

On MIT, and addressing other licenses: The L^AT_EX way

Thomas Echevarria

November 12, 2025

1 Abstract

I write this in order to address questions and concerns about my choice of using the MIT licensing for the ™Darwine project. The following contents will be used to explain my reasoning and philosophy behind many licenses and my personal reasons as to why I do not use them. Do keep in mind that although I do not use these licenses, it doesn't automatically mean I am strictly against endorsing them in anyway unless explicitly stated otherwise. I hope this billet-doux suffices as a answer to many of the questions. Also, I do not mention every single license here. I simply tackle on the common and (hopefully) the one's that the general public are most familiar with.

As a side-note, since people apparently can't read, I will also state (Ironically, in writing) that any works that I reference here are NOT mine unless given a ™Trademarked status. Is this the conventional way of doing things? No. But it suffices. Any intellectual properties mentioned do have their own rights perserved.

Contents

1 Abstract	1
2 On The MIT License	2
2.1 What exactly is the MIT License and how does this compare with GPL?	2
2.2 Why MIT over GPL?	2
3 In the beginning, I address: Arguments!	3
3.1 “GPL is in full support of open source”	3
3.2 “What about companies that try and use your software as spy- ware, don’t you take accountability for that?”	4
3.3 “If you use GPL2, wouldn’t you be able to just sue companies that inappropriately use your works?”	4

2 On The MIT License

2.1 What exactly is the MIT License and how does this compare with GPL?

To put it simply, *The MIT License - Open Source Initiative* [3] states that you can take my codebase and use it however way you want. What exactly does that mean in the world of ambiguity? Well, it explicitly state's that you can use my code to redistribute, modify, sell, essentially do whatever you want with it. You can even claim it as your own and face no consequences. You don't even have to give credit! You can take the code and modify it to your heart's desire. Compare this with GPL2 which states that you can use the code in however way you want *under an exception*. And the exception is: you must also license your project under the same *GPL-2.0* [2] License. This mean's that your project also has to be free and open source and give rights to the original creator. On the surface, this sounds like an ideal. Why would anybody want their code just taken and used in whatever way without their knowledge? This brings me to the main crux of my argument.

2.2 Why MIT over GPL?

As stated in the MIT Licensing, you may modify, copy, redistribute, essentially do whatever you want with my codebase. And, you don't even have to give credit. That means no special gold plate with my name stamped on every computer saying "Good Work!". Some would like that, which is completely okay. But I do not, nor is that the intent with the TMDarwine project. My software is meant to be used as tools and things in which you can accomplish a specific task, not nothing more than a excuse to add attention to a name. Why not? Quite frankly, it's just not something that I truly care about. These are tools and if people are using the code that's in them as part of a means to an end, I think that's great. It's serving the purpose. Technology should never rule over another's intellectual property. They shouldn't rule over a person, so why should it rule over their works? Free-software is not saying: "Okay, you can use me but only under my terms". free-software that maintains its own name states: "Do whatever you want". 'Free-software' as defined by GPL2 is just not democratic. You ultimately have no say in how you want your work should be used (which by 'work', I mean your own intellectual property). But the one who does get the say, is good-old Richard Stallman. And you effectively don't actually truly own your work. That's not freedom, that's authoritarianism.

As *The MIT License - Open Source Initiative* [3] states:

"permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the software), to deal in the software without restriction, including

without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the software, and to permit persons to whom the software is furnished to do so, subject to the following conditions: the above copyright notice and this permission notice shall be included in all copies or substantial portions of the software. the software is provided as is, without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. in no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.”

I won't state the entire GPL2 license as it's way too big to copy and paste but I will *provide a link: GPL-2.0* [2]

3 In the beginning, I address: Arguments!

3.1 “GPL is in full support of open source”

GPL2 simply doesn't fit most of the project's requirements, nor my philosophy and general vision of what the free software community should look like. GPL2 advocates for a "Do whatever you like, but make sure it's under my terms". At the crux, this is what most licenses are like (with the obvious exceptions like EULA and other non-free software licenses), but this is especially what GPL2 is. In the license, you are free to copy, modify, redistribute, reuse, in however way you want. However, this can only be done if the project is also correspondent under the GPL2 license. The reasoning behind this, is simply to ensure that every project that uses it remains free, open-source, and ultimately transparent at the codebase level. On the surface, this sounds like a great idea. This seems like this can put a stop to any malicious big corporations that want to steal your code and subvert it's direction to being nothing more than spyware. However, this gets a bit more nuanced when libraries are suddenly thrown into the license. License's like these take away the property's rights to license their own work (which is more than likely, the majority of the codebase) in anyway they want. Although, I am heavily against corporations using another developer's source code for malicious intent, I think fairness and rights extend even to anybody who owns their own property. Having very micro and individualized components effectively dictate the entire licensing for a codebase is inherently bad. This is besides the many practical complications you'll run into. What if you want to use a library that is under GPL2 and also want to use another library which is under a similar license that also require's that you use their license. This especially gets worst when frameworks make up %30 of the project (Which is way more, these days). This license works best under entire project's that are far more than just small components, but ultimately don't fit ™Darwine.

3.2 “What about companies that try and use your software as spyware, don’t you take accountability for that?”

No. Just simply, not true. Corporations are run by their own individual person(s) involved. This mean’s that each and every action that is done on their end is simply just that, on their end. There is no reason for you to take responsibility for changes and actions that were done by a different party. This is like shaming a person for what their ancestors did 200 years ago because they have the same bloodline. It’s silly, childish, and utterly embarrassing. Not only does this argument not belong in the basics of logical reasoning, it also make’s sense of a legal standpoint. Companies are legally acknowledged as people (Take that for what you will), and as such, you legally don’t have a responsibility for their choices (See *Wikipedia - Corporate Personhood* [4] and *court-cases: Citizens-united-v* [1]. This is besides acknowledging the established permissions and rights in the MIT License.

3.3 “If you use GPL2, wouldn’t you be able to just sue companies that inappropriately use your works?”

If on the rare occassion that nintendo uses your little API for their shitty console, and they violate the GPL license being withholding the project and keeping it as EULA, you have every legal right to fight against that. That doesn’t mean that you’ll win. This is just a dose of pessimism, but a ”Yeah but, they just violated GPL2” just simply won’t fly in court. Not especially against corporations (such as nintendo) who have billions of dollars and a giant law firm. It just won’t work. And they are plenty of examples of corporations using software with similar Licenses, meanwhile they face zero reprocussions.. I wonder why.

4 I saw that it was good..

Anywho, my general point is this: Free software should be as is without any strings attached. The puppet is the software, and the developer controls it. The strings shouldn’t extend over to somebody else’s project. MIT is and has been a great and suffice license to satisfy the ‘permissible’ requirements that I seek in the ™Darwine Project.

References

- [1] *court-cases: Citizens-united-v*. <https://www.fec.gov/legal-resources/court-cases/citizens-united-v-fec/>.
- [2] *GPL-2.0*. <https://www.gnu.org/licenses/old-licenses/gpl-2.0.txt>.

- [3] *The MIT License* - Open Source Initiative. <https://opensource.org/licenses/mit>.
- [4] *Wikipedia - Corporate Personhood*. https://en.wikipedia.org/wiki/Corporate_personhood.