the recent events, Apple is not caring about it at all. There is a whole documents on the apple's site about their latest iCloud spyware.

They start their document with a soothing set of arguments:

> CSAM Detection enables Apple to accurately identify and report iCloud users who store known Child Sexual Abuse Material (CSAM) in their iCloud Photos accounts.

I mean... Right? This is selling the spyware pretty well. Here is a boogie man. He is scary. We want to find him. Thus, all of you will be under constant surveillance. They could change the boogie man to terrorist and the message will be the same. They are implementing malware to spy on files stored on iCloud to track people better. And if they dislike those people, they gonna report them. Only, saying it straight will scare the users. So they make a pretty presentation, with an argument that is hard to argue against.

Privacy is not trending right now. It was trending a few years ago. But then COVID-19 happened. And the new boogie man was anyone with the virus. As soon as they find how to scare you and make you believe that your privacy is less important than the surveillance. They will implement the surveillance.

Unless of course they are Free Software. In which, well... You know what happened to Audacity.

With proprietary software there is no security against the proprietor. It sounds cliche. And in the same time it sounds like a tin foil hat conspiracy theory. But it's true. Apple develops proprietary software on proprietary hardware you are not allowed to fix. They have power over you from all directions. They implement back doors. And store your data on iCloud's servers so to be able to know whether they like you or not.

If they say that they like privacy and fight with Facebook. If they show you how much terrible things TikTok does. It's only because privacy is trending right now. Because this will convince you to buy. Not because they really care.

Rav Kav

I was thinking about writing about Rav Kav for a long time. But couldn't get myself into finding a long enough narrative. So instead I think I gonna sneak it into here.

For a long time in Israel you had a choice when ridding a bus. Do you want to pay cash, or do you want pay with a special bus card controlled by the government, called Rav Kav? It was already malicious. Since it had your identity data on it. So if you use the card, they know that you purchased a ticket to a specific bus at a specific time, in a specific area.

The card gave people a little discount. So a lot of people were convinced that this card "worth it" and used it all the time. I was paying cash. Until one day the cash was no longer excepted in the bus.

For a while I was ridding a bicycle when ever I needed to go somewhere. This card thing is too ridiculous. I didn't want to be spied on when going from one place to another. Later I didn't have a bicycle and was stuck in a remote location. Which meant that I needed to take the bus.

I found out a partial solution. In some stores where you can put money into Rav Kav, you can buy so called "Temporary Rav Kav". It's not signed with your name. Doesn't give you discounts. But it's anonymous. You fill it up with cash in a store. And it's transferred to the card's number. Which is not associated with you.

Also the card is very cheap. So if you are afraid of government figuring out who uses this specific anonymous card. You can change them often. So if you use a bus in Israel. I recommend you, for now to use the anonymous cards. And switch them ones in a while.

Rav Kav was malware presented as a solution. But ultimately was malware. And while people thought about it as only the solution. They were convinced to use it. And then those who were left were coerced to use it against their will. This is not good. I don't like it. I gonna fight with it. If anonymous Rav Kavs will not be supported tomorrow, I will walk.

Not all innovations are good.

Richard Stallman says quite often:

> Democracy was ones an innovation. Tyranny also was ones an innovation.

When people ask Richard and other Free Software advocates about innovation and how Free Software apparently discourages people from innovating, since the monetary incentive is hard with Free Software. The question can be split into three parts.

You can sell Free Software. Free Software doesn't mean software gratis. But means that software respect the 4 Essential Freedoms of the user.

A lot of innovation happened in Free Software. You can list them a good amount of protocols developed for Free Software like Tor, Matrix, LBRY and ActivityPub.

Not all innovations are good. Which is a way more complex issue, but one that needs to be explained.

When it comes to proprietary software, the innovator thinks in terms of "What is something that I can sell to the users.". And I mean sell in terms of convince the user to use it. If he succeeds, he can monetize the data or maybe sell the software for money. Or both.

For example:

- Apps that spy on your child.

- Facebook.
- YouTube music, iTunes, Spotify.
- iCloud malware.

Those are all innovations. There were designed with one goal in mind. And it's to maximize profit. They do not care whether the innovation is good. They care only if they can find a good enough reason to convince people to use it.

Look on the other hand into Free Software. Where anybody can scrutinize your code. And if they disagree with you, they fork you and leave you alone. In such environment every decision that you make should always be a good one. Or you will be forked and forgotten.

This is what they are talking about when they argue about Free Software being bad for innovation. Not all innovations could happen if you need to respect the user.

Your Turn To Help Free Software!

In this section I ask people to help software freedom. And we did a lot of good things already. Today's request will be a little different.

It's possible to make privacy trending so companies like Apple will suddenly start promoting it. I want to go a step beyond.

I think we could potentially do a change if we make Freedom trending. Freedom as in 4 essential freedoms of software. Which make things more private as well. But which ensure that somebody like Apple doesn't just come back to surveillance as they want.

If enough people will resist software that's not free or hardware that's not fixable. We can lower their share value enough to notice us. And then maybe even make changes in their business model. Maybe Apple will ship their products without custom screws and DRM on each part. Maybe Microsoft will release windows as Free Software.

So what I will ask from you to do is this. Make your personal effort to resist installing ***new*** proprietary software. And also ones every week or so, make an effort to migrate ***one*** proprietary program to Free Software.

If you are on android. You can use F-Droid to list applications that are exclusively free software. You can type there words like YouTube and Facebook, to find a free software client for those proprietary platforms.

For other software, if you are not sure, look it up online. If it has a Wikipedia page. It usually says what license it's on. If you can't find any information about it's license. It's probably proprietary, so stay away from it.

Happy Hacking!

Imagine a World Without Cloud Computing

There is no cloud. Only other people's computers.

lbry://@blenderdumbass:f/Imagine-a-world-without-cloud-computing:2

> Cloud - a grey or white mass in the sky, made up of very small floating drops of water.

(Cambridge Dictionary).

There is no such thing as a cloud in computing. There are only other people's computers. Those who believe that their files are stored in a magical cloud, their thinking is cloudy. And while "cloud" storage is relatively harmless. It's like to save money in a bank. Or ask your friend to hold something for you. "Cloud" computing is an issue of an entirely different proportion.

When I was a kid, I imagined internet as a kind of magical substance of data floating around us in the air. I imagined a file on the internet being thrown back and forth by all the computers connected to it. And so this is how I could arbitrarily, from any point, just download that file.

This is possible, more or less. With protocols like torrent and LBRY it's doable. And this article is currently stored on multiple computers at ones. And when you load it, it downloads the article from one of those computers. But the reality of the internet is much less magical. There is no cloud. There are only other people's computers.

A Server

Long time ago computer hackers (before hacking was associated with breaking security) managed to hack on something, to make an interesting new type of technology. They took two computers that were apart from each other. And wired them to a normal telephone wire. When they would make a call, instead of sending sound waves, the wire would send information in binary form. The other computer would receive this information and depending on the program, it would do something with this information.

They managed to make two computers talk to each other over the phone. Inventing Internet.

After a while, standard protocols were developed to make communications like this possible. And more and more computers were connected to the world wide web.

Currently to load a page from the internet, you need this page to exist somewhere. It's not floating in a cloud. It's on a computer, at a storage facility somewhere. Those computers, that serve you those files over the telephone line, are called servers.

A server is basically a beefy computer with a lot of storage and a fast as hell processor. So it can take a load of multiple hundred people connecting to it at ones. Sometimes a server could be your basic laptop. But those websites usually don't get as much visitors. So they can afford such a low powered server.

For things like Torrent and LBRY there are levels of servers. Torrent has servers that track and record what users have what files. LBRY on the other hand uses a block-chain technology to record that data. The files them selves are stored on computers of people who downloaded them. And maybe a few actually beefy servers, so the protocol will work a bit faster. Like the LBRY / Odysee servers.

"Cloud" Storage

Some people are using other people's computers to store their data. They call it "Cloud" storage. Trying to convince the poor people, that it's some kind of magic. In reality though, the files are stored on a server. If you are using Google Drive, Dropbox, iCloud or something silly as Could or Mail.ru. Your files. Photos that you store there. Photos with your face. Faces of your family members. Are uploaded to a computer in a remote location that you don't have a direct access too. Somebody else has.

Don't get me wrong. It's their computer. They control it. And it's fine. The problem is... It's your files. Yes, they give you means to see them and download them back. But what is happening on the server side, you don't know. People who control it could be messing around with your files. Selling this data to data brokers. Making copies of it for them selves. Laughing at your family pictures. Or ever parsing the files, to see who they don't like, to report you to authorities. Like with the recent iCloud spyware.

You may delete the file from the cloud. But is it actually gone? Maybe you pressing "delete" just switches a setting on a server, not to show you that file again. But keeps the file on the server, for some, probably nasty, reason.

A good advice when using server storage services like this is to encrypt every file with a very strong encryption first. So even though they will have the encrypted file, they will not be able to de-crypt it. Do not encrypt it with their tools. Do not use "Cloud" encryption. When you send them files, they should be already encrypted.

Do not use sync on your mobile phones. It sends files un-encrypted to a storage server. Which allows people you don't know to see everything that happens on your device, pretty much live.

On the other hand, spending a little bit, to buy a storage medium, like HDD, SD cards or even blank DVD disks, is an even better option. Since then you, yourself control the files. A lot of telephone manufacturers exclude an ability to use an SD card, to force people to use a "Cloud" service. Please stay away from such devices.

"Cloud" computing

While using a server to store files is at least understandable. And you can take measures to prevent misusing of your files, like encrypting them before-hand. With "Cloud" computing there is nothing you can do.

SaaSS - Service as a Software Substitute

This term is used by the Free Software activists to talk about "Cloud" computing. SaaSS is any service to which an offline software either exists or could exist. That does not need connection to the internet to be implemented, but that is presented as an internet service anyway.

For example: Translation of text from one language to another. On a very basic level it could be done using a dictionary file. But for something like Google Translate, you will need a set of files on different languages to go through a neural network software. To produce a translation data file. That can be used to translate more accurately.

Note that non of those need internet. They need data. Files that come with the software, to help software produce the outcome that you desire. Like game art in a game. If Google wanted to, they could distribute their Google Translate as a package. And make various translation data files, like French to

English, be a kind of add-on that you download and install locally.

But instead they chose to make a service of translating. Perhaps because the text that you are translating is valuable to them. For example. They can sell it as your personal data. If the program was fully offline. They would not have this ability.

On the other hand SaaSS is the worst nightmare in terms of user freedom. I talk quite a lot on this channel about the 4 essential freedoms that all computer users should have. For example. With SaaSS you can't run the program when ever you want, only when the server is running. You can't run it for what ever purpose, usually. Since the service has their terms of service. You cannot study the source code and make modifications to it. Since it's not running on your personal computer. And demanding being able to modify the software on somebodies else computer is not nice. And so on...

In order to control your computer, you have to have the 4 essential freedoms with all software you are using. And with SaaSS you are not getting it.

Free Software SaaSS and AGPL

There is an interesting Free Software project licensed under AGPL which provides server software for a SaaSS translator. It's called LibreTranslate. They claim :

> 100% Self-Hosted. No Limits. No Ties to Proprietary Services.

Which is good. But... Instead of making a program that's installed on your computer, they made another SaaSS service. Though this time, with the full source code available, and installable on your own server. They even licensed it under the GNU AGPL with an additional clause to require anyone who runs the software as a server to provide it's corresponding source code to the users.

The irony of this situation is quite amazing. Most people will use the SaaSS server and not install it locally. Since installing it locally requires creating a whole server, and then browsing the local website with a browser. This is just insane. At least they could package it into an Electron app, or something.

I'm not against having a Free translator Software. The problem here, in my opinion, is that even in Free Software people do not see SaaSS as a problem. A lot of people advertise GNU / Linux with words "All you need is a browser these days.". And while it's true that there is enough SaaSS to make most of your work in the browser. This is not Freedom.

A lot of people insist on developing any software under the AGPL license today since other people might turn their software into proprietary SaaSS. But instead, I would argue against using SaaSS. People should use software when ever possible. To have maximum control over their computing.

Web Applications

One of the hardest things to put into words are Web Applications. Some are okay. You can think about it like this. If you have all the computing power, all the storage, and all the data files that you need. Would this program be implementable offline? If yes, you are using SaaSS. If not, like with publication platforms, or chat services. Or email. It's okay to have a server that serves this.

Web applications, with Electron and progressive web apps blur the lines of a program and a service so much that the whole thing becomes a mess. When I load an Electron app, I'm not sure if this feature is implemented locally, or this is a part of a SaaSS service. This is probably what's rubbing a lot of people wrong about having electron apps.

But even with pure browser thing, what part is executed where? Is it a part of the JavaScript malware that I got. Or is it sent to a server somewhere and executed there. I don't know. And this makes me very angry.

Your Turn To Help Free Software!

In this article I want to make a design context. We already made a few articles back a very good posted explaining the 4 essential freedoms. And the readers helped me translate it.

This time I want a poster that will explain SaaSS and why it's not good. Something like the famous There is no Cloud stickers. But better. And that will make normal people understand what we are talking about from nearly a glance.

Contribution guidelines: - Do not use assets with non Free Licenses. CC-0, CC-BY and CC-BY-SA are fine. But some will require you to credit the author of the asset. - Publish your version with a Free License yourself. I recommend CC-BY-SA. Since it will protect the work from being turned into non Free. - Publish the work with sources. So we could translate, modify and improve upon the work. - Use formats favorable to Free Software. Something that will not require any proprietary program to modify.

There are links in the bottom, where you can send your designs. I want to see some good ones.

Happy Hacking!

Anti Cheat is MALWARE !!!

People don’t like when other people in the same game cheat. Thus anti-cheat became popular. But there is a problem with it. Anti-Cheat is malicious by design.

`lbry://@blenderdumbass:f/Anti-Cheat-Is-Malware:6`

A lot of people are excited that Valve are working on bringing Anti-Cheat software to GNU / Linux for 100% game compatibility. But unfortunately there is a problem. Valve is a company that wants to make profit. I don't object to making profit. I myself very enjoy having money. And using it the way I want. Valve is doing it primarily to sell their new Steam Deck console. And it's seems to be working for them.

But, as I said, there is a problem. A problem that people may not understand. And that I will try to explain in this article.

My experience with Multiplayer

When I was young I was learning is a very religious school. They were afraid of any idea that might turn us away from the religion so much, that Internet was prohibited. But we still played proprietary multiplayer games in that school.

Those were usually very old car games, since first person shooters "teach to kill" and it's not a good lesson for a Yeshivbocher (Religious Young Man). They would even avoid Need For Speed games, unless pre-modified to remove all the pictures of women. Which we learned to do quick, since they were more fun than the old stuff.

The computer class used to have a router. A router without a connection to the external web. It was only serving connections between the computers in the class room. And that's it.

When a multiplayer session would start, every person in that game would sit elbow to elbow, back to back, with the other person. Making a room full of sweaty kids playing one game on multiple monitors. Every body trusted each other. Everybody knew each other. And if one person would cheat, he would suffer consequences in the social life of that school later. So cheating was very rare.

A few times where a cheater appeared. It was usually either a bug, or a fun little joke, that was understood by everyone else. There were no accounts online. No Kill to Death ratios. No statistics. No reputations at stake. It was gaming for the fun of it. And everybody understood it.

Compare it to the multiplayer on the real internet. With lonely people sitting in front of their screens, nearly 24 / 7, sweating over the next achievement, screaming at this dude on the other side of the map who took a gun that kills too easily. Complaining about how people on the other side of the world chose to play the game. And caring way too much about their Kill to Death ratios.

This doesn't resemble fun. It resembles labor.

An Average Multiplayer Gamer

Since I don't play proprietary games, I don't experience it myself. But I know a person, which anyone who had a call with me is probably familiar with, my brother @bypiffa. He is obsessed with a multiplayer shooter game by a malware manufacturer called Battlefield. As I type this article, he is currently screaming at the game. He runs Windows to play it, since it requires the Aniti-Cheat software.

Just to make it clear. I don't touch his computer. When I say that I don't have a Windows installation in the house. I mean it. Since I refuse to touch his installation of it. The only thing that I will allow myself to do with it, is to click the reboot button.

Since he has a full GNU / Linux system on the other drive. And I can boot into that one instead.

I hear him complain a lot. At first I was shocked that he doesn't like other players to use specific types of weapons. Like Shotguns and Type 2A for example. His reasoning is that those weapons are too powerful. And it becomes a point and click game for a player that uses them. Which doesn't resemble "pro-gaming", as he says. He says that it's "Guns for Noobs". And gets very angry in anyone use those weapons in the game.

Later he complained about a DLC item that you can buy in one of the Call Of Duty games. That make your steps silent. Making you able to kill without the other person hearing you. He often complains about cheaters in the game. And if he finds a cheater software ad, he gets pissed off.

An Anti-Cheat software, by what ever company that develops it, tries to solve that problem. To balance the game a bit, so the players would not feel terrible about loosing too much, to those who figured out a way to cheat. Funny, how on the other hand they implement features like DLC items for beginners to play easier with the pros. Which are "legal" ways to cheat in the game.

It seems like the Anti-Cheat software is meant to be this promotion card in an Airliner that explains you how to "save" yourself when the plane is falling from the sky. Interestingly enough almost no people survive plain-crashes. So these promotions are only meant to make you feel better. Similar to how anti-cheat makes you feel better. Or how DRM makes publishers feel better. All those thing do not work in the way it advertised to work. It's just meant to make you think it works.

Not true.

I just finished the last chapter of this article with a sentence that I personally disagree with. But that worth exploring non the less. DRM, for example, works. But not in a way that you think. Even though DRM is implemented almost everywhere, we can still find DRM free copies of almost anything. People call them pirated, but I would disagree. It's unreasonable to compare people who share copies with people who violently attack ships.

I'm talking here about DRM to give a reference frame for my next argument about Anti-Cheat Software.

DRM is not meant to stop you from copying the file technically. It has to do something. But this restriction doesn't need to be sophisticated to work. Since together with the adoption of DRM, United States passed a law prohibiting breaking DRM.

Basically, if you develop even a smallest, stupidest program, that works as a kind of DRM. It doesn't need to work really. It just needs to do the most basic job, so the law will recognize it as a DRM implementation. And then the person that copies a file restricted by it, breaking this DRM, becomes a criminal. And it's a few years of prison in the United States.

This is a primitive, but simple example of companies wanting to take more power into their hands, easily. They will put basic forms of DRM into basically anything to have power over you. It could be a sophisticated Blu-Ray encryption. Which actually hard to break. Or it could be a script in a browser that hides the URL of the video, so you would not be able to download it.

If you use Software under the GNU General Public License Version 3. You can legally break DRM, if it's implemented in this software. In the license it has a specific section allowing you to do such. This is why it's recommended to use GPLv3 (or later). Additional section exists in Apache 2.0 license. It's recommended if by any reason, you want your code to be implementable in proprietary software.

Now, think about this. If DRM is to control you better. What is Anti-Cheat for? Let's look at how it works, to find answers.

How Anti-Cheat works?

We cannot know for sure how it's implemented, since it's proprietary software. And there is no source code available to read. But some people were able to revere-engineer an answer for us. And I will present to you a very brief version of that answer.

There are multiple versions of Anti-Cheat software. Each is implemented differently. But works on the same premise. Detecting software running either in the modified version of the game, or along side it, that changes the perception of this game to the player. Or edits some values, to make the player have an unreasonable advantage in the game.

To do this, the anti-cheat software should have access to the entire computer. All the files, all the stuff currently in RAM, all the stuff running in the processor. One of the hardest challenges in implementing anti-cheat on GNU / Linux is that it has to run on the kernel level. Or even above it. For one, because the kernel it self, on GNU / Linux is modifiable.

Do you like the 4 Essential Freedoms? With Anti-cheat, forget about it. You can't modify anything. Since you will be recorded. And I'm not exaggerating. Anti-Cheat software constantly records everything and sends it over to the developers. So they could find the "cheaters" and report them.

The price that you pay for a "fair game" with a stranger is literally your freedom and privacy. It's not Facebook in a browser. It's not a little app that might mess up some files. It's a system overlord that watches even your kernel choices. If I would make a list of the most malicious software ideas ever. Anti-Cheat would be at the top of this list. Below maybe only a Universal Back Door.

How to play then?

There was a lecture of Richard Stallman that I saw a while ago. In the end of the lecture he took questions from the people in the audience. And a smart kid asked his opinion about the Anti-Cheat software. I don't remember his exact quote. But he answered something like - "Play with people you trust.".

With Free Software you have the 4 Essential Freedoms. You are Free to modify the software. If a multiplayer game is implemented as Free Software, it could not stop you from cheating in this games. Since you may exercise your freedom 1. And modify the game to do anything you would like it to do.

Some stuff could be partially solved by coding most of the game in the server software. So the server would do the calculations for each player equally and send back the results. It might introduce lag on movement. It might make the game a bit slower. But with modern connections, it's implementable. And thus if multiple people connect to the same server to play the game, they will be playing with equal rules.

But this doesn't stop people from looking through objects and doing things like aiming automatically. Hell, they can code a whole bot, to play the game instead of them. Since the server cannot enforce how the game in rendered and controlled by it's user. The server might calculate the entire frame of the game. But there we enter the SaaSS territory.

So I think there could be 3 solutions.

Solution 1

My memory of the multiplayer in that school *(earlier)* was a fun one. It was a real game. A real social event. I think making games that encourage people to play in person with one another is a good thing.

One thing you can do is split screen. This way you will play the same exact copy of the game at ones. This will make anything in the game be equal to both users. And the users could modify the game as much as they like. They both would experience the modifications.

Alternatively, you can encourage the game to be played only by people that know each other very well. They can setup a special encryption key for their game, so only people that they trust could join the servers. Maybe, normal multiplayer servers could also be accessible. But with a "Random Stuff Warning" or something. So people would know that on these servers random shit could happen to them.

Solution 2

Design a game in which cheating is the main feature. And the winner is the best cheater. This will make people interested in coding. Since you will encourage them to modify their copy of the game, until they win the opponent's copy.

Combining it together with the previous solution, would make for a very interesting hacking game. We could probably base it on something like Colobot. Where hacking code is the main feature of the game. You have to program robots to optimize wining conditions. The last level requires your colony to be very optimized. Winning it manually is nearly impossible.

Solutions 3

Making a game that's not competitive in a real way. Where the game it self is built upon the fun factor. Where there is not a lot of competition. You can think of something like VR Chat. Where the whole idea is to hang out together with your own skins in a virtual world.

Maybe you can even base the game on the random fun stuff that can occur from different people having their own copies. For example, somebody coded him self a feature that nobody else has. And this feature makes other players amazed. Maybe those modifications could have a publication place. Where various modifications could be rated by other players. Which will encourage people to make things that are actually fun. So people could use those, and have this same fun.

Your Turn To Help Free Software!

Usually I have a section where I ask people to help Free Software. This time I could ask you to build such games, but I feel like it's too much of a burden. Too much to ask.

I could ask you to talk to people about Anti-Cheat and how it's not good to have it. But people are excised to use GNU / Linux because Anti-Cheat is coming. And it's a very complex moral issue at stake here. What is a better benefit? Not having Anti-Cheat, or making more people go one step towards Freedom? It's a hard philosophical question to answer. Even Richard Stallman doesn't have a straight answer.

I would still talk to people about malware in gaming. Like the problems with Anti-Cheat. So even if they are still going to use it, this would be an informed decision. A one that they would not feel good about. I know most people don't see games as potential malware. It's a very weird thing for some people. But ultimately games are software and software can be malware. Especially if it's proprietary.

We are developing a Free Software Gaming Platform (#FreeGILE:matrix.org) where after we gonna publish all current Free Software games, we would probably need to develop more games, to make the platform a bit bigger. Maybe after Moria's Race I will make a game development project.

I don't know what to ask you in this article. I'm literally confused. Maybe I would ask you to figure out what to do about this issue. Maybe more brains then just mine can contribute to solving this problem. And then we could have some kind of solution that we can implement.

Yes. I think this is good. This is what I gonna ask you. Talk to smart people about this. Ask them, how with the 4 essential freedoms a game could be multiplayer and still playable? Maybe if a lot of us will tackle this issue, we will finally arrive at a good enough answer. An answer worth implementing. Maybe by the end, this article will inspire a breathtaking Free Software multiplayer game.

If you want to start chatting with me about this, for now you can use the (#BlenderDumbassChat:matrix.org) chat room. I think if all of us will join the conversation in the same place, it's gonna be easier to keep track of it. But if you want to talk about it elsewhere, you are welcome.

Happy Hacking!

Proprietary Software Companies Love Free Software (When It's Not Copylefted)

Programmers are the laziest people in the world. If they can copy paste code instead of writing it, they will do it.

lbry://@blenderdumbass:f/Proprietary-Software-Companies-Love-Free-Software-whenitsnotcopylefted:2

Audio version of this article:

lbry://@blenderdumbass:f/Proprietary-Software-Companies-Love-Free-Software-whenitsnotcopylefted-audio:b

If you read the Microsoft Windows License you will find a very interesting line. A line in which they mention the GNU Lesser General Public License. Admitting to the fact that to build Windows they used Free Software code published under this license. It's not a secret that Proprietary Software companies use code from Free Software when ever they can. Paying for programmers is expensive. Writing code is hard. Using an already written code is way cheaper and easier.

Look at Disney and their main business model. They take things that are in public domain. Like fairy tales. And they make films about those. Lately they started buying companies that hold copyright for various modern fairy tales. Star Wars, Marvel... They are trying to expend their resume of stuff, without writing anything of their own.

This is just like a proprietary software company that gets code under a push over license. And uses it to create their proprietary software. Without sweating a single drop of sweat. Without flexing a single muscle.

Without acknowledging our work unless legally required. And even then, putting it in a document that nobody reads and claiming that they made the hard work of "writing all this code".

Hypocrisy of Calling GPL a virus.

It's widely documented that in the 90s and through the 2000s Microsoft has called the GNU General Public License a virus. A virus since if they use the code of the GNU GPL, they will have to release the whole thing under the GNU GPL. They were actively fighting against it. Trying to prove that it's illegal. Lobbying for making it invalid.

The motivation behind their actions is quite understandable. There is a lot of good Free Software that they might use. With source code fully available that they want to copy paste. But it will make their program Free Software too. Which they don't like.

They started spreading this idea that GPL is not Free since it doesn't give them freedom to use it in Proprietary Software. They have convinced a number of developers to publish their code under a push over license. They made it so MIT, Apache 2.0 and other licenses similar to them are quite popular.

But why? Because they don't want to share. Well, we don't want to share with them. And they are calling us wrong for not sharing with them. While they don't want to share with nobody. If they would want to share, there would not be a problem. The problem starts only when they don't want to share.

Proprietary Software companies claim that their code is so important. So secret. So expensive. That sharing it is not an option. Thus they don't share it. Even though most of their code is probably copy pasted from Free Software code anyways.

We come and do the same exact thing. But only allow using the code on one condition. If you turn all of your code into Free Software. And suddenly they are in rage. They are free not to use our code. As we can't use their code, since they are not sharing it. But they are in such a need to use our code that they are in rage. They suddenly forgot how to program. If Free Software is copylefted they suddenly can't develop.

If you are a proprietary software developer and you claim that it's not the case. Please prove it by **sharing with us the source code.** Don't worry, we are going to use it primarily to search for parts of it on git repositories. *Maybe to reverse engineer it.*

Strategical Push Over Licenses

Free Software developers are not stupid. And sometimes a push over license that allows proprietary software companies to use the code is important. Usually it's wrong to do it. But there are some strategical exceptions.

For example if a Free Software developer wants to develop a protocol, or a file format that should be supported by the proprietary software client, to gain popularity. It's okay to publish it's source code under a push over license. For example as with OGG format. To make it able to rival MP3 it should be supported by, for example, the Windows Media Player.

Of course while some companies, like Microsoft, don't mind implementing formats like this for more compatibility. Other companies like Apple will refuse anyway. If you try loading an OGG audio, or a video in WEBM format on Apple's hardware. For example if you try to watch videos from AUDIO-VIDEO.GNU.ORG on an iMonster, it will refuse to load. Since OGG and WEBM rival their own, MP3 and MP4 formats.

You can read the licensing information of FFMPEG to understand how complex the issue is with media formats. A lot of companies design things and release libraries for them. And a lot of those libraries are not compatible with a lot of Free Software licenses. Thus making the whole thing a big mess.

This is why we need our formats to be supported everywhere. Using a format that's maybe even reverse engineered. That probably is compatible with most Free Software. Still makes it a bureaucracy headache. Because even if the format is implemented in Free Software, it's not necessarily also Patent Free. Which may cause even more bureaucracy headache.

For example MP3 patent in the US expired only in 2017. Which is not that long ago. Before that any implementation of it, Free or Proprietary had to pay a licensing fee to the Patent Holders. For example the

Free Software MP3 encoder LAME, could not publish it's binaries. Only the source code, as it was Free Speech. But not working software.

For MP4 the licensing problem continues until at least 2027. Though if you use the H.264 codec for MP4 video. The use of the format is currently gratis. Still they might change their mind. And the next day you will have to pay. This is why using WEBM or OGG video is recommended instead of MP4. At least if you live in a country where Software Patents are a thing.

Telegram X & Visual Studio Code

Pavel Durov, a man behind Telegram, is a person I feel very conflicted about. From one side he made a wonderful messenger that he released under the GNU GPLv3. *At least it's client side*. From the other side, he made a Facebook clone named Vkontakte to do the same exact business as in Facebook. Only in Russia.

Russian government, obviously, came to Pavel Durov with demands to get private data of the people using Vkontakte. He didn't give it to them. What a man... He sold the company to somebody who did. The other person renamed it to VK.COM and now, in Russia, there are two Facebooks.

He could design Vkontakte so it doesn't collect data. So he could just say "I'm sorry, it's technically impossible." But instead he made a decision to collect data and just personally not to give it away. Instead let someone else give it away. At least he didn't do the same with Telegram right?

Oh... what? The Telegram servers contain all the messages in an unencrypted form? At least it's Free Software. Oh... the servers are proprietary? At least I have choice of what client to use. Here Telegram X. Looks nice. Oh it's proprietary?

Telegram is a weird thing of the modern era. It's a product presented to people who like Free Software. A product that's designed to convince Free Software enthusiasts and privacy nerds into using it. But, actually, just another trick of a proprietary software developer. There were even talks about Pavel Durov trying to sell Telegram to the same people he sold Vkontakte to.

Okay enough Telegram. Think about how much we are winning. Visual Studio Code is a Free Software program released by Microsoft of all companies. Isn't it great? Well... The official version is proprietary. The source code is free. And you can build it from source. There are even projects distributing Free versions of it. But it's just another bait. Another trick in the sleeve of Proprietary Software companies. Another way to make us be used by their stuff.

Getting away with GPL violations.

Later, when companies like Microsoft understood that their efforts of trying to cancel the GPL were failing. They started to do something else. They started to pay serious money to lawyers to find loopholes in the GPL. Trying to find ways to get away with GPL violations.

Mobile phone companies use this GPL loophole concept to control software on mobile phones, while still using the convenient Linux Kernel for the operating system. If Linus Torvalds had updated the license to the GPLv3, the loophole would be closed. Since as soon as Free Software Foundation started discovering those loopholes, they started drafting a new license. Which gonna word it self so no known loopholes would be possible.

Recently Microsoft found another loophole. What if they scan through a large collection of software code and push it through a neural network. This neural network will get a sample of code, started by a developer. And will fill the code with suggestions, learned by scanning all this code.

This new invention they called Github Copilot. Obviously the question of legality of this invention was placed upon Microsoft, since a lot of software on Github is thankfully copylefted. And thus, it might infringe on the copyleft. It might be a GPL violation. To which the answer is yet to be found. It's unclear whether it's copying pieces of code, or does something else based on the neural network algorithm. And it's not yet tested in court.

Your Turn To Help Free Software!

This is a part of my articles where I want to encourage you to help me deal with things. I have this weird issue, that I don't like to have problems just sit there, unresolved. I don't like annoyances. And at least, if I can, I'd like to brainstorm a possible solution. Implementing it would be a much better thing. But one person has only so many hours in his day.

At the Free Software Foundation website, there is a campaign to fight against Github Copilot. They offer a price of $500 to a person that will do an interesting thing.

From the page:

> With all these questions, many of them with legal implications that at first glance may have not been previously tested in a court of law, there aren't many simple answers. To get the answers the community needs, and to identify the best opportunities for defending

> user freedom in this space, the FSF is announcing a funded call for white papers to address Copilot, copyright, machine learning, and free software.

They need experts in the fields of computer science and copyright law to provide them information about the current GitHub Copilot problem in order to know, what kind of actions to do about this.

There are 3 things that you can do to help:

1. Migrate your repositories to something else. Stop using Github. I would recommend notabug.org, but you may choose something else if you so desire.

1. Share the FSF page. Let people know that the Call For White Papers On Github Copilot is a real thing. Spread this message, so the expert will appear.

1. Research the topic at hand. And write the White Paper yourself. Become the expert. You may even get the $500 reward.

Happy Hacking!

Why does Book Burning Continues in 2021?

Book Burning – Activity of burning books. Usually with ideas a given party disagrees. Used by Nazis in Germany and other occupied countries during the World Was II to silence voices that were unacceptable to the regime.

`lbry://@blenderdumbass:f/Book-Burning:e`

In the late 30s and early 40s, Nazis used to burn books that they didn't agree with. It was an act of victory over the people who did "wrong-think". The books could range from communism propaganda to Jewish Religious texts. But all it was doing, is trying to make people exited about censorship. A very heavy act of censorship.

It looks like any ideology is pro freedom of speech until they reach a critical mass. As soon as they are, they become pro censorship. Let's say a new idea starts a small movement. Something that most people still disagree about. This movement will use their Freedom Of Speech to argue for their new movement. They will be all for Freedom Of Speech, since if there was no such freedom, they would already be defeated.

But as soon as they get enough people on their side. As soon as they become "mainstream", they will switch. They will now lobby for removal of Freedom Of Speech. They will lobby for censorship. They will start arguing why certain ideas are "too dangerous". Since as soon as they are "mainstream", Freedom Of

Speech is a thread. A thread to their ideology's current "mainstream" status.

Social Justice Warriors

You can hate on SJWs as much as you like. But there is an interesting observation that I noticed about them. I am an SJW myself. But not in the way most of them are. Social Justice Warrior means a person that seeks Social Justice. And while most people today associate SJW with angry Karen with colored hair, screaming about white men being too privileged. It's actually about any type of political activism.

You can use the words Political Activist or simple Activist to describe an SJW. But I think the problem of why most of them are so hated today has to do less with the message and more with the messenger. SJW is associated with breaking logic to argue about an issue that might be of a great importance.

In an article Misinformation is Free Speech I outlined an idea that any type of mistake made by a person using speech is still except-able. Since people cannot be correct. Even more then that, most arguments are wrong anyway. And we can only try to be less wrong.

But with some people, they don't care about seeking the truth at all. They see the political landscape as "us versus them". It's not about who's opinion is right. Is not even about trying to argue for why you think you are correct. But rather it's a fight using words.

Most SJWs today are ignorant of other person's perception of the situation. They are usually convinced a certain argument to be true. And they are convinced that everybody else also is convinced. And if they see somebody that is not convinced, they perceive this person as a lunatic. They get angry. While in reality, they are probably filtering their own perception of opinions around them.

When you see a person that disagrees with you. One type of people will understand that the other person has a different viewpoint. And they can try to persuade this person into changing his mind.

Second type of person will be convinced that they are right and that the other person knows and believes it already too. So they will argue in a fashion of saying "You know you are wrong.". Which is not nearly a productive conversation.

This second type of persons lead to heated arguments. Leads to people being divided. Leads to one group hating the other. Leads to the reputation of a Political Activists to be similar to the reputation of a Tin Foil Hat Conspiracy Theorists.

I don't hate the messages of Respecting groups. I agree with them. I'm anti-racism, anti-sexism, anti-ageism, anti-any-ism. But I do not agree with the presentation techniques used. And while Freedom of Speech is there and people can use any words in any combination they like. I believe that a more careful wording could help any argument. Just knowing that you are free to express yourself, doesn't necessarily mean that people have to believe you.

Look at Stephen Fry for example. A proud gay person that is an SJW no less then other people. But that is so good at putting words together that he is one of the most respected political activists out there. He is pro Freedom of Speech. Which is weird for a person of his political viewpoint.

I think the lesson here is: When you know how to argue. You are not afraid of other people arguing against you.

Richard Stallman and the FSF

A lot people criticize the FSF and Richard Stallman on how strong they act against the Proprietary Software. For instance they criticize the words used and recommended by the Free Software Foundation. Things like "Please say GNU / Linux instead of Linux." or "Please do not use the words Intellectual Property. This is misleading.".

There is a whole page of words to avoid on the GNU.ORG website. This looks like a form of word policing. A form of censorship. The same kind of thing an SJW would do to police other speech.

From the other side, Free Software Foundation also fights for Freedom of Speech. For example the Freedom 0 means to use the software **for what ever purpose**, including those, the developer or the FSF disagrees with the user about. Which is weird and seems counter intuitive. It seems almost like a hypocrisy. What is going on here?

If you look on the page mentioned above. The page starts with:

> There are a number of words and phrases that we recommend avoiding, or avoiding in certain contexts and usages. Some are ambiguous or misleading; others presuppose a viewpoint that we disagree with, and we hope you disagree with it too.

In the end of the page, you can see that the page is under the Creative Commons Attribution-NoDerivatives 4.0 International License. Which is not a Free / Libre license. What is going on here?

If you go to FSF.ORG site instead of GNU.ORG and look for any article. For example this one. You can see that they use the same, non free license there too. This time with a link "Why this license?".

This links to a text explaining the decision of using this license.

> Works that express someone's opinion—memoirs, editorials, and so on—serve a fundamentally different purpose than works for practical use like software and

> documentation. Because of this, we expect them to provide recipients with a different set of permissions: just the permission to copy and distribute the work verbatim. Richard Stallman discusses this frequently in his speeches.

Basically, to avoid random people editing somebodies opinion and presenting it as the original opinion. They use a license that prevents changing it's text. Which makes a lot of sense. Now you can notice the word "opinion", "recommendation". It's not the same as trying to make a law prohibiting words. It's just a set of arguments why from the point of view of the FSF certain words do not make sense. You may or may not agree with this opinion.

When Bryan Lunduke in the end of his interview with Richard Stallman asked him about what he would do when Proprietary Software will be illegal. Richard answered that he doesn't want it to be illegal. He doesn't fight for a legislation of making Proprietary Software illegal. He fights for users to choose software better. For developers to release software better. He fights for Freedom. Not for restrictions.

Your Turn To Help Free Software!

This is a part of the article where I usually ask people to help Free Software. You can look for my earlier articles for other things that you can do still.

This one will be a little bit more selfish. But still will help both Free Software and Free Speech.

Odysee.com is a bloated mess and it has a built in censorship. For example all publications tagged as mature are not accessible from Odysee.com. But they are hosted on the LBRY network.

To make a restriction free, non-bloated LBRY client, somebody needs to make the work. And hopefully I started the work. The repository is there. The only problem is... I fried my brains, trying to do too many things at ones.

So please, go to the Official notabug.org Git Repository of FastLBRY and start hacking on it. You can join the chat at FastLBRY:matrix.org to discuss various things and ask me questions.
Happy Hacking!

Free Software that's NOT AN ALTERNATIVE!

We always talk about alternatives when it comes to Free Software. But how about taking a different approach and building something truly unique?

lbry://@blenderdumbass:f/freesoftware-thats-not-an-alternative:1

A lot of people are looking for Free Software alternatives these days. Some are moving from Maya to Blender. Some are moving from Photoshop to GIMP or Krita. Some are moving from Windows and Mac to GNU / Linux. But the problem is, they are alternatives. Alternatives to something that's considered "the default". Alternatives to the "industry standard". Not original software.

A lot of people call this "Alt-Tech". This is a word I hear a lot when people describe Odysee.com or Bitchute. Funny thing is, Bitchute being an "alternative" to YouTube is not Free Software. While Odysee is. But what's rubbing me wrong here is the presentation.

The LBRY Inc. people behind Odysee.com developed a wonderful LBRY protocol. A protocol capable of very interesting things if you have enough imagination. But they decided to call the whole thing an "alternative to YouTube". Implementing with it, the YouTube Sync program and other features, to strengthen this statement.

Most of it is not an alternative

LBRY is not an alternative. It's probably something like a mishmash of various protocols and formats to create something unique. You can think about it as Torrent meets Bitcoin. But the LBRY developers gone a route of advertising it as an alternative to YouTube. Maybe because people were frustrated with YouTube.

And don't get me wrong. This kind of advertisement works. When a person is frustrated about one piece of software, he might search for it's "alternative".

But there is a huge difference between calling LBRY or PeerTube an alternative for YouTube. And, lets say, calling FastLBRY an alternative for Odysee. One is it's own thing. While the other is a different client for the same protocol.

How much should a given software be different to be called an original? Is it just a use of a different protocol / format? So then LibreOffice, as it defaults to ODT instead of DOC is an original program and not a Microsoft Word alternative. Or is the line somewhere else?

Trying to develop an alternative

I think most of the Free Software developers are trying to make alternatives instead of focusing on new and original stuff. You can look no further than the Free Software games.

Super Tux is an alternative to Mario. Super Tux Kart is an alternative to Mario Kart. Minetest is a clone of Minecraft. Nexuiz is a clone of Quake. O A.D. is trying to be Age of Empires.

Even with non-game Free Software. GNU's not UNIX. The name it self make it clear that GNU is an alternative to UNIX. As Linux is an alternative to the UNIX kernel.

It seems like most people are just stuck trying to copy as many features from the proprietary programs as they can. Trying to copy as many proprietary software as they can. Not really thinking in terms of what they can make, but in terms of what they don't have, that people outside of the Free World have.

I don't particularly hate this approach. As sometimes, it's not easy to argue with people about

Free Software if you cannot show them that they could still do all what they were doing, but in Freedom. So alternatives and clones are welcome. But...

I notice people get excited about GNOME more

People are often saying that for the new GNU / Linux user, something like *GNU/*Linux Mint or Solus OS is good. It mimics the layout and feel of Windows. And thus people will not feel wrong. Will not need to learn a whole lot about their computer.

Say that to Steve Jobs when he presented the iMosnter. Think about people not wanting to use a pretty / animated touch device, since they are not familiar with touch. And thus, the UI of iPhones should mimic the button layout and menu structure of the older, button phones. Even tho I dislike Apple a lot. This would still be ridiculous.

I remember installing GNU / Linux to some new users, to restore their computer. They had it broken and they thought to throw it away. I found that the computer was fine, but they accidentally erased the disk. So I took this opportunity to teach them about Software Freedom.

I went with a risky decision of installing GNOME right away on their machine. Instead of going for some "windows-like" UI. They were excited as hell. Suddenly, their computer is not this boring machine. But a pretty / animated, fun to use, exciting device.

The husband of the family sat down in front of the computer and could not stop playing with it. And this guy already had grey hair. I turned these old people into young children again. By not restoring their old device. By not trying to imitate their old device. But by making it fresh and new.

I think that developers of Free Software should take a lesson from GNOME. Even though GNOME could be bloated at times. Even though I use KDE Plasma at the moment. GNOME is unique. It's not trying to be a mere alternative.

Designing fresh and unique

Two good examples of Free Software that could fall under a category of an alternative. But that were developed by smart people that had a unique vision would be Blender and Emacs.

Blender could just try to imitate the UI of let's say Maya. And try to implement the same exact features in the same exact way. Trying to attract Maya users as a kind of Free / Gratis alternative to Maya.

But instead, the Blender developers took an attitude of **Fuck You All**. And until recently, it was so hard to learn, most people gave up. Selecting object was done using Right Click before 2.80. And now, you still can use this setting. They just provided a new setting for newbies, to select using the regular left click. Until 2.50 Blender would not save using Ctrl-S. But you had to use Ctrl-W. (In most software it closes the file instead of saving it)

Also Blender developers are trying to focus on Blender's internal design and flush it as much as possible to the user. Instead of hiding it away under UI that's meant to mimic something else.

Odysee for example is complete opposite. It has an interesting protocol that could be presented in a way that will give it justice. But instead you see a player taking the large majority of the window on the left side. A list of recommended publications on the right. And a comment section below. Trying to imitate the design of YouTube. A video platform. While LBRY supports publishing of literally anything at all.

Blender has a feature called Outliner. It's a UI browser that let's you browse the data inside the current .blend file. It exposes you to all of the internal parts of the file's structure. Letting you interact with literally anything inside the file. Alternatively there is a python console inside blender, but with a modification. A modification that let's you browse the Blender's python API. To find all the data that you need in order to modify absolutely anything you want.

Compare it to Odysee. The LBRY protocol is so much richer than what people expect from it by looking at Odysee.com. Since Odysee is not trying to be an implementation of LBRY. It's trying to be an alternative to YouTube.

Also, Emacs. It's technically an alternative text editor. But... you know. It's an entire operating system of it's own, in the same time.

Your Turn To Help Free Software!

This is a part of where I want to encourage people to help Free Software. And this time I want to encourage developers. I want to encourage them to give proper treatment to the stuff that they are developing. Not to try and make something that's a mere alternative. But to try and think outside of the box.

Imagine that only your program existed. *There was never a Photoshop, only GIMP exists.* What would you do differently? What would be the right UI, or the right protocol / feature, that would make the software the right judgement.

Stop thinking about " *that* feature implemented by *that* proprietary program". Start thinking about what would actually be cool to have with **this project**.

What fits here... Not what you except to fit here, since it's done in a different place.

If you are not a programmer, or you are lazy, or you don't have much time. Help us think about the UI implementation of FastLBRY GTK that will come next. We don't want it to look like YouTube. We want it to look like LBRY. Maybe we can do a better work than the LBRY Inc. developers at presenting the protocol.

Happy Hacking!

Freedom to Insult (The Odysee Controversy)

Sometimes people are pro Freedom of Speech only if they are personally weren't attacked by it.

lbry://@blenderdumbass:f/Freedom-to-Insult:5

In this article I am going to talk about the recent Odysee Issue. It involves swearing, gross stuff, and politics. So if you don't like these kind of things, you've been warned. Also this post will not have the usual Moria's Race and Help Free Software sections. It doesn't seem to fit the nature of this publication. But there will be something I would encourage you to do in the end, anyways.

What happened?

On July 13nth 2021 @Odysee announced a game to find most weird videos on the platform. And put them into a playlist. They announced it in their article titled ' "WTF Did I Just Watch!?" Competition '.

On August 3rd 2021 @Odysee released an article titled " Unprettiest Human Worldwide is now on Odysee ". In this article they took some time to promote a channel on the LBRY network. A channel with a vulgar style. A channel that uses swearwords and allegedly records himself having instances of strong schizophrenia. If you go to a video on this channel titled " You want to FUCK my TWO HOT GIRLFRIENDS in the ASS??? ", you can find a comment by @wesleyanderson36 saying :

> Wtf did I just watch????????????????

Probably linking this channel somehow to the @Odysee post from July 13nth.

On August 9th 2021 (6 days later) a big LBRY channel @Lunduke published an article titled " Lunduke condemns LBRY & Odysee " in which he outlined the latest issues with Odysee and why he dislikes the promotion of the Unprettiest Human channel. A part of the article:

> A few days ago, the official Odysee company channel made a post promoting a creator channel. The first such promotion, of any channel, in many months. What channel did they choose to promote? One of the hundreds of high quality channels that currently publish to the platform? One of the exciting, popular channels that have joined recently?
>
> No.
>
> The official Odysee company channel made this post, promoting one of the most low quality, vile, vulgar, degenerate channels on

> the platform. A channel with only a couple dozen subscribers, no less.

On August 22nd 2021 @Lunduke posted another article. This time titled " Lunduke content is not on Odysee / LBRY " in which he described a plan to delete all of his publications from LBRY protocol, to stop being associated with the platform.

On August 23rd 2021 @OfficialZaney on LBRY posted a video titled " Odysee Has A Serious Issue | The Lunduke Situation " in which he took time to explain the post by Odysee and the decision of Lunduke to remove his stuff from LBRY. As well as he himself announced, while not deleting his LBRY channel, he is going to avoid promoting it from now on.

Freedom of Speech vs "User Friendly"

Everybody has their own standard of what is right and what is wrong. Some people have "no skin", so to speak and anything a little bit "non-safe" will make them very angry. Others might be the type of people who sit on Tor looking for the ugliest gore porn, because of boredom. But it's not like there are only those 2 types of people. There is no line. People are all unique.

Some may agree with you on one thing and disagree with you on another thing. And sometimes you are fine with them disagreeing with you while other times you are not fine. Situations like this one shows this dynamic very well. A channel posts about something that they think deserves promoting further. Some people find it hilarious, others disagree.

As of typing this article the post by @Odysee has 53 fire and 335 slime. In this case the majority disagreed. Making a quick math (100 - (53 / (335+53) * 100)) you can see that 86.34% of people disagreed. Which gives to some people an urge to complain about the post.

Platforms like YouTube have "figured this out". They are heavily censoring the platform to avoid situations like that. They have a huge pool of data. They know what is the preference of the majority of the people. They have the budget and the personal to design any message in such a way that will take them out of the hot waters. Even if the message is highly disturbing in it's core.

The whole point of LBRY to be there, at least as some kind of "alternative to YouTube" is that YouTube is sterile. Any opinion, word, attitude or joke that's not preferred by the majority, or by YouTube executives, will get demonetised, blocked, kicked out... And all in the name of making the platform more "user-friendly". Or as they call it "Advertisers Friendly".

LBRY core concept is to make it hard for anyone (including the LBRY developers them selves) to censor things they don't like. They put up a block-chain. They made the whole thing Free Software. Just so if they will find a publication that they personally disagree with, or that makes the platform less "user-friendly", they still couldn't do much about this publication.

The Free Speech part of Odysee / LBRY idea is that something like YouTube's sterile nature could not be transferred to LBRY. It's designed against it. And what is the best way to celebrate it's success at suppressing censorship? Promote a channel that will not survive long on a sterile platform like YouTube.

Freedom to Insult

I already talked about Freedom to Misinform, as a part of Free Speech. But there is one more freedom that people like to avoid talking about. It's a freedom to insult. Without it, there cannot be Freedom of Speech.

There is a famous video of Rowan Atkinson (the actor who played Mr. Bean) giving a speech on Freedom to insult with a banner behind him saying "FEEL FREE TO **INSULT** ME!". Here is the re-upload of this video to LBRY:

lbry://@AussieFighter:8/In-full-Rowan-Atkinson-on-free-speech:c

He is arguing about ludicrousness of some English laws, making it possible for police officers to arrest people simply because they said something that could be viewed as insulting by someone else. He gives examples like a man calling a horse gay. Or a store owner displaying bible passage on a screen.

With laws like that anytime anyone opens a mouth, writes a word, or does anything that has meaning to it, it may result in this person getting arrested. Since there will always be somebody who can find this word or this action as insulting.

In my opinion people should be Free to yell at each other with swear words. To call each other expletives. To argue and to use what ever words, sounds or gestures that they like. It's not Freedom of speech if somebody has to apologise or get punished for their expressions.

Oscar Bait

Free speech does not mean required speech. Free insult does not mean required insult. You may think that I want the world to be this place of angry people going bananas on one another. While it is going to be fun to look at, I wouldn't lie. I still believe in a presentation skill. And it may require avoiding certain things.

In art, there are no rules. Only guidelines. Things that people found to work most of the time. You can argue that swearing and bad behaviour like the one displayed by the @Odysee channel is one of those things most publishers should avoid, not to scare people. And it might as well be true. Since 86.34% didn't like that post. But from the other side, to set those guidelines in stone would be a bad decision as well.

There is a term in the movie enthusiast circles called "Oscar Bait". It's describing a film that is trying to do everything by the book, but fails. It has a good structure with an emotionally rich source material. It has a good cast of well known good actors. It has a good cinematography. And it looks like this movie should take the next best picture Oscar. But it doesn't. More than that, Oscar Bait films are usually boring and un-interesting. They are not fun in anyway. It's like you've watched this film before thousand times, but didn't.

Think about a neural network making a film. It will combine things that are considered good practices in film and will most likely output another Oscar Bait. It will look expensive, feel expensive, but be this familiar, boring pulp all over again.

This is what happens when you restrict speech, or tell people what some expressions are unacceptable. You make the entire world this familiar, boring pulp. This is why YouTube is boring, while LBRY is not. It's not about the political stuff. It's about the feeling of being forced to watch the same exact video thousands of times per day. YouTube figured out, through their analytics, the way to make the most "user-friendly" videos. And through their protocol, they are pushing publishers to apply this style. Turning every next video into a copy of the last one.

YouTube is the neural network making Oscar Bait films. It will look expensive, but will be void of anything interesting. A far better strategy is to walk along the guidelines, but do a thing or two that will shock your audience. It will keep the engagement. It will make the publications better. Look at the @Odysee situation again. People reacted, yelled, thrown opinions and took actions. This post is hell of a successful act of attracting attention.

Very good film directors usually do the formulaic thing, just like the Oscar Bait directors. Sometimes they might have their own, unique style, like Wes Anderson or Martin Scorsese. But the idea is, they are trying to do something weird all the time. Little shocking thing that will get them news coverage. They are trying to play with expectations. Trying to subvert the expectations. Trying to be just a bit too vulgar, just a bit too cringe, or in some cases just a bit too friendly.

Please Insult Me!!!

I want to end this article with a request. A request to insult me. It's your freedom to do so, but you are not required. It's also my freedom to insult you back and I'm also not required.

Go to #BlenderDumbassChat:matrix.org and let’s start yelling at each other.

Happy Hacking!

Why Security by Obscurity is a ludicrous idea...

What would you rather use. A cheap plastic lock, but hidden. Or a huge volt with a half a meter width door, but that everyone can see?

`lbry://@blenderdumbass:f/security-by-obscurity:4`

A person on the #BlenderDumbassChat:matrix.org recently brought up an issue that I wanted to articulate in one of my articles. One of his parents believes that Password Managers should not be Free Software, since if it is, the source code would be available and crackers would find easier ways to crack it.

Using a proprietary software password manager, he believes, may increase the security, since crackers do not have the source code to look at. All they have is the binary which is hard to reverse engineer. Thus making it harder to know how the software works. And thus making it harder to crack the passwords.

This is called Security by Obscurity and it is a valid argument. But only to an extent. If for example, both software being implemented in the same way. And if both software have no actual security features. Then - yes. It would be a valid argument.

In reality though, there is a thing called cryptography that is used hopefully by both types of software. It uses various clever tricks to conceal information is such a way, that knowing about how the trick works doesn't increase your chances at reading the messages. More than that. Knowing how a good cryptography method works will make a cracker give up immediately.

In this article I want to touch on what is cryptography, how it works and why Security by Obscurity is not a good idea.

The Enigma

Think about a problem. A world war is going. The world is split in half. Bombs are flying at you. People die everyday from fire. It's a serious situation requiring hard thinking and good decisions. You have to communicate messages over a radio with the troops, but in such a way that your enemies will not read the messages.

This is the problem that Germans faced during the World War II and they had an idea. They could believe that simply using a language that the other's do not understand would work. Like speaking German, while you fight with Russians. But even though it could be soothing to think that Russians do not understand German, they still do. As well as British, American and any other army against the Germany at that time.

Funny how Americans during the war with Japan used an Native American language as a code for a few battles. The code was very insecure. There was no guarantee that the other side doesn't speak this language. But they used the, so called "Security by Obscurity" anyway. They found a language that not a lot of people are familiar with, to communicate messages. But then quickly realized that it was a mistake and stopped doing it.

Germans on the other hand didn't think that such an idea will work. Instead they developed a machine that was creating a coded letter when typing on it. Think about a type-writer but larger and that outputs weird, unreadable mess. This machine was called an Enigma.

The idea was, that the machine it self was clever enough, that even if enemy forces would get their hands on an Enigma machine, they still couldn't use it to break the code. They would need another thing. **The key.**

Keys in cryptography are usually data of some kind, passed through a crypto-graphic technique to either encrypt or decrypt messages. In something simple like a Cesar Cipher or the Enigma, same keys would be used to encrypt and decrypt the message. It could be a word, a long string of numbers, or anything really, that both sides would keep in secret. Thus even if the technique would be know, like the machine would be captured, or the software would be readable, unless the cracker knows the keys, he can't do much.

Breaking the Enigma

The Enigma machine was broken multiple times. A few successful attempts were done using math, pen and paper, by mathematicians. But their efforts were wasted, so to speak. Because until they break the key, Germans would already swap the key a couple of times.

The Enigma Machine had a flaw. It would mess up the characters, but would not allow the same exact character, as one typed, to be displayed in the encrypted version. Meaning, the other side had an interesting advantage. But they still couldn't do much with it.

One bright individual had to invent a computer to break the Enigma machine. His name was Allan Turing. He designed a device that would be the first security cracking machine. It would exploit the flaw to brute force a key until it gives something readable. From there, humans would input the key found by the machine into captured Enigmas and be able to read the messages.

The British, since then, used a modified version of the Enigma. They made one modification to it. Removing the restriction mentioned above. Making the machine way harder to break. The British used Free Software idea before software was a thing. They modified a flaw in a security protocol. Making security stronger.

The Enigma was a machine developed almost one hundred years ago. Since then a lot of bright people added their expertise into making better algorithms to make security stronger. Each generation had better and better computers. Which meant stronger encryption protocols.

Today the **key** is no longer used. There is now a way more interesting idea of **private key** and **public**

key. Using some very complex math, it makes possible to have a private key to encrypt messages, and a public key to decrypt them later. This allows things like digital signatures, crypto-currency, GPG encryption and more.

There is no Security by Obscurity

Even though not having source code will make it hard to learn how the software works, it will not make it impossible. We insist on source code because it's the preferred way of making modifications to the software, if we find things we personally don't like. But crackers crack anything. Proprietary or Free. In the Free World there are hackers doing reverse engineering for software. Which is de-compiling binary code and figuring out how it works.

It takes way more time and energy to do it. But there are things that were successfully built from doing reverse engineering. Things like Microsoft Office formats support in the Free Software LibreOffice. Or support for FBX and such in Blender and other 3D packages. These are harmless ways of doing reverse engineering. But some crackers do this for malicious purposes.

If you remember not so long ago a Russian cracker managed to reverse engineer a bug in Windows Operating System that allowed him to build a virus that would encrypt people's files and then require a ludicrous ransom fee to decrypt them back. There was no system preventing this from happening on windows. No protocol designed in a clever way that will not allow it. Only a promise that nobody except Microsoft car read the source code. But the cracker didn't need the source code. He needed a mere de-compiler.

De-compiling the Linux kernel or the GNU Privacy Guard will not teach you anything new about how it works. The source code is already available. And you can just read the source code. The system it self, though, is built with enough cleverness, that knowing it will not help you crack it.

If you want to learn about the cryptography techniques them selves, you can read about the Modern Cryptography on places like Wikipedia. It goes deep into the technical stuff. Making you realize how simple and in the same complex the cryptography stuff is. And how knowing or not knowing about how it works will not increase or decrease chances of breaking it.

But for the simple person I would like to present an analogy.

Analogy

Imagine two locks. One is hidden, but weak. Another one is clearly in front of you, but it's a massive, 20 ton thing. A cracker would rather go with the first one. The only challenge he will have, it's to find it. Then a simple knock on it will break it.

The massive 20 ton thing will be on display for anyone to try and break it. It will be like the Excalibur sword. It will be set in stone and only the greatest of crackers would even attempt breaking it.

Yes, King Arthur may appear. Some cracker may finally break the lock. But in the same time, we may learn from it and design a lock that fixes all those issues.

In Free Software people are interested in fixing issues

Companies that rely on Reputation may censor any of their big mistakes. Since a mistake may cause a scandal and it may result in people stopping using the products. This is why even though they probably know about a lot of vulnerabilities, they do not disclose them or patch them. It's to keep the reputation "clean".

On the other side, Free Software usually has a publicly available issues page where anyone is welcome to complain about anything. Free Software rely on good arguments to keep their reputation. And even then it might not work. Some people still think Tor is for criminals.

With Free Software anyone can not only complain about an issue, but also suggest a change to the software that might fix the issue. Since anyone is legally allowed to make their own copies of the program's source code and distribute it's changed versions. People could offer their "fixed" software. Software with modifications. That may or may not fix the issue that you are looking to fix. But the fact that this is possible make developing security features easier.

If a security vulnerability happens in a proprietary software, it will most likely get the board of directors meet and decide whether the issue is worth dealing with. If it's a big problem which may result in a loss of income for the company, they will hire developers to fix the vulnerability. But if they decide that the issue is of minor importance, they will sweep it under the rug. After all, it costs money to implement a change. Even if it's a patch to a security vulnerability.

The same thing, when happening for Free Software, causes a massive action from all around the world, of people implementing changes that will solve the issue. And the best one is probably chosen for the main fork. While other forks can coexist and compete, to create the best software.

It's not a promise that Free Software will be the best. It still could be an unknown program that nobody gives a damn about. Or it's not a promise that changes would actually be good. In the same time as there are no such promises from Proprietary

Software. Programmers are still humans. And humans make mistakes. Only with Free Software mistakes are brought to the attention way more frequently. And fixes are proposed way more frequently.

Your Turn To Help Free Software!

Some people express hopelessness towards me. In the mentioned #BlenderDumbassChat:matrix.org room, a kid had an argument with his dad, that believes in Microsoft and doesn't let this kid use a Freedom respecting operating system. There is no other computer. There is no compromise that can be made. A person is a stone cold brick, rubbing from his own child his Freedom.

Most of the arguments were easy things to confront the father character with. Things that he already understands and works towards improving. Like moving from Microsoft Office to something like LibreOffice. But the argument of security by obscurity made my blood boil, my head busy and my night sleepless.

I was thinking about this article ever since I read about this yesterday. And I could not, not write about it today. The kid is probably reading the comment section. And he is available on the chat. I will not reveal his username. Maybe he may do it him self. I still want to respect his personal privacy in that sense.

I want to ask you to help me. Help me speak to his parents and fix this issue. Help me at least get him a computer of his own, where he could put an operating system that he would choose, with software that he might use. If his parent wants to keep being a slave to Microsoft, he is Free to do so. But it would be nice if we could win him over as well.

I wrote this article for his dad to read.

Happy Hacking!

Free Software Teaches You To Be Mature

With great Freedom come great responsibility.

lbry://@blenderdumbass:f/SJhEMvfO_8dE3E-sl9ZedXxJaMzOiiXO0jRELfOY77JhCZTEHp64vbwvwTmvbJf13VHV4s:2

Not so long ago I read (using my own LBRY client) an article by @Mythologos titled "FSR Digest #2: Is There Too Much Choice In Linux?" in which he presented an interesting thought that was on my mind for a long time, but not yet articulated. And here it is, finally, articulated.

The article brings up the argument that sometimes people think that there is too much choice, while in reality they are just not ready to be mature. They want parents in the form of proprietary software companies that decide for them what is their computing going to look and feel like. They don't want to control anything. Since it's too big of a burden, too much of a responsibility, too much freedom.

With Great Freedom Come Great Responsibility

Uncle Ben in the famous Sam Raimi film was talking about Power. Control over lives of other people. Let's talk about a different type of control. Let's talk about Freedom. Control over your own life.

As children, humans don't have almost any Freedom. Everything is decided for them by parents. Thus if anything goes wrong, the parents are to blame.

When a person becomes mature, they get more and more freedom to decide things on their own. Where to live, where to work, if to go to work, how to wake up in time, what to say and so on... And if they make a bad decision, they are to blame themselves.

Some people want their Freedom more. They want to be independent. They want to show everybody that they did everything themselves. They want people to respect them. Others, on the other hand, want to keep being children.

There was a court case, turned meme, about a forty something year old man, with long hair, not wanting to leave his parents. He still lives with his mom. And it's all because responsibility (and freedom) scares him.

I know a lot of people in the Free Software world compensate for lack of freedom in the personal life by using Free Software. A lot of us are still living with our parents. A lot of people that are following this channel are teenagers that have no other choice. But some are adults that do not want to become mature.

Proprietary Software companies raise a generation of dependant children. Not children as in 17 years or younger. But children as those who are not capable of Freedom. I strongly believe anyone at any age could be capable of Freedom. But it's not easy. And thus people are scared.

But Rich People Love iMonsters

I know a lot of Rich people who refuse to use anything but Apple products. They claim that it's "quality products" or "premium products" and anything less then "premium" is not for them.

But these are the same people who don't work. They have personal for anything they need. They have a company that operates away from them. They are children, but instead of parents they have money.

Money buys them a road away from responsibility. As much as they claim to be Free, they are dependant. Dependant on the personal that cleans for them and makes them food. Dependant on Proprietary software that gives them "premium quality" animations.

There is a David Fincher film called The Game about a Rich man turning into a homeless man during a series on unfortunate circumstances. By end of the film, a person who has never even made a sandwich for him self, is stuck in Mexico, alone, with no money. He has no choice but to start dealing with the situation like a mature man. And that's what brings him back.

A Free person might not need to be Rich to make a change. Richard Stallman is estimated to have no money at all. But he is a very famous and a very vocal man. You can be Rich or Poor, the question is, how mature you are... That's what gonna matter when shit hits the fan.

Most People Don't Read Agreements

I saw multiple times how people either install software or sign a document, but in a way that justifies a famous scene in A Clockwork Orange. Where the character was signing a paper and a the police man yelled at him "Do not read, sigh!".

A lot of people are either careless, or lazy to bother reading any type of document. This results in them being frustrated when something that they thought was common sense was suddenly taken away from them.

A good example of it would be YouTube channels that believe in the fair use laws after joining Spotify. And being confused why YouTube doesn't reinstate their videos after a manual review. Well Spotify's license agreement has a section preventing the use of the music for things that are allowed by the copyright law. Same goes with Netflix and other stupid platforms.

YouTube ones had a feature to browse a music collection allowed for use in videos. They were either under the CC-0 or CC-BY license. But lately YouTube started using their own license on those songs. Presumably to make it illegal to use this music outside of YouTube.

With Free Software and the knowledge about Software Licenses, people are usually more cautious about what they sign. They are more careful, more mature.

Conclusion

Be mature. Use Free Software.

Happy Hacking!

It's NOT about the FSF or Richard Stallman

Too much people think that Free Software has something to do with personal believes of Richard Stallman. Or the FSF. Or any-who. Freedom is important on a personal level. Richard Stallman is just a person that talk about Freedom.

`lbry://@blenderdumbass:f/its-not-about-richard-stallman:4`

A lot of people come to me and ask me whether something is okay from the point of view of Richard Stallman, or the FSF or the Free Software activists in general. People think that Free Software is something similar to a religion and there is no way to figure it out alone. But it's not true.

Free Software is about User's personal freedom to do what ever they so desire with their hardware. And software is usually the biggest hurtle on the path to such *personal* freedom. A fight for Free Software is a fight for us to be able to control our computers. And because computers are so important to our lives, to control our lives.

Richards Stallman doesn't ask you to switch all at ones

People talk about Richard Stallman as this angry dude who fights with people for choosing to run proprietary software. I heard about a man complaining about the Anti Cheat Malware support on GNU / Linux, who said that the only way he runs Free Software games is if Richard him self was watching him.

But contrary to popular believe Richard Stallman doesn't ask people to switch all the software to Free Software at ones. He knows that it would be a burden to some people. Instead he asks to switch some every ones in a while.

In this video on *20:22* Richard Stallman said:

> Now, when I talk about how hard I work to reject non-free software in my own life, some people get the idea hat I will despise them if they don't try just as hard. No. I understand you may be under various kinds of pressures. And it may look very difficult to say no. Especially if you already said yes to lots of these things for years. It would look like a big change. I didn't ever have to make a big change. Just when I get offered some additional nasty thing, I say "No thanks.". It's easy for me a little step at a time. So I don't demand people resist as much as I do. I hope you'll resist some. And if you resist some, I hope you'll start resisting more.

Controlling your computer, or having user freedom in a form of Free Software is important. And it's a fight that worth taking. But it's not freedom if you suddenly have to use Free Software.

This is a big problem with Free Software. We need people to resist non-free software enough, that companies will not see a potential of developing a proprietary program. But we don't want to take away your freedom to run non-free software.

And it's not even a question of convenience. If a presentation, or a marketing strategy could be done, the strategy should be better about the core idea. About Freedom. So even if the proprietary program is more convenient, there will be no value in it's convenience.

Personal control

I've got a response from the previous article that some people just don't have time to set everything up themselves. Thus they use proprietary software.

I think I finally have an answer to it. Software it self is developed to simplify a process to win some time. For example the script that I use to make the sponsored section. I could do that manually, go an look in the wallet for who supported what. I could manually craft the images and upload them to LBRY. And then manually write the part of the article. But it would take too long. While I can type a command and it will do it for me in a fraction of the time.

Software is there to do your computing. With Free Software people can make the computing simpler. If you want to write an article like me and do something similar. I could make my script proprietary. Then your articles would look the same way as mine. With the yellow border and round rectangle. And with the ridiculous Dyuthi font. But since it's Free Software, you can grab it, modify the theme and the text and use it for your own articles. Saving yourself time, while keeping your theme.

Yes, Apple's Operating System looks nice and works. But I would not be able to use it since it's not very customize-able. There was a meme, not sure if it was true or not, but seems close to reality. About an Apple credit card. To which they had a restriction not to put it into leather or denim, to prevent it from scratching.

Proprietary Software companies do not want you to be able to modify too much of their software.

Because it will go against the vision of the proprietor. There were no themes in Windows 7. Restriction on scratching with Apple Credit Card. They are too afraid about their image.

Me, I don't give a damn if you modify my program in such a way that it will result in ugly outputs. It's your problem, not mine.

Your Turn To Help Free Software!

We need 3 levels of presentation for Free Software to succeed:

- **Philosophical**. Telling people what it's all about. Fighting for their personal freedom. Not trying to make it seem like a game of them versus us. If tomorrow Microsoft or Apple releases a Free Program, it's a part of the victory.
- **Presentation**. People need to know that their computing could be done with tools that are not proprietary. And presenting those tool in the good light is crucial. We need people who can "sell" Free

Software programs to the average users. To liberate them.

- **Convenience / Personalizing**. We need good software, tailor-able for the users needs. Convenient, but most importantly Free. So the user could grab the program that's useful for them. And not be too lost inside of it. And if they need to modify it, the modification aspect should also be relatively straight forward.

For these tree things we need people with tree different skills. Skills of a speaker. Skills of an artist / presenter. Skills of a good programmer / UI designer.

If you are one of those people, you know what to do.

Happy Hacking!

Free Software Game Design

Let's talk about a potential game that can be done as Free Software, that will be epic.

`lbry://@blenderdumbass:f/FreeSoftware-Game-Design:3`

When I was about 3 years old, my dad brought me to a place where he used to work. I don't remember much of what was his job. But I know it was a place full of children. Perhaps it was a kindergarten or something similar. Maybe even an orphanage of a kind. I don't remember much of it, but I will try to recreate the situation.

I see a room that reminds me a big kitchen, but it was a weird room. Since kitchens don't usually look that way. I know a kitchen at my house. And we have it way smaller, and without so many rooms around it. I was looking at the kitchen, fascinated with it's size, or at least this is how I remember it. Then my father came from behind. "Come, I gonna show you something." he said. And I went with him.

There was a door to one of the rooms surrounding the kitchen. It was already open, but there was no light inside. Together with us, the kids that lived there, or that were there for any other reason, entered the room as well. There was no need for the light. The room, even though small, was filled with many bright computer screens. Kids sat in-front of them and launched various games. I remember seeing a colorful version of Tetris, that one of the kids played. I was already familiar with Tetris from the small hand held, cheap, gaming devices that were so popular in those days.

My dad took me up and put me in-front of a computer too. He gave me a mouse and told me to try and draw something with it. On the screen there was no game, but a session of Microsoft Paint. I pressed the button on the mouse and when I moved it around the table, it drew a line on the screen. I was playing with a program that is built to create things, while other were playing games for pure joy of it.

When we gone from that place, I was constantly thinking about this "computer" thing. And how I could do various interesting things with just that one program. Later I revisited my interests with being able to use more of Microsoft Paint in peoples homes and in school. Kids in school looked at me weird. From their perspective I wasted time painting instead of doing what "computers are built for" - Gaming.

Don't get me wrong, I was enjoying a fair share of gaming. I was exposed eventually to that as well. But my gradual understanding of computers started from a program that lets you create thing. Not a program designed to entertain. I saw games as a kind of good to have option, not a requirement. Other people thought about games first and anything else later.

Even my brother had his first experience with computers from a video game. And thus now, as we sit in the same room, I type this article in Emacs on a Free Software operating system, while he is playing a First Person Shooter on his Windows hard-drive. For him and millions of other people games and other entertainment comes first from a computer. And I can't blame them. It was their first experience with it.

A lot of people that freed themselves by switching an operating system from a proprietary one to Free, have started the same way as I did. They see computers as a kind of tool to create. And until proton started to give an option for people to play many games on GNU / Linux. People who would switch didn't see games as important. Switching didn't feel like giving anything away.

But now we have a problem. More and more people expect more and more games to simply run on their system of choice and while it's good that proton gives them more choice, they are not really freeing themselves. A few can enjoy playing only Free Software games. And while some "clones" like Minetest feel almost indistinguishable from the proprietary "original" people want the "original" more. And thus something has to be done about it.

A lot of people buy a computer to play a certain game. I and many of you would disagree with such people, but unfortunately it's the case. And the "gaming industry" took notice of it. There is no special purpose of having a gaming console in your house, if you already have a PC that can run most games flawlessly. But people still have gaming consoles. The reason is, there are games that are "worth playing" so much, that people will buy a separate computer for that game. And companies making those "separate computers" (consoles) make sure that those "worth playing" games would be not only available on their consoles. But available **only** on their consoles.

People see an advertisement of let's say "The Last Of Us" and they want to play it so much that buying a full console for this game is not a big problem. Especially if there are multiple such *exclusive* games on that one console. Nintendo does the same exact thing with Mario and Zelda. And it results in people having PC, PlayStation, XBox and Nintendo <put name here> all at ones.

A lot of us see the rise of the Steam Deck as the kind of addition to the console market that will free the users from the Proprietary systems of other gaming offerings. And while it's built on Arch and has the Linux kernel in it. It still run all the same Proprietary games. Only this time with an option to take it away with you on the walk. It's a step forward, but it's not the goal.

I want a future in which all software for all uses will be Free Software that respects the 4 essential freedoms. And games are not in anyway excluded from this future. So something has to be done.

Currently I'm working on my second big short film project called Moria's Race. I call them short films because they do not run for 90 minutes. But the previous one was a healthy 32 minutes long. I'm not sure about this one. But the script for Moria's Race is a good 50 pages long. I did all of the 3D assets and characters myself. So I think making a game project seriously could be a thing that I can do.

I already made a little game called J.U.M.P Limited that is download-able and playable for anyone who wants. And it's Free Software. Also I made a great amount of programming lately, starting from VCStudio, the famous FastLBRY terminal and smaller things like the script that I use to populate the Supported section. I think I can handle a reasonably large game project.

In this article I want to share with you my thoughts on a potential game project that I may take on when I will be finished with Moria's Race.

Open World?

Open World games are games in which the character is free to go into any direction at any moment. A game that comes to mind when thinking about Open World is Grand Theft Auto. All of the games in the series are Open World. With a whole city usually available to the player all at ones.

Sometimes they may deploy a technique that is designed to make the Open World feel larger. While the world is relatively small, as in the city is not very large. Making some part of it closed from the start of the game and open-able as you pass more of it, makes the discover-ability more enjoyable.

If you had the access to the entire world at ones, you will explore most of it immediately, and the world exploration will not make you feel any better. You will feel like it's too small for your, since you know all of it already. But making it so after a certain level a new part of the open world opens up for exploration, makes you feel like you play the open world.

In Need For Speed Most Wanted for example, the locked parts of the world, the two other towns, were open-able only after you beat in the race a certain boss. It would happen twice. Since there are 3 towns in the world. One of those towns will be explore-able from the start. Other pars would be locked with a semi-transparent wall that would literally not allow you to drive through to the other parts.

In Grant Theft Auto San Andreas the entire world is available from the start. Only the developers made an artificial restriction for the parts of the world that should still be considered closed. First they put traffic blocks on the roads that go to parts that are not explore-able yet. But these blocks are possible to overcome, either by water or by air. In which case, if the part is still closed, you will automatically get a 5 star criminal rating, so all the police and other forces will suddenly start hunting you.

There is no way to escape them since the 5 star will not get off you unless you drive back to the allowed location. And the police will sooner or later catch up with you. The developers of Grant Theft Auto made a better decision in my opinion. Since for one, it feels like the world is truly open. And in the same time, it gives you an interesting challenge. A challenge to stay alive as much as possible in the closed area of the world.

Other games like Minecraft or No Mans Sky may exploit a different open world technique what so ever. Instead of hard work of designers actually building a city for you to explore, they opted to use an algorithm that builds the world randomly. Creating unique worlds each time, but making them very monotone in nature.

In Free Software games, there are Open World games from both categories. There are games like Minetest (a clone of Minecraft) and Veloren that deploy a similar technique to Minecraft. With an open world being generated by the computer as you play it. From the other side there are games like Flare or FreeDroid that are clones of Diablo. Which have a basic 2D RPG style, but with a world being designed ahead of the time by artists.

I think it's reasonable to assume that in order for people to buy a GNU / Linux machine for a game, the game should be Open World. And it's also reasonable to assume that the world should be designed. People usually feel effort. And with games it's easy to showcase. Since the more stuff you put into a game, the more effort is perceived.

If the game is not open world, there should be a hell of good reason for it. Maybe it's a racing game. But then again, Need For Speed Most Wanted is a racing game and an open world game.

But there is a problem. You can think of SuperTuxKart as an open world game. The level choosing area is freely explore-able. And you can call the levels totally fine, since even in Grand Theft Auto San Andreas there were racing mini-games. And they were in it's own little worlds. Just like the tracks in SuperTuxKart.

But the problem is, the explore-able world is not fun. The tracks are way more fun then the explore-able stuff. And thus it doesn't have any feeling of a true open world game. In my opinion, if the game is open world, the exploration of this world should be an interesting thing to do. It should be fun. You should be finding things left by the developers to immerse you into the experience of playing this game more.

On the other hand we can do something that will resemble an open world game, but will be a lot simpler thing to pull off. If you look at a game like FlatOut 2 which is not open world, the designers still took effort into making it feel like it's open world. It's a racing game with multiple tracks existing on the same, larger map. Meaning that you play one track and recognize areas from another track on the background. If feels like it's all just one big world, while it's actually just a track. Where it's not even possible to drive off into that other part.

A similar technique is used in the original Unreal Gold in the extension called "Return to Na Pali". Where you play a simple level based game, but you see glimpses of both previous and next levels in the background of the level. Sometimes even including very distant objects that you will visit only in a couple of levels. This game is not open world by any stretch of imagination, but it feels like it is. You can see from level 1 the location on the level 3. And you can imagine a path towards it. And a path is there. Only it will be filled with enemies and cut scenes to transition to the next little piece of world.

Any reasonably sized level, either a whole open world or not, will still need to be broken up into LOD-able pieces. LOD or Level Of Details is a technique used in games to optimize them. You don't need to see the full resolution version of objects when they are in the distance and take a very small part of the screen. So for that any LOD'd object is prepared in multiple resolutions. The full resolution is visible only when you come right to it. As you go farther, the object becomes simpler. And when you are too far away, it turns either into a 2D picture of it self, or disappears completely.

I was thinking about making a level based sequel to J.U.M.P Limited using the new UPBGE engine which combines the Blender Game Engine (dropped by the Blender Foundation) and the new EEVEE engine. I already played with it and the graphical side of things looks promising. If it has the same underlying engine as the Blender Game Engine with which I'm familiar, I think something like a level based design with parts of different levels visible in the distance is more then possible. Combining it with a unique game idea, as in jumps being limited, will make for an interesting test.

But I still want to try and make a fully open world game. And I think I have just the right material for it. It would be nice if I could reuse, or purpose some of my older assets into a full game. Which is what I plan to do with Moria's Race assets. I already have a full city built for it. Maybe an extension will be needed. With more tracks and other areas to race in.

I want the game to be a sequel to Moria's Race since then I'm not obligated to follow the same plot. And it's going to be Dani's Race. This time the main character is Dani and he want to prove Moria that he also can win a race.

I think for him it's not going be as simple as for Moria. Since he will need to find way to reach pedals first. And it will be a unique gimmick of the game. Maybe a short race as Moria or something like this could be done in the beginning, to tease the player into what's to come. And then Dani will be given to you and you need to figure out how to make him reach pedals.

Then a full world of interesting people, street racing, character and vehicle customizations and other interesting things will be presented to the player. Finally you will open areas in which by the end you will open a racetrack. And you will have multiple big races. Probably either the first or the last track will be the racetrack on which Moria had won previously.

Story vs Game-play

Lately a lot of games and game developers are focusing on crafting interesting stories to fill up their game worlds. Some go as far as to create a near cinematic experience. Others use a very basic plot, for example a princess was captured so a plumber must save her.

There are no shortage of stories in Free Software games as well. From the penguin trying to save his loved penguin girl to a penguin trying to save GNU Stallman by winning a race versus the devil. But I rarely see Free Software games with a true narrative genius. The closest thing to it might be Never Alone by @OfficialZaney. And even that is not sure to be a 100% Free Software game. As I understand his problem, he has models that he can't share alone. Only as a part of a complete game. *(Maybe he will accept help in replacing them)*

Maybe Dani's Race could be such a game. But I'm afraid it's too whimsy to be considered a cinematic experience. Unless I figure out how to make this whimsy to tell a nice story in a pretty way.

From the other side, sometimes games are too cinematic. For example the infamous Need For Speed The Run was criticised for being too cinematic. Instead of giving people the game-play that they expected from a racing game, they got plenty of cut scenes. The game was trying too hard to be a movie. Not understanding that it's still a game about cars.

I think a good balance is done by games like the Grand Theft Auto series. They have a plot with cinematic cut scenes, but they also give you the freedom to do what ever the hell you want. And both are fun to engage in. Sometimes they combine the two. During missions you have voice over narrating the current state of the mission. Instead of showing you the progress bar for how far you've got. The cinematic-ness of the game is used instead.

I think this technique probably could be pulled off by Dani's Race. I want to have a few cut scenes. Like when a new areas is unlocked. Or when you meet new people. And the rest of it will be fully game-play.

Micro-transactions

So a little history lesson. In the early days, all software was Free. The source code was always available and people would pay for programmers and not for programs. Then the world was split in two. Hackers would form the Free Software movement. Others would make a ludicrous idea to restrict people in the name of profit. Using this new proprietary model, game developers started developing more ambitious games.

In the late 90s some developers got together and made a program called Napster. It wasn't Free Software, but it gave the hackers new ideas. Napster was a program made for people to share audio files with each other. And they started sharing music, making the music industry very unhappy. The music industry lobbied for new laws that made Napster illegal. Shutting down the service.

Later other services appeared. Some where as easy to shut down as Napster. Others like Torrent operate to this day. Allowing people to share files with each other.

As you can tell, nobody gave a crap about the laws put by the music industry and people kept sharing music, movies and video games with each other. Most of this sharing was happening outside of the US where laws are not as strong about harmless things like this. But companies were still feeling a metaphorical hole in their pockets.

First to implement in game purchases were Asian developers where sharing was the most prevalent. Creating a "Free to play" model. Basically, they would design a game in such a way that sharing was best for them. Since more people would join a little market inside a game that would provide the developers with the income.

Later the entire world jumped on the same bandwagon and started implementing in game purchases. Both in "Free to play" and Payed titles. Which to some people started to seems a bit wrong.

Later another gimmick had appeared, that explored the psychology of the player to squeeze from them even more money. Resulting in a real addictions of the players and even resulting in bans of such games in some countries. This gimmick is called

"Loot Boxes". Basically a player buys (or otherwise obtains) a box with an unknown content. And as the player opens the box, the contents of the box are generated randomly. Basically, the player doesn't know ahead of the time what they are buying. And this causes the same level of addiction as lottery or gambling.

Is it okay to implement Micro-Transactions in Free Software games? Well. I think it depends on what you mean by that. This article is hosted on the LBRY protocol. And there are publications on this protocol that cost money. You get to use the protocol gratis but sometimes to watch a video or to download a file, you may pay for it. This is a micro-transaction in a Free Software platform.

You can think of a game as a platform too. Let's take for example the Dani's Race game. What if a user could purchase more skins into the game. More cars, more clothes, more mods, more racetracks, more towns. Anyone can put an extension into the game, since it's Free Software, it's allowed by default. So I think what could be done is this.

We could make a special tag / file format for the game data and make the extensions available through the LBRY. And anyone can upload them with any price. This will make a competitive market in which best items could cost more. And thus if we put some of our own extensions, we will have to compete on the same market. Which will force us to do better job then anyone else.

SuperTuxKart has a library of extensions that anyone can make. But more often then not those extensions are crap. Now, if you give people a real incentive to put in the work, there might actually be good extensions made by the users. Especially if they are playing the same game on the same conditions as the developers.

Race-cars, car parts, pieces of clothing, characters, plot lines... Anything really can be done extendable. Maybe with a good enough design we can make a game with the same level of extend-ability as Emacs. Think about an Emacs type game. But which is also a market.

Loot boxes are not cool in my opinion. But if people want them in our game, they can fork one into it. It will buy them an item on the Dani's Race market randomly. Maybe it could be an option. I'd like to simply browse through the game's market and see what is being offered.

The Plan

So now I gonna tell you what is my current plan on this whole Game making idea.

- **Finish Moria's Race**. First I will need to finish Moria's Race because making multiple projects at ones is too tiresome for my mind. And think about making a movie and a game in the same time. I will be the first person dead from migraines, if I tried.
- **FreeGILE**. I will need to make the FreeGILE project work fully. Since it's going to be the market place / download app where I gonna publish my next games.
- **J.U.M.P Limited II**. (The Test). On J.U.M.P Limited II I will figure out the limitations of the engine. I will figure out ways to overcome these limitations. I will figure out how make graphics that works, with LODs and other stuff not required for films. As well as I will brush myself in programming for games.

- **Dani's Race**. And finally I will start a game project that should come out just right. It should be epic, have a story, have an open world, have extendable items and be ultimately fun. Fun enough that I wish it will bring people from outside.

  You may think that it will not work since there will be versions of this games working on windows. Especially if it's UPBGE. But I think with enough smart design I can sneak into it just enough GNU / Linux code making it a mess to port to windows. It will still be possible, since it's Free Software. But it will require work. And until it will be done, maybe I can succeed into having a few people come to the free world.

Your Turn To Help Free Software!

I think I may ask you to help me. Please research. I will take almost a year to finish Moria's Race at my current pace and it's good. Since I will have time to plan everything for the games. But a help from the side may be required.

If you can, write essays, like this one, about what you personally prefer to have in a game and how would you like it to be realized. And link them to me.

Also a great help would be to research how games that are popular became so popular. Is it in the good graphics or polished animations, or something to do with freedom of movement more?

If you have time research about Free Software game engines and differences between them. I want to go with UPBGE since I'm familiar with Blender and it seems like the right choice. I mean Yo Frankie! looks hella nice for it's year and for being Free Software. And it was done using the old BGE. Now with the new EEVEE engine, I think we can make a masterpiece.

Thank you for help in advance.

Happy Hacking!

This is the end of the Volume one. Why? Because I ran out of articles to put here. I will make another one as soon as I will get enough articles.

Meanwhile you can go to any LBRY client either FastLBRY, LBRY Desktop or Odysee.com and go to:

lbry://@blenderdumbass:f

Also you can fire up any matrix client of your choosing and chat with me and other hackers in:

#BlenderDumbassChat:matrix.org

Happy Hacking!

This book is your road to Freedom.