On Open Source in Guncad

Table of Contents

Introduction

The 3D printed firearms community was intentionally founded on the principles of free and open source software (FOSS[1]). This ethos, adopted over a decade ago, is still widely practiced in our community today.

But what is FOSS? The FOSS ethos was proposed, debated, refined and crystalized nearly 40 years before guncad adopted it. This refinement defined four fundamental freedoms[2] at the core of the FOSS ethos:

- Freedom to use software as you see fit, free of use restrictions or enforced guidelines.
- Freedom to explore, investigate and test software, free of obfuscation or requests for permission.
- Freedom to modify and remix software, free of any intellectual property claims
- Freedom to repost, repackage and resell software, free of legal NDAs, anti-share agreements and anti-commercial dogma.

These freedoms were originally conceived as *legal* freedoms, enforced via copyleft licenses such as the GPL. When software is published under these licenses, the freedoms listed above become legally inalienable from the software and its derivatives, and any publishers of derivative softwares are

With the wider adoption of the FOSS ethos across the software industry, its second role has emerged in the establishment of norms to guide software communities. Instead of living as mere defensive measures taken against corporate “closure” of free software, FOSS in its normative role helps direct and correct the behavior of open source software developers.

The four freedoms remind us how to act. Their behavioral role is as important as FOSS' legal role, if not more so. Communities that claim to practice open source frequently exhibit hoarding behaviors, intellectual property ideology and covert methods of censorship. FOSS norms help these communities to return to proper behavior.

Freedom to Use

The Freedom to Use software means that a user must be able to operate the software in any way he sees fit. This idea developed in an era of frequent "conditions of use" placed on software, e.g., a video player that might disallow users from playing copyrighted videos. A better example for guncad may be a slicer which disallows slicing firearm-related files.

Non-FOSS software can violate this freedom both technically (by literally disallowing[3] an action) and legally (even if the action is technically possible, it violates the software’s terms of use).

In its legal sense, Freedom to Use insists that a developer should not release a 3D printed gun project under terms that disallow certain actions being taken with the project. While this does not generally occur in the guncad community, it is still conceivable – we can imagine a bizarre “pacifist” species of file release that insists any user of the design agrees not to actually print it.

As a behavioral norm, Freedom to Use implies that open source developers should act in a broadly "permissionless" manner. This means that they should not insist that others ask permission to do things like repost or remix their file, or to do any other thing that a user has a right to do. A developer respecting the “permissionless” Freedom to Use says: *I bequeath my project to you. Do with it what you will.*[4]

More broadly, Freedom to Use as a behavioral norm insists on a definite attitude towards users in the guncad community: community members should be assumed to be, and treated as, mature adults capable of making decisions and taking risks. They should not be treated as immature or in need of supervision.

This violation has its opposite angle as well. Are ideas (or developers) scrutinized and questioned, or are they unthinkingly accepted based on mere popularity? Treating fellow developers as adults means not only respecting them, but insisting they conform to our community's standards.

Respectful, responsible treatment is proper FOSS behavior, and improves the health of our community. Tribalistic, elitist or sycophantic behavior is anti-FOSS, harms the community and keeps it small.

Those who treat their fellows as adults and act permissionlessly are proper practitioners of the Freedom to Use.

Freedom to Explore

The Freedom to Explore refers to the idea that a software developer should be able to explore a software's underlying architecture. Related to the Freedom to Use, this freedom entails being able to edit a program, test it, and - most importantly - to read the program's source code. It is commonly noted in FOSS texts that this freedom does not *strictly* require access to the source code, but as a practical matter such access is necessary for this freedom to mean anything.

The historical context of the Freedom to Explore is the desire that software releases be accompanied by an accessible copy of their source code. Additionally, a program should *not* include legal terms forbidding the reading, compiling or testing of that source code.

As a behavioral norm for our community, Freedom to Explore asks several things of us. Most obviously it asks that we, too, release the guncad equivalent of source code: STEP files, or an equivalent CAD file.

Fortunately, most developers release their STEPs upon publication. There are those, however, who do not. A variety of justifications are offered for this refusal:

- The general desire to retain control over a design
- Fear of design theft
- Fear of the STEP facilitating unsafe modifications, which might cause harm to the modifier, or harm to the reputation of the original designer

These justifications, while understandable, are contrary to FOSS. The first two represent an improper attitude towards one's designs (an open source design cannot be "stolen") and the last indicates an

We can go yet further. There is a broader sense of Freedom to Explore which asks us not just to open our source, but to also open our development process. This means the appropriate integration of common software development practices, including:

- Publishing the first minimally functional version as soon as possible, followed by frequent, subsequent updates
- Comfort with releasing work deemed “incomplete”, given appropriate warnings and provisos
- Exposing work to as many eyeballs as possible

ESR teaches us[5] that this “bazaar” style of development is both more philosophically compatible with open source and tends to produce better software, as opposed to a corporate or “cathedral” development style.

Guncad development differs from general software development in critical ways, so we must adapt “bazaar” practices accordingly. For example, the consequences of catastrophic software failure are generally less severe than those of catastrophic firearm failures. This may justify raising the safety standards or duty of care that a project must meet before its designers open its development cycle to the world.

That said, if we overvalue concerns of safety or reputation so much that we fall back into a “cathedral” style of development, keeping our work closed to the very point of release, we have gone too far. This is the anti-FOSS cure that kills the patient, resulting in endless closed, secret projects. Remaining siloed, we observe that the final products are irrelevant upon release, long bypassed by open projects which benefited from the attention and energy of hundreds of eyeballs.

The Freedom to Explore helps us remember that what truly matters is not reputation but released work - STEPs and all - that fellow developers can learn from, extend and remix.

Freedom to Modify

The Freedom to Modify extends from the Freedom to Explore. If a developer has a right to access the source code of a program, then should he not also have the right to edit that source code so that the program can better serve his needs?

Historically, this freedom was challenged by terms of use which forbade anyone except the original software developer from editing the code. Stallman tells a story[6] of wishing to improve the print modify the code, leaving users entirely at their mercy.

Legally, the Freedom to Modify ensures that developers cannot be stopped from modifying software. Technically, this freedom only applies to modification *for one's own use*, but the Freedom to Share, covered next, also ensures that modified versions can be distributed.

The belief that an idea, or its realization in a product, can be controlled by an author once it has left his hands is known as *intellectual property* (IP). This ideology is rejected wholeheartedly by FOSS, which asks developers to disclaim any intellectual property interest in their designs. Rejecting IP and its ideology ensures that no developer fears being sued for modifying (or sharing, etc.) someone's work.

Of course, even if a developer does not *legally* assert IP, he can still act in ways that imply a property interest in his work. Freedom to Modify, understood as a behavioral norm, encourages communities to also avoid these "intellectual property-ish" behaviors.

Examples of IP-ish behaviors:

- Insisting that developers seek permission before remixing a design
- Attempting to control what sorts of remixes are acceptable for a design
- Claiming that an "unauthorized" remix is "theft"

We may mistake this behavior in guncad for matters concerning accreditation (or the lack thereof). Proper accreditation *is* a requirement of FOSS norms, but it is there that the responsibility ends. Provided credit is maintained, Freedom to Modify tells us that no further permission is needed.

The problematic IP-ish behavior usually found in our community often concerns not just ownership of a particular *file*, but ownership of a particular *idea*. This can include:

- Claiming that another "stole" the idea for a design and used it in their own work
- Claiming ownership over a particular practice or method (of testing, etc.)
- Claiming that you alone have the right to work on a particular idea (because, for example, you started the project first)

If claiming IP in a realized open-source design is bad, then claiming IP in an idea (realized or unrealized) is even worse. The concept that ideas cannot be owned is a fundamental belief in FOSS. Proper understanding of FOSS dissuades us from this behavior.

ESR notes[7] that despite IP-ish behavior being one of the most blatant violations of FOSS ethics, it is also one of the most common. Open source communities have long practiced this contradictory behavior despite themselves. Though the communities of which ESR was a member in the '90s could remain stable in spite of this contradiction, this is not obviously true for guncad culture.

We find the majority of conflict in our community can be traced to IP-ish behaviors: arguments over original ownership of an idea, or whether someone has the right to work on a project or concept to which another developer lays claim.

If we wish to practice FOSS, then we will protest this behavior. An idea cannot be owned, and so cannot be "stolen"; neither does a developer have an exclusive claim to an idea simply because he came to it first. If you wish to "protect" your work, our legal system affords you copyright and licenses, but you cannot claim to be practicing FOSS when using these.

Practicing the Freedom to Modify means rejecting intellectual property behaviors and ideology, encouraging remixes of your work and embracing fellow developers working in similar spaces.

Freedom to Share

If you own something, then you must be able to transfer it to someone else, under whatever terms to which you both agree. This basic intuition is, in essence, the Freedom to Share.

The Freedom to Share as a legal concept ensured that free software could be given away, reposted, shared, and sold without liability. This was meant to ensure that users could, among other things:

- Give a copy of a piece of owned software to a friend
- Rehost a copy of software somewhere else (e.g., for marketing or preservation)
- Create collections of software for sale
- Release a modified version of someone else's software

…all without legal penalty

Anti-FOSS arguments against the Freedom to Share use terms like "piracy" and "theft" to describe this behavior. The classic rejoinder is to note that, with software, no injury occurs in sharing / reposting / reselling. The original software does not disappear when copied, and so the original owner experiences no loss in the creation of more owners.

- Attempts to prevent designs from being reposted
- Attempts to prevent designs from being sold

The first objection is easy to counter. Why would someone ask that their design not be reposted? If some justifications are sympathetic (a designer fears that if a design can be reposted anywhere, outdated versions will accrue), some are certainly not (e.g., egoistic dislike of another community or organization).

As with the sympathetic objections to the other Freedoms, we note that the putative concerns here can and should be addressed without violating fundamental FOSS ethics.[8] In any case, we maintain that, especially for a community under threat as ours is, files should be reposted as widely as possible, on every platform.

The second violation, when properly understood, is more egregious. A common objection is: isn't selling files anti-FOSS? Doesn't the "F" stand for "free"?

This objection betrays a total misunderstanding of FOSS. Those taken to this misunderstanding need not be embarrassed, however, as this is one of the most common mistakes made in learning FOSS philosophy. In any introduction to FOSS we find the oft-made distinction between *libre* - free as in lack of constraints - and *gratis* - free as in lack of cost.

The "free" in FOSS always refers to *libre*. There is no open source tradition which claims otherwise, and every FOSS license, even those labeled "noncommercial", explicitly allow the selling of software. Stallman himself, likely the most anti-commercial of the major FOSS philosophers, repeatedly notes[9] that file sales are not just ethically acceptable, but are *necessary* for a healthy FOSS community.

How can this be? Should we not credit the explosive growth of the guncad community to the fact that developers are not forced to pay to access files? We can, but this is due not to the fact that most of the files are *gratis*, *but because they are released under FOSS licenses*. If a file is released for *gratis* in one location, then making it available for sale elsewhere does absolutely no harm. Similarly, if a file is released under an open-source license, but is initially only made available for a fee, then anyone who acquires it risks no legal or ethical harm by reposting it anywhere else.

Proper understanding of the Freedom to Share leads to a voluntary "surrender of monopoly"[10] over your files. This means accepting, and embracing, whatever methods of distribution the community sees fit to practice.

So ends a restatement of our laws. The proper adoption of FOSS norms requires a level of intellectual rigor and emotional maturity that does not always come naturally. It is also the case our laws *are contextual*, and if they cannot be broken they may still be bent. This bending, however, must start from an acceptance of our original position. It is the deviation from our ethos that must be justified, not the ethos itself.

In general, the guncad community practices FOSS well. As mentioned above, it is this practice that has allowed the community to flourish as it has over the last decade, in the face of overwhelming odds and enemy action.

Yet all communities require reminders and correction. This restatement is that reminder. To those in the community who fail to practice FOSS - to those who claim ownership of ideas, and who threaten intellectual property lawsuits, to those who rail against individuals exercising their right to use and share files as they see fit - we suggest taking the above lessons to heart. Commit them to memory. If, after due consideration, you find our laws wanting, the only loss is your claim to the practices of free and open source software.

DEFCAD will always enforce and protect these freedoms. To all FOSS developers, those who are and those who are to be, we leave you with the great reminder:

Free Men Don't Ask.

---

[1]. A longstanding controversy exists over the most appropriate name for this philosophy. Taking no sides in this controversy, we will use the neutral term "FOSS" for this statement. Conveniently, this happens to be the term that most of the guncad community itself uses.

[2]. Stallman, R. Free Software Definition. Originally sourced in Free Software, Free Society, 2002. An updated version is available online at https://www.gnu.org/philosophy/free-sw.en.html

[3]. https://arstechnica.com/information-technology/2013/06/worried-about-accidentally-3d-printing-a-gun-new-software-will-prevent-it/

[4]. Or, *come and take it.*

[5]. Raymond, E. The Cathedral and the Bazaar. http://www.catb.org/~esr/writings/cathedral-bazaar/

[6]. Stallman, R. Free Software: Freedom and Cooperation. https://www.gnu.org/philosophy/rms-nyu-2001-transcript.en.html

[7]. Raymond, E. Homesteading the Noosphere. http://www.catb.org/~esr/writings/homesteading/homesteading/

[8]. Consider, for example, creating metadata / documentation standards by which obsolete fileholders can locate newer versions of the files.

[9]. Stallman, R. Selling Free Software. https://www.gnu.org/philosophy/selling.en.html