

Fall 2020

Super Mario Decompiled

Joseph Godfrey

Follow this and additional works at: https://repository.uchastings.edu/hastings_science_technology_law_journal



Part of the [Science and Technology Law Commons](#)

Recommended Citation

Joseph Godfrey, *Super Mario Decompiled*, 12 HASTINGS SCI. & TECH. L.J. 1 (2020).

Available at: https://repository.uchastings.edu/hastings_science_technology_law_journal/vol12/iss1/3

This Article is brought to you for free and open access by the Law Journals at UC Hastings Scholarship Repository. It has been accepted for inclusion in Hastings Science and Technology Law Journal by an authorized editor of UC Hastings Scholarship Repository. For more information, please contact wangangela@uchastings.edu.

Super Mario Decompiled

by JOSEPH GODFREY*

TABLE OF CONTENTS

Abstract.....	1
I. Introduction	2
II. Reverse Engineering Law	3
III. Super Mario 64 Decompiled.....	6
A. The Purpose and Character of the Use.....	6
B. The Nature of the Copyrighted Work.....	7
C. The Amount and Substantiality of the Portion Used.....	8
D. Effect of the Use upon the Potential Market.....	8
E. Other Considerations.....	10
F. Precompiled Modifications.....	11
IV. Conclusion	11

* J.D., University of Derby, First Class Honours, 2019.

ABSTRACT

Super Mario 64 was a video game released to critical acclaim in 1996. 24 years later, a group of dedicated fans reverse engineered the game and released reconstructed source code publicly on GitHub. Soon afterwards, an unofficial PC version of the game began circulating the internet. The decompilation project presents an opportunity to examine the legal status of reverse engineering under US Copyright law, and whether publicly releasing reverse engineered code is a fair use.

I. INTRODUCTION

Super Mario 64 was a launch title for the Nintendo 64 video game console, released to critical acclaim on June 23, 1996, in Japan and September 26, 1996, in North America. Despite the game's age, you can still buy a copy (albeit digitally and for a different console) from Nintendo today.¹

The game was written almost entirely in C and was compiled using a Silicon Graphics IDO compiler.² The compiled code was written to the read-only memory (ROM) chips contained in *Super Mario 64* cartridges.

Over two decades later, a group of fans took on the task to reverse engineer *Super Mario 64* and rewrite it in C to better understand the game's code. The process involved procuring a copy of the ROM, decompiling the ROM to assembly code, and painstakingly converting the assembly code back to C by hand. The newly reconstructed C code was capable of producing an exact replica of the original assembly code when compiled with the same Silicon Graphics compiler.³ The decompiled code was released September 2019, absent any non-coding assets (like character models and sound files) which users would have to source themselves.⁴

In May 2020, a *Super Mario 64* version that ran natively on PCs (likely derived from the source code) began circulating the internet. Within a few days of its release, Nintendo began taking action against sites hosting it.⁵

The law concerning the reverse engineering of software is fuzzy, to say the least. This essay aims to make sense of the current law and reach some kind of conclusion regarding the legality of this *Super Mario 64* situation. Fundamentally, the issues at hand are ones of fair use.

¹ NINTENDO, *Wii U Super Mario 64*, GAME STORE, <https://www.nintendo.com/games/detail/super-mario-64-wii-u/> (last visited Sept. 24, 2020).

² C is a programming language that is relatively easy to work with, compared to low-level languages like assembly, which lack a large number of pre-defined functions. A compiler takes that code and converts it to a set of instructions a machine can understand. See *Sega Enterprises Ltd. v. Accolade, Inc.*, 997 F.2d 1510, 1512 N.2 (9th Cir.1992); See BRIAN W KERNIGHAN & DENNIS M RITCHIE, *THE C PROGRAMMING LANGUAGE* 8 (2nd ed. 2006)

³ Kyle Orland, *Beyond emulation: The Massive Effort to Reverse-Engineer N64 Source Code*, ARS TECHNICA (May 6, 2020, 9:33 AM), <https://arstechnica.com/gaming/2020/05/beyond-emulation-the-massive-effort-to-reverse-engineer-n64-source-code/>.

⁴ GITHUB, INC., *n64decomp/sm64: A Super Mario 64 Decompilation*, <https://github.com/n64decomp/sm64> (last visited Sept. 24, 2020).

⁵ Andy Maxwell, *Nintendo Lawyers File Copyright Complaints Against Super Mario 64 PC Port*, TORRENTFREAK (May 7, 2020), <https://torrentfreak.com/nintendo-lawyers-file-copyright-complaints-against-super-mario-64-pc-port-200508/>.

II. REVERSE ENGINEERING LAW

Reverse engineering is awkward because there is very little law on it. There have been no Supreme Court cases on the matter. Accordingly, most law comes from the small number of cases that directly deal with reverse engineering, and extrapolation of more general copyright principles.

Under section 106 of the 1976 Copyright Act, copyright holders have the exclusive rights to authorize a number of activities, most notably:

- to reproduce the copyrighted work;
- to prepare derivative works based upon the copyrighted work;
- to distribute copies . . . of the copyrighted work to the public by sale or other transfer of ownership, or by rental, lease, or lending⁶

Exercising any of the rights afforded in section 106 without permission is copyright infringement. These rights, however, are not unlimited and are subject to several exceptions contained in sections 107 through 122 of the Copyright Act. Most relevant to reverse engineering are sections 107 and 117; with the latter providing that software owners may make copies necessary for utilization (i.e. loading a program) and for archival purposes.⁷

Section 107 serves as a codification of the fair use doctrine, which allows the use of copyrighted material to be non-infringing if used “for purposes such as criticism, comment, news reporting, teaching . . ., scholarship, or research.”⁸ There are four statutory factors to consider when evaluating fair use. These factors are “illustrative and not limitative,” and should not “be treated in isolation . . . [a]ll are to be explored, and the results weighed together, in light of the purposes of copyright.”⁹ The four factors are:

1. the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes;
2. the nature of the copyrighted work;
3. the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and
4. the effect of the use upon the potential market for or value of the copyrighted work.¹⁰

In 1998 the Digital Millennium Copyright Act (DMCA) amended US copyright law, complicating the status of reverse engineering. Section 1201 of the Copyright Act¹¹

⁶ 17 U.S.C. § 106 (2018).

⁷ *Id.* at § 117(a)(1).

⁸ *Id.* at § 107.

⁹ *Campbell v. Acuff-Rose Music Inc.*, 510 U.S. 569, 587 (1994).

¹⁰ 17 U.S.C. § 107 (2018).

¹¹ *See id.* at § 1201.

prohibits the “circumvent[ion of] a technological measure that effectively controls access to a work.” The wording of this is vague, and it has been suggested that it may apply to more than just copy protection.¹² Subsection F provides an exception for reverse engineering, wherein circumventing a technological measure is permissible to achieve interoperability (cross-compatibility of software). For the sake of simplicity, this article assumes this DMCA provision only applies to breaking copy protection.

Outside the remit of the DMCA, as a general rule, the reverse engineering of software in and of itself is a fair use when done to understand non-visible concepts. Most legal issues depend on what is done with the reverse engineered code.

*Sega Enterprises Ltd. v. Accolade, Inc.*¹³ held that the disassembly of object code is a fair use when used to understand elements “that are not visible to the user when operating—and then only when no alternative means of gaining an understanding of those ideas and functional concepts exists.”¹⁴ The later case of *Sony Computer Entertainment, Inc. v. Connectix Corp.*¹⁵ builds on this idea, holding that “the fair use doctrine preserves public access to the ideas and functional elements embedded in copyrighted computer software programs,”¹⁶ and that the number of times software is copied during the disassembly process is largely irrelevant.¹⁷

The court in *Atari Games Corp. v. Nintendo of America Inc.*¹⁸ was very careful to protect reverse engineering when finding against the party that engaged in it, noting that “[r]everse engineering, untainted by [a] purloined copy of the [] program and necessary to understand [the program], is a fair use. An individual cannot . . . understand[] the object code on [a] chip without reverse engineering.”¹⁹ The court stressed that “[a]n author cannot acquire patent-like protection by putting an idea . . . in an unintelligible format and asserting copyright infringement against those who try to understand that idea.”²⁰

It may also be worthwhile to examine the effects of anti-reverse engineering clauses in software licenses (or shrink wrap contracts). Unhelpfully, this is the subject of a circuit split. In *Vault Corp. v. Quaid Software Ltd.*²¹ the Fifth circuit held a state law that enabled prohibiting reverse engineering to be preempted by the Copyright Act.²² Conversely, the Federal Circuit in *Bowers v. Baystate Tech., Inc.*²³ held, under First Circuit law, that the Copyright Act does not preempt such terms.²⁴

As mentioned earlier, most legal issues will arise after reverse engineering has taken place, where it is a matter of comparing the two works to determine if there is infringement. To determine this, both copyright ownership and evidence of copying must be proven. As

¹² *Coders’ Rights Project Reverse Engineering FAQ*, ELECTRONIC FRONTIER FOUNDATION, <https://www.eff.org/issues/coders/reverse-engineering-faq>.

¹³ *Sega Enterprises Ltd. v. Accolade, Inc.*, 997 F.2d 1510, 1512 N.2 (9th Cir.1992)

¹⁴ *Accolade*, 977 F.2d 1510 at 1520.

¹⁵ *Sony Computer Entertainment, Inc. v. Connectix Corp.*, 203 F.3d 596 (9th Cir. 2000).

¹⁶ *Connectix Corp.*, 203 F.3d 596 at 603.

¹⁷ *Connectix Corp.*, 203 F.3d 596 at 605.

¹⁸ *Atari Games Corp. v. Nintendo of America Inc.*, 975 F.3d 832 (Fed. Cir. 1992).

¹⁹ *Atari*, 975 F.3d 832 at 843.

²⁰ *Atari*, 975 F.3d 832 at 842.

²¹ *Vault Corp. v. Quaid Software Ltd.*, 847 F.2d 255 (5th. Cir. 1988).

²² *Vault Corp.*, 847 F.2d 255 at 270.

²³ *Bowers v. Baystate Technologies, Inc.*, 320 F.3d 1317 (Fed. Cir. 2003).

²⁴ *Bowers*, 320 F.3d 1317 at 1323.

direct evidence of copying is often unavailable, copying may be inferred where the two works are substantially similar.²⁵

Substantial similarity in reverse engineering cases is often established when more code than necessary is copied. For instance, in *E.F. Johnson Co. v. Uniden Corp. of America*,²⁶ a 'barker code' had to be copied to achieve compatibility.²⁷ However, (among other similarities) three unnecessary lines of code were found in the same place in both programs.²⁸ The court noted that "[t]he existence of the identical unnecessary instructions in both codes is strong proof of substantial similarity."²⁹ Conversely, in *Vault Corp. v. Quaid Software Ltd.*³⁰ just 30 characters of source code were copied for functionality. This was found to be neither quantitatively nor qualitatively substantial enough to constitute copying.³¹

Of course, if copyright infringement is found, it may be justifiable under the fair use doctrine.

²⁵ Atari, Inc. v. N. Am. Philips Consumer Elecs. Corp., 672 F.2d 607, 614 (7th Cir. 1982).

²⁶ E.F. Johnson Co. v. Uniden Corp. of America, 623 F. SUPP. 1485, 1494 (D. Minn. 1985).

²⁷ Uniden Corp., 623 F. SUPP. 1485 at 1494.

²⁸ Uniden Corp., 623 F. SUPP. 1485 at 1495.

²⁹ Uniden Corp., 623 F. SUPP. 1485 at 1496.

³⁰ Vault Corp. v. Quaid Software Ltd., 847 F.2d 255 (5th. Cir. 1988).

³¹ Vault Corp., 847 F.2d 255 at 267-68.

III. SUPER MARIO 64 DECOMPILED

The *Super Mario 64* decompilation project's goal was primarily to understand the game better and to aid with finding tricks that may benefit speedrunning.³² Aside from that, the project has two apparent applications. Firstly, it allows for users to modify the game—be it graphically, adding new levels, changing how the game plays, making the game more challenging, adding accessibility features, or allowing it to run on different devices. Secondly, the code may be learned from and used to develop completely different applications. For instance, allowing a game developer to better emulate the feel of *Super Mario 64*'s controls, which are still regarded quite highly.

The reverse engineered version of *Super Mario 64* (with additional assets) is capable of producing an exact copy of the original game's ROM when compiled with the same compiler. However, to get these additional assets, one must already have a copy of the game's ROM. In other words, to make copies of *Super Mario 64* using the reconstructed source code, one must first have a copy of *Super Mario 64*. As such, using the source code to make unmodified copies is functionally equivalent to just copying a file you already have.

This puts the project in an interesting legal position. While the project does not distribute anything from the compiled *Super Mario 64* ROM, it does distribute a C translation of the ROM's assembly code—something highly similar, if not identical, to the original source code. However, the code alone is largely useless and, at most, is a way to better understand the background processes of *Super Mario 64*.

While the translation of the code is, unambiguously, a derivative work, I would posit that the project is well within the conventional understanding of the fair use doctrine.

A. THE PURPOSE AND CHARACTER OF THE USE

The reverse engineered code was publicly released and free of charge. The purpose of the project was twofold: it allowed for a better understanding of *Super Mario 64*'s mechanics, and it allowed owners of *Super Mario 64* to modify a copy of their game. The fact that the release was non-commercial and ostensibly for educational purposes does help with the fair use argument, though “the mere fact that a use is educational and not for profit does not insulate it from a finding of infringement.”³³

Under the first factor, one must also ask whether the decompiled code

merely supersedes the objects of the original creation, or instead adds something new, with a further purpose or different character, altering the first with new expression, meaning, or message; it asks, in other words, whether and to what extent the new work is “transformative.”³⁴

³² Orland, *supra* note 3 (Speedrunning is the act of playing through a game in as short a time as possible).

³³ *Campbell v. Acuff-Rose Music Inc.*, 510 U.S. 569, 584 (1994).

³⁴ *Sony Computer Entertainment, Inc. v. Connectix Corp.*, 203 F.3d 596, 606 (9th Cir. 2000); *See Acuff-Rose*, 510 U.S. 569 at 579 (internal quotation marks and citations omitted).

Given that the project is a rewriting of code, it is unlikely that it would be considered transformative. I would, however, argue that it allows for transformative works of *Super Mario 64* to be made.

Lewis Galoob Toys, Inc. v. Nintendo of America, Inc.,³⁵ which concerned a device (the Game Genie) that allowed a user to alter three features of a Nintendo game, may be an appropriate case to consider. The court found that “Game Genie users are engaged in a non-profit activity. Their use of the Game Genie to create derivative works therefore is presumptively fair.”³⁶ Perhaps more importantly, the court held

a party who distributes a copyrighted work cannot dictate how that work is to be enjoyed. Consumers may use a Betamax to view copyrighted works at a more convenient time. They similarly may use a Game Genie to enhance a Nintendo Game cartridge’s audiovisual display in such a way as to make the experience more enjoyable.³⁷

While the Game Genie is a separate device that uses no code from the games it alters and leaves no lasting derivative works or copies, functionally, the released source code has a similar effect insofar as allowing for user modifications.

Courts are also free to consider the public benefit of a use.³⁸ It should go without saying that modifying a piece of software allows users to get more enjoyment or usage out of it—in some cases allowing for creative expression. Modification of software has been common practice for decades and is even encouraged by some software developers.

B. THE NATURE OF THE COPYRIGHTED WORK

The second factor asks what kind of work is being used. Factual works, like news reports, are less protected than creative works.³⁹ Video games muddy this distinction. As a whole, they are creative expressions, yet the code is utilitarian. Generally, though, code is considered to be sufficiently creative to qualify for copyright protection.⁴⁰

One may also wish to note that the second factor “typically has not been terribly significant in the overall fair use balancing.”⁴¹

³⁵ *Lewis Galoob Toys, Inc. v. Nintendo of America, Inc.*, 964 F.2d 965, 967 (9th Cir. 1992).

³⁶ *Galoob*, 964 F.2d 965 at 970.

³⁷ *Galoob*, 964 F.2d 965 at 971.

³⁸ *Sega Enterprises Ltd. v. Accolade, Inc.*, 997 F.2d 1510, 1523 (9th Cir.1992).

³⁹ *Campbell v. Acuff-Rose Music Inc.*, 510 U.S. 569, 586 (1994).

⁴⁰ *Oracle America, Inc. v. Google Inc.* 750 F.3d 1339, 1356 (Fed. Cir. 2014).

⁴¹ *Dr. Seuss Enterprises, LP v. Penguin Books*, 109 F.3d 1394, 1402 (9th Cir. 1997).

C. THE AMOUNT AND SUBSTANTIALITY OF THE PORTION USED

In terms of substantiality, the project distributes rewritten assembly files extracted from *Super Mario 64*'s ROM. It does not include models, textures, or sound files. The code, on its own, is somewhat useless, as it requires the assets that were not included in the release to make a functional ROM image. However, it would be foolish to write off the code as unimportant, as it is crucial to make the game function. Due to the project's nature, there is a large amount of code used—indeed, the *entire* project is rewritten assembly code.

This may not necessarily be an issue, however, as making entire copies of a work can be a fair use in some circumstances. *Sony Corp. of America v. Universal City Studios, Inc.*⁴² would suggest that recording a television broadcast for private home use is a fair use, as such copying “merely enables a viewer to see such a work which he had been invited to witness in its entirety free of charge.”⁴³ This reasoning was applied in *Lewis Galoob Toys, Inc. v. Nintendo of America, Inc.* where it was found to apply both to works that have been purchased (as opposed to invited to view for free), and to the creation of derivative works (as opposed to exact copies).⁴⁴

Such reasoning only applies to private home use, and while that is likely how the project will be used by most people, distributing derivative code is certainly not that. Although, *Sony* holds that the “copying of [an] entire work does not preclude fair use.”⁴⁵

The third factor depends on if the amount used is “reasonable in relation to the purpose of the copying,”⁴⁶ however “the extent of permissible copying varies with the purpose and character of the use.”⁴⁷ Generally, “[i]f the secondary user only copies as much as is necessary for his or her intended use, then this factor will not weigh against him or her.”⁴⁸ Given that the project's aim is to rewrite the assembly code of the original game into C, the fact that the only assets distributed are rewritten C files suggests that it is reasonable, at least for that purpose.

D. EFFECT OF THE USE UPON THE POTENTIAL MARKET

The fourth statutory factor is considered to be the most important,⁴⁹ as “a use that has no demonstrable effect upon the potential market for, or the value of, the copyrighted work need not be prohibited.”⁵⁰ One should consider market harm and what the effects of widespread conduct of similar actions would be.⁵¹ There are two apparent issues to consider:

⁴² *Sony Corp. of America v. Universal City Studios, Inc.* 464 US 417 (1984).

⁴³ *Universal City Studios*, 464 US 417 at 449–50.

⁴⁴ *Lewis Galoob Toys, Inc. v. Nintendo of America, Inc.*, 964 F.2d 965, 965-72 (9th Cir. 1992).

⁴⁵ *Sony Computer Entertainment, Inc. v. Connectix Corp.*, 203 F.3d 596, 606 (9th Cir. 2000); *See also Universal City Studios*, 464 US 417 at 449–50.

⁴⁶ *Campbell v. Acuff-Rose Music Inc.*, 510 U.S. 569, 586 (1994).

⁴⁷ *Acuff-Rose*, 510 U.S. 569 at 586–87.

⁴⁸ *Kelly v. Arriba Soft Corp.*, 336 F.3d 811, 820–21 (9th Cir. 2003).

⁴⁹ *Harper & Row, Publishers, Inc. v. Nation Enterprises*, 471 U.S. 539, 566 (1985) (citing 3 *Nimmer on Copyright* §13.05[A], at 13-76).

⁵⁰ *Sony Corp. of America v. Universal City Studios, Inc.* 464 US 450 (1984).

⁵¹ *Universal City Studios*, 464 US 417 at 451.

I. DOES PUBLICLY AVAILABLE SOURCE CODE AFFECT THE MARKET?

While it's difficult to ascertain the effects of public source code, *Super Mario 64* is not unique in having it available (there are a few hundred games with available source code), nor is it the first game to have reconstructed source code.⁵²

Given the number of instances of available source code, and the fact that other developers have released their own source code (sometimes with the game), I find it unlikely that it has a particularly negative impact on sales.

Anecdotally, both *Pokémon Red* and *Pokémon Blue* had been decompiled by the time they were re-released on the Nintendo 3DS console,⁵³ in spite of this, the games broke first-week digital sales records.⁵⁴ Similarly, *Half-Life 2* had its source code leaked a year before release,⁵⁵ but went on to sell at least 12 million total copies.⁵⁶ Source code releases are largely useless to the vast majority of video game players. It is much, much easier to buy a game (or pirate a game in a more conventional way) than it is to download the source code and compile it yourself.

II. DOES THE RELEASE METHOD AFFECT THE MARKET?

This particular release of recreated source code requires a copy of *Super Mario 64* to be of any use. This would suggest that this has no effect on sales of the game and may even increase sales. Actually getting a copy is easier said than done, however. At the time of writing, this either requires additional hardware (to copy the ROM contents from a cartridge) or modifying the software of a Wii or Wii U console with a copy of the game installed.⁵⁷ The latter having its own EULA and DMCA issues. Both methods would appear to fall within the exception in section 117 of the Copyright Act.⁵⁸

The (much easier) alternative is to download a pirated copy from the internet. One could make the argument that this release amounts to contributory infringement, as piracy is the

⁵² *List of commercial video games with available source code*, https://en.wikipedia.org/w/index.php?title=List_of_commercial_video_games_with_available_source_code&oldid=982781397 (last visited Oct. 10, 2020).

⁵³ *Pokered*, GITHUB, <https://github.com/pret/pokered>.

⁵⁴ Eddie Makuch, *Classic Pokémon Red, Blue, Yellow 3DS Re-Releases Break eShop Records*, GAMESPOT (10 March 2016), <https://www.gamespot.com/articles/classic-pokemon-red-blue-yellow-3ds-re-releases-br/1100-6435554/>.

⁵⁵ Geoff Keighley, *The Final Hours of Half-Life 2*, GAMESPOT (28 April 2016), <https://www.gamespot.com/articles/the-final-hours-of-half-life-2/1100-6112889/>.

⁵⁶ Oliver Chiang, *The Master of Online Mayhem*, FORBES (9 February 2011), <https://www.forbes.com/forbes/2011/0228/technology-gabe-newell-videogames-valve-online-mayhem.html>.

⁵⁷ Wii U releases of Nintendo 64 games contain a ROM file that can be copied, provided the user can access it after modifying their console; *See Legal Information (Copyrights, Emulators, ROMs, etc.)*, NINTENDO, <https://www.nintendo.com/corp/legal.jsp#roms> (last visited Sept. 24, 2020). A similar process can be done for the Wii releases (see <https://github.com/Plombo/vcromclaim>), although the service to purchase digital games on the Wii was discontinued in 2019; *See* Brian Cain (Plombo), *Vcromclaim*, GITHUB (May 4, 2016), <https://github.com/Plombo/vcromclaim>; *See also* *Wii Shop Channel Discontinuation*, NINTENDO, https://en-americas-support.nintendo.com/app/answers/detail/a_id/27560/~wii-shop-channel-discontinuation (last visited Sep. 24, 2020).

⁵⁸ Copyright Act, 17 U.S.C. § 117(a)(1) (2018).

easiest way to obtain the missing assets. Contributory infringement occurs when “one with knowledge of the infringing activity, induces, causes or materially contributes to the infringing conduct of another.”⁵⁹ Given that the project does not tell users where or how to obtain a pirated copy, let alone encourage the practice, I find it unlikely that this is applicable. At worst, it is comparable to the “constructive knowledge of the fact that [Betamax] customers may use that equipment to make unauthorized copies of copyrighted material,”⁶⁰ which is not enough for contributory infringement.

E. OTHER CONSIDERATIONS

*Micro Star v. FormGen Inc.*⁶¹ may be a particularly relevant case to consider. The defendants sold a CD containing 300 user-made levels for *Duke Nukem 3D*.⁶² The level (MAP) files did not contain any assets (only level geography and references to the base-game’s assets) and had to be played through *Duke Nukem 3D*.

Ultimately, the court held that

[t]he work that Micro Star infringes is the [*Duke Nukem*] story itself . . . A copyright owner holds the right to create sequels, and the stories told in the [] MAP files are surely sequels, telling new (though somewhat repetitive) tales of Duke’s fabulous adventures. A book about Duke Nukem would infringe for the same reason, even if it contained no pictures.⁶³

A footnote clarifies that this would not be the case if the MAP files allowed for a different (distinct-from-Duke) story to be told.⁶⁴

Perhaps more importantly, the level pack was found not to constitute fair use on the basis that the commercial sale of these levels impinged on FormGen’s rights to make new *Duke Nukem* works.⁶⁵ I would contend the fair use component of this case is inapplicable to the non-commercial rewriting of source code, due largely to the commercial nature of the release. The case does not comment on fair use, and simply applies the (different) facts to the four factors.

It has been suggested that, due to the simplicity of the modifications in the *Micro Star* case, the precedent should be reexamined for cases of more complex user modifications.⁶⁶

⁵⁹ Gershwin Publ’g Corp. v. Columbia Artists Man., Inc., 443 F.2d 1159, 1162 (2nd Cir. 1971).

⁶⁰ Sony Corp. of America v. Universal City Studios, Inc. 464 US 417, 439 (1984).

⁶¹ *Micro Star v. FormGen Inc.*, 154 F.3d 1107 (9th Cir. 1998).

⁶² *Micro Star*, 154 F.3d 1107 at 1109.

⁶³ *Micro Star*, 154 F.3d 1107 at 1112 (internal citations omitted).

⁶⁴ *Micro Star*, 154 F.3d 1107 at 1112 n.5.

⁶⁵ *Micro Star*, 154 F.3d 1107 at 1113.

⁶⁶ John Baldrice, *Mod as Heck: Frameworks for Examining Ownership Rights in User-Contributed Content to Videogames, and a More Principled Evaluation of Expressive Appropriation in User-Modified Videogame Projects*, 8 MINN. JL SCI. & TECH. 681, 691–692, 708 (2007).

F. PRECOMPILED MODIFICATIONS

Working on the assumption that the reconstructed source code of *Super Mario 64* is a fair use, does distributing a modified, compiled version of the source code also constitute a fair use?

Probably not. While the first and fourth factors of fair use may weigh in favor of released source code due to the non-commercial nature and minimal market effects (in part due to requiring a copy in the first place), a precompiled modified version (with all the missing assets included) is tantamount to piracy.

A key consideration is whether or not the use of a copyrighted work supplants the original.⁶⁷ Source code alone does not supplant the original game by virtue of being incomplete and largely useless to most players. The original game with some bells and whistles added absolutely does—especially when you can get it for free.

It is important to distinguish that if the modifications are distributed as separate files that a user has to add to the decompiled version (and subsequently compile themselves) that would, more than likely, be a fair use. The modifications may be entirely original code, causing no copyright issues themselves. Compilation of modified sourced code, thereby creating a derivative work, should be considered a fair use, provided it is for a private, home use.⁶⁸

IV. CONCLUSION

Despite the relatively few cases concerning reverse engineering software, there is a clear understanding by courts that, in and of itself, reverse engineering is a fair use when the purpose is to better understand software.

While a project of this nature has (to my knowledge) never been before a court, I find it likely that reconstructing and publicly releasing source code would be considered a fair use, primarily due to its non-commercial nature, and minimal market impact. This is perhaps why Nintendo has not taken action against the project's GitHub repository, and only sought to remove compiled versions.

⁶⁷ See *Campbell v. Acuff-Rose Music Inc.*, 510 U.S. 569, 579, 587-88, 591 (1994); See also *Harper & Row, Publishers, Inc. v. Nation Enterprises*, 471 U.S. 539, 566 (1985).

⁶⁸ *Lewis Galoob Toys, Inc. v. Nintendo of America, Inc.*, 964 F.2d 965, 972 (9th Cir. 1992).