VIDEO GAMES AND REVERSE ENGINEERING:

BEFORE AND AFTER THE DIGITAL MILLENNIUM COPYRIGHT ACT

JOE LINHOFF*

TABLE OF CONTENTS

* I would like to thank my classmate Judith Richards and professor Philip Weiser for their help and encouragement with this note.

INTRODUCTION

One way to learn is by taking things apart. Reverse engineering is the process of using tools to analyze a product, in a way its designer did not intend, to learn how it works.[1] Reverse engineering is a cornerstone of innovation and an intellectual property "safety valve."[2] Reverse engineering plays an essential role in keeping the video game industry healthy and competitive.

Even if you're not a video game fan, reverse engineering in the video game industry is important. The industry brings a large number of disciplines together into a single consumable package—the video game.[3] The industry has produced enormous revenue growth in a relatively short period of time.[4] This growth has occurred largely under the radar of big business and regulation as video games historically have been considered a novelty or entertainment aimed at a narrow subculture. However, advances in the industry have turned this novelty into mainstream entertainment with revenues comparable to movies.[5] Now the industry appears on a lot of radars including those of Sony,[6] Microsoft,[7] Qwest,[8]

---

1. *See* Pamela Samuelson & Suzanne Scotchmer, *The Law and Economics of Reverse Engineering*, 111 YALE L.J. 1575, 1578 (2002) (defining reverse engineering as "the process of extracting know-how or knowledge from a human-made artifact"); Kewanee Oil Co. v. Bicron Corp., 416 U.S. 470, 476 (1974) (defining reverse engineering as "starting with the known product and working backward to divine the process which aided in its development or manufacture"). In *Sony Computer Entertainment v. Connectix Corporation,* the court defined reverse engineering as:

> Reverse engineering encompasses several methods of gaining access to the functional elements of a software program. They include: (1) reading about the program; (2) observing "the program in operation by using it on a computer;" (3) performing a "static examination of the individual computer instructions contained within the program;" and (4) performing a "dynamic examination of the individual computer instructions as the program is being run on a computer."

Sony Computer Entm't, Inc. v. Connectix Corp., 203 F.3d 596, 599 (9th Cir. 2000).

2. Universal City Studios v. Reimerdes, 111 F. Supp. 2d 294, 322 (S.D.N.Y. 2000) (recognizing the doctrine of fair use as a "safety valve").

3. Video games bring together software, hardware, music, sound effects, choreography, story telling, user interface design, physical simulation, database programming, communications, and more.

4. *See* John Markoff, *Recession? Don't Tell The Video Game Industry*, N.Y. TIMES, May 24, 2002, at C4 ("Sales of game software alone reached $6.4 billion last year, putting the game industry in striking distance of Hollywood, which had box-office sales of $8.35 billion in 2001. And video game executives predict this year will be even stronger."); Chuck Salter, *Playing To Win*, FAST CO., Dec. 2002, at 80 ("Last year, U.S. computer- and video-game revenue surpassed domestic box-office receipts, and this year, the game industry is expected to widen that gap with more than $10 billion in sales.").

5. Markoff, *supra* note 4.

6. "Sony Computer Entertainment America Inc. (SCEA) markets the PlayStation family of products and develops, publishes, markets, and distributes software for the PS one

the Federal Communications Commission,[9] and Congress.[10] Reverse engineering is a technique widely used in the industry to understand and improve on others' work. It is also used to gain access to third party game machines, giving motivated game designers access to standard platforms. Reverse engineering plays an essential role in the industry's growth and is now threatened.

This note shows how reverse engineering is used in the video game industry and how the Digital Millennium Copyright Act (DMCA)[11] can make reverse engineering a crime. The note argues that Congress should amend the DMCA to expand allowances for reverse engineering practices. Section I provides the intellectual property (IP) context for reverse engineering. Section II explains how reverse engineering is used in the video game industry. Section III explores some of the pre-DMCA case law and the reverse engineering balance arising from those cases. Section IV looks at the DMCA and the impact the DMCA will likely have on reverse engineering in the video game industry. Section V concludes this examination by calling for Congress to amend the DMCA to allow reverse engineering practices.

---

console and the PlayStation2 computer entertainment system for the North American market." *See* PLAYSTATION, ABOUT SCEA, *at* http://www.us.playstation.com/about.aspx (last visited Jan. 6, 2004).

7. Microsoft notes on its web site that one if its "seven core business units" is "Home and Entertainment, including Microsoft Xbox, consumer hardware and software, online games, and our TV platform." *See* MICROSOFT, OUR COMMITMENT TO OUR CUSTOMERS, THE BUSINESS OF MICROSOFT (Jan. 25, 2004), *at* http://www.microsoft.com/mscorp/articles/business.asp.

8. Qwest is looking to video games to add to the demand for broadband Internet connections. *See* Qwest CEO Dick Notebaert, Remarks at the Silicon Flatirons Telecommunications Program at the University of Colorado School of Law: Cleaning Up the Telecom Mess (Feb. 26, 2003) [hereinafter *Notebaert Remarks*] (transcript available through the Silicon Flatirons Telecommunications Program at http://www.silicon-flatirons.org).

9. "Broadband technology will potentially allow users to download more information, including new multimedia applications, streaming news, music, games . . . ." FCC, BROADBAND: FREQUENTLY ASKED QUESTIONS, *available at* http://www.fcc.gov/cgb/broadband.html (last visited Jan. 6, 2004).

10. *Violence in the Media: Antitrust Implications of Self Regulation and Constitutionality of Government Action*, *hearing before the Senate Comm. on the Judiciary*, 106th Cong. (Sep. 20, 21 2000), *available at* http://frwebgate.access.gpo.gov/cgi-bin/useftp.cgi?IPaddress=162.140.64.88&filename=74413.wais&directory=/disk2/wais/data/106_senate_hearings.

11. Digital Millennium Copyright Act, Pub. L. No. 105-304, 112 Stat 2860 (1998) [hereinafter DMCA].

I. THE INTELLECTUAL PROPERTY CONTEXT OF REVERSE ENGINEERING

The practice of reverse engineering is subject to different treatment by different IP regimes. For example, patent law does not provide a reverse engineering defense to infringement, whereas trade secret laws do allow reverse engineering. The status of reverse engineering in the copyright regime is questionable and the subject of this note. This part looks more closely at the IP context of reverse engineering.[12]

A. Intellectual Property Law and Reverse Engineering

Patent law does not directly address reverse engineering. However, patents can deter and endanger reverse engineering by making the results of the effort unusable.[13] Patent law grants exclusive rights to an inventor to make, use, and sell an invention for up to 20 years.[14] The grant is "nearly absolute, barring even those who independently develop the invention from practicing its art."[15] Patent law does not provide a reverse engineering defense—patent holders have the right to sue those who reverse engineer their invention.[16] Video games have many elements that could qualify for patent protection including elements of hardware, software, algorithms, and data structures.[17]

Another danger to the practice of reverse engineering is the possibility that patent holders may extend protection beyond their actual invention. For example, the patent holder's exclusive right to make or use an invention could prevent others from "using" that invention in any of the steps that are necessary for reverse engineering.[18] Thus, if a patented element is difficult to decouple from unpatented elements, the

12. Also discussed in this part, contract law, through shrinkwrap or click-through licenses, can be a great deterrent to reverse engineering.

13. Julie E. Cohen & Mark A. Lemley*, Patent Scope and Innovation in the Software Industry,* 89 CAL. L. REV. 1, 21 (2001).

14. 35 U.S.C. § 154 (2000).

15. ROBERT P. MERGES ET AL, INTELLECTUAL PROPERTY IN THE NEW TECHNOLOGICAL AGE 14 (3d ed. 2003).

16. *Id.* at 197.

17. For a short list of issued software-related patent types, see ROBERT P. MERGES ET AL, INTELLECTUAL PROPERTY IN THE NEW TECHNOLOGICAL AGE 1032 (2d ed. 2000). Since *Chakrabarty*, the patent office has taken seriously the statement that almost anything under the sun that is made by man is patentable, except for laws of nature, physical phenomenon, and abstract ideas. Diamond v. Chakrabarty, 447 U.S. 303, 310 (1980) (citations omitted).

18. Patent law does not provide an intermediate copying allowance such as is found in copyright case law as discussed later in the note. Thus, a patent holder could characterize copying a program as running afoul of the holder's exclusive right to "make" and "use" their invention. Cohen & Lemley, *supra* note 13, at 26.

patent could be used to block access to, and thus block reverse engineering of the unpatented elements. In this way, patents could be used to stop reverse engineering and pose a threat to the video game industry.

Trade secret laws are primarily state law doctrines that protect "against the misappropriation of certain confidential information,"[19] provided reasonable steps have been taken to keep the information secret.[20] Trade secret laws do not prohibit "reverse engineering a legally obtained product to determine the secrets contained inside."[21] Nor do they affect those who independently discover or invent a product. Thus, trade secret law alone does not present an obstacle to video game reverse engineering.

Before the DMCA, copyright's fair use doctrine allowed for reverse engineering. Copyright protects "original works of authorship fixed in any tangible medium of expression"[22] but does not protect ideas, procedures, processes, or methods of operation.[23] Thus, although copyright protects the fixed source code and the fixed object code, it does not protect the functional aspects of a computer program. Copyright infringement actions can be brought against someone who makes literal copies of a program,[24] or who has access to a copyrighted program and makes a program that is substantially similar.[25] But copyright law does not prevent independent creation, nor does it protect functional elements. Basic copyright law supports the growth of the video game industry because it prohibits literal copying but does not prevent sharing of the underlying ideas.

The doctrine of fair use provides an important limit on copyright's strength. Regardless of how a copy is made, it is not infringement if the copy is a fair use "for purposes such as criticism, comment . . . or

19. MERGES ET AL, *supra* note 15, at 22.

20. *Id.*

21. *Id.* at 23. However, trade secret law combined with restrictive licensing terms prohibiting reverse engineering can prevent reverse engineering. *See* Bowers v. Baystate Tech., Inc., 320 F.3d 1317, 1323 (Fed. Cir. 2003).

22. 17 U.S.C. § 102(a) (2000).

23. "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." *Id.* at § 102(b).

24. Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 837 (Fed. Cir. 1992) [hereinafter *Atari I*]. *Atari I* was a Federal Circuit decision affirming the Northern District of California's decision to preliminarily enjoin Atari from exploiting Nintendo's copyrighted computer programs. *See id.* at 835. After the Federal Circuit's ruling in *Atari I*, the case returned to the Norther District of California where Nintendo filed a Motion for Summary Judgement against Atari. *See* Atari Games Corp. v. Nintendo of Am., Inc., 30 U.S.P.Q.2d 1401 (N.D. Cal. 1993) [hereinafter *Atari II*].

25. *Atari I*, 975 F.2d at 837.

research . . . ."[26] Fair use permits "public understanding and dissemination of the ideas, processes, and methods of operation in a work."[27] Fair use also "permits an individual in rightful possession of a copy of a work to undertake necessary efforts to understand the work's ideas, processes, and methods of operation."[28] In this way, fair use provides legal permission for reverse engineering copyrighted works.[29]

To counteract the fair use exception, video game vendors have begun to rely on shrinkwrap licenses in attempts to prohibit reverse engineering. For example, the popular and controversial game *Grand Theft Auto* includes a "Limited Software Warranty And License Agreement."[30] The license states that the "act of installing and/or otherwise using the software" constitutes agreement to be bound to the terms of the license.[31] The terms of the license expressly prohibit reverse engineering and any copying of the software not specifically allowed in the license.[32] However, the enforceability of these licenses, including the question of when state contract law can preempt federal copyright law, is unsettled.[33] If shrinkwrap licenses become enforceable and binding on use of software, there could be "no logical stopping point" as to what "limitations on copyright protection might be eliminated."[34] Because of the uncertainty and great potential of shrinkwrap licenses to change the legal landscape, the topic is beyond the scope of this note.

---

26. 17 U.S.C. § 107 establishes the four factors courts must use in examining fair use. *See infra* text accompanying note 146.

27. *Atari I,* 975 F.2d at 843.

28. *Id.* at 842.

29. *See* Philip J. Weiser, *The Internet, Innovation, and Intellectual Property Policy*, 103 COLUM. L. REV. 534, 551 (2003).

30. ROCKSTAR GAMES, GRAND THEFT AUTO, VICE CITY, TOURIST GUIDE 24 ¶ 1 (2003).

31. *Id.* at ¶ 1.

32. *Id.* at ¶ 5. You are allowed to install the software on your computer and "keep the original disk(s) and/or CD-ROM [] only for backup or archival purposes." *Id.* at ¶ 4. The license claims that "The Software and Accompanying Materials are protected by the United States copyright law and applicable copyright laws and treaties throughout the world." *Id.* at ¶ 3.

33. *See* Bowers v. Baystate Tech., Inc., 320 F.3d 1317, 1323 (Fed. Cir. 2003) (holding the shrinkwrap license prohibiting reverse engineering enforceable. "Under First Circuit law, the Copyright Act does not preempt or narrow the scope of Mr. Bowers' contract claim."); ProCD, Inc., v. Zeidenberg, 86 F.3d 1447, 1449 (7th Cir. 1996) (holding shrinkwrap licenses "enforceable unless their terms are objectionable on grounds applicable to contracts in general"); *but see* Vault Corp. v. Quaid Software Ltd., 847 F.2d 255, 269 (5th Cir. 1988) (holding license term unenforceable because the provision in Louisiana's law that allowed licenses prohibiting adaptation using "decompilation or disassembly" conflicts with and "'touches upon an area' of federal copyright law"). *Vault* recognized that a license restriction "against decompilation or disassembly is unenforceable." *Id.*

34. *Bowers*, 320 F.3d at 1338 (Dyk, J., concurring in part, dissenting in part).

B. No Coherent Treatment of Reverse Engineering

Patent, copyright, and trade secret law each treat reverse engineering differently.[35] The same game software, or parts of it, may be simultaneously protected under one or more of these regimes.[36] The danger is that if reverse engineering is not protected under each regime, it will lose protection altogether.[37] Commentators "advocate a coherent treatment of reverse engineering across intellectual property law."[38]

In a recent article, Julie Cohen and Mark Lemley suggested the creation of a coherent reverse engineering policy.[39] Cohen and Lemley "advocate a limited right to reverse engineer patented computer programs to permit study of those programs and duplication of their unprotected elements."[40] Under their treatment, reverse engineering of software would be treated consistently under patent, trade secret, and copyright law.[41]

A more developed, coherent treatment of reverse engineering of platforms is presented by Philip Weiser. Weiser presents a "competitive platforms model"[42] that would allow or prohibit reverse engineering depending on market conditions and the purpose of the reverse engineering.[43] A game maker would be allowed to reverse engineer a platform for vertical access, i.e. "between a platform and a complimentary product."[44] But prohibited from reverse engineering a competitor's platform for horizontal access, i.e. "between rival platforms."[45] The model views reverse engineering as a corrective action that should be allowed only after two preconditions are met.[46] First, "to the extent it seems clear that a company lacks market power," that company should be permitted to use its IP rights to prevent reverse engineering for

35. *See, e.g.,* Weiser, *supra* note 29, at 551.
36. *Id.* at 553.
37. *See* Cohen & Lemley, *supra* note 13, at 27.
38. Weiser, *supra* note 29, at 553; *see* Cohen & Lemley, *supra* note 13, at 6.
39. Cohen & Lemley, *supra* note 13, at 6.
40. *Id.*
41. *Id.* at 29.
42. Weiser, *supra* note 29, at 537. Regulation of rival systems is the core concern of this model. *Id.* at 556. This model would not allow firms to clone inventions as that would undermine important investment incentives. *Id.*
43. As market conditions change, so would the legal protection. "[W]hen a platform standard reaches or is headed for a dominant position in a market, intellectual property protection against reverse engineering should recede." *Id.* at 591.
44. Weiser, *supra* note 29, at 591. This occurred in Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992). *See also infra* Section III.
45. Weiser, *supra* note 29, at 560. "[T]he Ninth Circuit should have accepted Sony's claim of infringement." *Id.* at 602. This occurred in Sony Computer Entm't, Inc. v. Connectix Corp., 203 F.3d 596, 599 (9th Cir. 2000). *See also infra* Section III.
46. Weiser, *supra* note 29, at 594.

horizontal access.[47] Second, IP rights should be enforceable to block reverse engineering when the platform market is developing—the goal of which is to provide investment incentives[48] and produce Schumpeterian competition.[49] Then, after these preconditions are met, where it seems clear a single standard will emerge as dominant, reverse engineering should be allowed to trump IP rights.[50]

Both approaches recognize the pro-competitive gains reverse engineering provides and argue for allowance of the practice across the IP regimes.[51] Either approach presents an improvement over the current state of affairs.[52] The next section looks at reverse engineering in the video game industry.

II. REVERSE ENGINEERING IN THE VIDEO GAME INDUSTRY

The video game industry is an incredible success. It started in the early 1970s[53] and in 30 years has grown to run neck and neck with Hollywood's box office revenues.[54] Sixty percent of Americans play video games.[55] It is on the cutting edge of the high-tech industry, yet it did not crash with the rest.[56] The video game industry is being looked at to fill telecom's broadband pipes.[57] It has been subject to very little external regulation and has turned only occasionally to the legal system for help. No manufacturer or game developer has been able to monopolize the

47. *Id.*

48. *Id.* at 584.

49. *Id.* at 593. A strong Schumpeterian view promotes the use of 'roadblocks' around which innovators must find a way if they are to get to the market. The belief is that "market power is temporary," and "monopolies are both acceptable and necessary to facilitate technological innovation." *Id.* at 576-77. Weiser rejects "pure Schumpeterian thinking" as a driver for IP policy. *Id.* at 581.

50. *Id.* at 593.

51. *Id.* at 600; Cohen & Lemley, *supra* note 13, at 22.

52. Cohen and Lemley's approach may be difficult in practice as a reverse engineer disassembling the software for a game will find it hard to distinguish between protected and unprotected elements. Weiser's approach will be particularly hard to apply in the video game industry due to the industry's 5 year cyclical nature – reverse engineering would be allowed when a manufacturer has market power in years 3 and 4, but not in years 1, 2, or 5 when the system is not as popular. It is also not clear if either approach sufficiently clears the hurdles out of the way of an unsophisticated entrepreneur.

53. STEVEN L. KENT, THE ULTIMATE HISTORY OF VIDEO GAMES 25 (2001). The Magnavox Odyssey, the first console game system, was released in 1972, and several competitors joined the market the next year.

54. Tom Standage, *Games Get Serious*, THE ECONOMIST: THE WORLD IN 2003, Dec. 2002, at 104; Markoff, *supra* note 5.

55. INTERACTIVE DIGITAL SOFTWARE ASS'N, QUICK FACTS ABOUT VIDEO GAME CONSOLES AND SOFTWARE, *at* http://www.idsa.com/consolefacts.html (last visited Mar. 5, 2004).

56. Standage, *supra* note 54, at 104; Steve Alexander, *Video-Game Industry Hopes to Take Success Online*, MINNEAPOLIS STAR TRIB., Sept. 30, 2002, at 1D.

57. *Notebaert Remarks*, *supra* note 8; Standage, *supra* note 54, at 104.

market.[58] Small teams come out of nowhere and create best-selling games.[59] As discussed below, reverse engineering plays a major role in the industry's success.

This section contains interviews with three industry veterans who discuss three kinds of reverse engineering. Part A discusses how Mike Schwartz successfully reverse engineered access to the Sega Genesis game platform for Electronic Arts at about the same time as the *Sega v. Accolade* case.[60] Second, in Part B, Mark Loffredo talks about the reverse engineering of video game hardware. And finally, in Part C, Will Carlin explains how a kind of reverse engineering can be used to analyze, understand, and build on other game designs.

A. Reverse Engineering Platform Access

Mike Schwartz worked at Electronic Arts (EA) and reverse engineered the Sega Genesis.[61] EA found Sega's licensing agreement onerous, and decided to reverse engineer the system.[62] EA setup a "clean room modeled after the PhoenixBIOS case."[63] A clean room is used to monitor and control information flow.[64] Schwartz had to be screened off from the rest of the company while he was exposed to Sega's copyrighted information. Schwartz worked in a room named "Chernobyl." This was the smoking room and the only room with a door.[65] Chernobyl was next to the kitchen, had a big ceiling fan to suck out the smoke, and a bunch

58. Magnavox, Atari, Nintendo, Sega, and Sony have at different times led the console market.

59. id Software, a four person company, created *Wolfenstein 3-D* in 1992, *DOOM* in 1993, and *Quake* in 1996. ID SOFTWARE, INC, ID SOFTWARE BACKGROUNDER, *at* http://www.idsoftware.com/business/history (last visited Feb. 6, 2004).

60. Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992).

61. Telephone Interviews with Mike Schwartz (Feb. 18, 2003; Mar. 6, 2003) (notes on file with author). This reverse engineering effort was accomplished at the same time and for access to the same platform that was at the center of *Sega v. Accolade*.

62. Schwartz, *supra* note 61. The licensing deal was that Sega would manufacture the game cartridges and sell them back to the licensee. On top of this, the licensee would pay to Sega a fixed amount per game sold.

63. In the late 1970s the BIOS on the IBM was successfully reverse engineered and licensed. BIOS stands for Basic Input Output System. It is a set of routines that serve primarily as a low level software interface to the hardware. This reverse engineering allowed others, including Compaq, to make IBM compatible motherboards. This was a critical step for "open architecture" and the success of the PC, it "set the course of computing in general." *Id.*

64. For example, Schwartz had his email monitored by attorneys. *Id.* A clean room process usually involves two figurative rooms. The room where the engineers are exposed to copyrighted material could be called the 'dirty' room. The engineers in this room work to produce a manual that does not contain any protected material. In the second figurative room, the 'clean' room, engineers use the manual freely to create games.

65. *Id.*

of EA's trophies in it.[66] For the task he used his own home-built reverse engineering software and hardware tools.[67]

In Chernobyl, Schwartz wrote a manual describing the functional elements required to program games for the Genesis. In terms of what the manual could contain, "addresses of registers and what they did was OK, but no snippets of code."[68] This manual could then be used by anyone in the company since the manual did not contain any of Sega's copyrighted information. Lawyers reviewed all the information he prepared before anyone on the outside could look at it.[69] At one point, he "corrupted someone by accident. This person became 'dirty'"[70] and was disqualified from subsequent development because of their exposure to protected information. Working in the clean room, it took Schwartz a month to reverse engineer the Genesis.[71]

This process turned out great for EA. They went to Sega in Japan, showed that they had reverse engineered the system, and were able to negotiate a very favorable licensing agreement.[72]

B. Reverse Engineering Hardware

Mark Loffredo has been designing hardware since 1982 and designed the arcade game hardware, including custom graphics chips, for many top selling games including *Mortal Kombat*, *Terminator 2*, *NBA Jam*, and *Crusin' USA*.[73] Reverse engineering arcade game hardware is not unheard of.[74] In the 1980s, there was a lot of paranoia in the industry that pirates were going to reverse engineer boards and make

66. *Id.*
67. *Id.*
68. Schwartz, *supra* note 61.
69. *Id.*
70. *Id.* The purpose of the clean room was to keep Sega's copyrighted information from the rest of the employees since copyright does not protect independent invention, if there were ever to arise a question, EA could show that none of their employees, except Schwartz, ever had access to Sega's copyrighted work.
71. It turned out the Genesis's graphics chip was very similar to one in the Colecovision system that he was familiar with. This saved him an enormous amount of time. He went home, got the manual for the other chip, and was able to test for differences. *Id.*
72. *Id.* As far as Schwartz can remember, "Sega demanded to make all the carts. EA could buy as many carts from Sega as it wished, but had to pay Sega's price. Sega's price included something like $15/cart in usury fee!" *Id.* Schwartz, on the other hand, was not allowed to develop original Genesis games for fear that he would unconsciously repeat five lines of code. *Id.*
73. Telephone Interview with Mark Loffredo (Mar. 6, 2003) (notes on file with author). *Mortal Kombat*, *Terminator 2*, *NBA Jam*, and *Cruisn' USA* are all arcade games developed and manufactured at Midway Manufacturing Co.
74. In the 1981 case of *Midway Manufacturing v. Dirkschneider* the defendants were "engaged in the manufacture, distribution, and sale of video games…virtually identical to" *Galaxian*, *Pac-Man*, and *Rally-X.* 543 F. Supp. 466, 472 (D. Neb. 1981).

copies.[75] However, the instances of this actually happening were very low.[76] Complexity of new chips, encryption, and copy-protection advances make reverse engineering cost prohibitive.[77] "You really need to make a minimum number of games to make it worth while. This minimum would have to be in the multiple thousands."[78]

Loffredo "rarely looks at other chips" as he does not want "their design bias."[79] He would rather figure out something on his own than "shoe-horn an inadequate design that's hard to understand" into his system.[80] He is not sure exactly what is and what is not reverse engineering, but "ripping off concepts in the industry is widespread."[81] Everyone "continually looks at the competition and figures out how to do it better."[82]

Loffredo's latest system design is built around Xilinx[83] programmable logic chips.[84] These chips blur the line between hardware and software. They are programmed with a bitstream of data that is sent to the chip.[85] The data, which is a type of object code, defines how the chip acts. For example, a bitstream of code could program the chip to act like the 6502 microprocessor found in the early Apple II computers, then a different bitstream could be sent to the same chip to re-program it to act like the 68000 microprocessor found in early Apple Macintosh computers, and then a third bitstream of code could again be sent to re-program the chip to emulate the game of *Pong*.[86] In effect, software programs the hardware to be hardware. "It takes a lot of time to create the IP to program these chips."[87] Hardware engineers now create

75. Loffredo, *supra* note 73.

76. *Id.* There are supposedly over 90,000 *Robotron* games worldwide and only 60,000 manufactured legally – however, there was no copy-protection built into these games and they were relatively easy to copy. *Id.*

77. *Id.*

78. *Id.*

79. *Id.*

80. *Id.*

81. Loffredo, *supra* note 73.

82. *Id.*

83. Xilinx, Inc., http://www.xilinx.com (last visited Jun. 22, 2004).

84. Loffredo, *supra* note 73. Programmable logic devices, in contrast to fixed logic devices, allow the device's function to be programmed or reprogrammed at any time. XILINIX, WHAT IS PROGRAMMABLE LOGIC?, *at* http://www.xilinx.com/company/about/programmable.html (last visited Jun. 21, 2004).

85. Loffredo, *supra* note 73.

86. *See* OPENCORES.ORG, *T65 CPU: Overview*, *at* http://www.opencores.org/projects.cgi/web/t65/overview (last visited Feb. 24, 2004). The Verilog for a Pong-like game is available at http://www.fpga4fun.com/PongGame.html (last visited Feb. 25, 2004).

87. Loffredo, *supra* note 73.

hardware using programming languages and tools much like software engineer's tools and languages.[88]

This illuminates a fundamental issue as to the reach and importance of copyright law—the distinction between hardware and software is collapsing, and as copyright reaches to protect software, it also protects hardware.

C. Reverse Engineering Game Designs

Game designs are often based on, and evolve from, other games. *Asteroids* followed after and improved on *Space War, Galaga* improved on *Space Invaders, Mortal Kombat* improved on *Street Fighter*, etc.[89] Reverse engineering can be used by game designers to analyze and understand how a game is put together and what makes it work. Reverse engineering can uncover the internal rules of a game, how the scoring works, the pacing of the game, how the camera works, the game physics, the number of frames and timing of animations, and more. An experienced designer can determine some of these things by playing the game. However, a form of reverse engineering is useful for discovering other elements. The game designer's main reverse engineering tool is the video recorder. A game's inner-workings can often be discovered by watching a video of the game frame by frame, slowing the action down enough to see every detail and every change.

Game designer Will Carlin says "the whole industry is built on reverse engineering."[90] He started designing games in 1984. His latest game, *Big Buck Hunter*, has been the number one arcade game for two years.[91] Successful games are made by borrowing ideas.[92] For example, Sega's new game *Getaway* has the same play mechanics as Rockstar Games's *Grand Theft Auto*.[93] A game's play mechanics are the subtle combination of algorithms, math, physics, and ad-hoc programming that

88. Hardware engineers can now "design a very complex logic circuit in front of a text editor." *Id.* They can design hardware using Verilog, a type of High-level Design Language (HDL) code, which "takes much of its syntax from the C language." *Id.* The HDL compilers and tools act very much like those for "software programming." *Id.*

89. In *Space War, 1961*, the first real computer game, two ships flew around space much like those in Atari's 1979 game Asteroids. RUSEL DEMARIA & JOHNNY L. WILSON, HIGH SCORE 12, 49 (2002). Namco's 1981 game of *Galaga* improved on the basic design of Taito's 1978 game *Space Invaders*. *Id.* at 46, 76. Midway's 1992 game *Mortal Kombat* built on the design of Capcom's 1987 game *Street Fighter* which can trace its roots back to Data East's 1984 game *Karate Champ*. *Id.* at 280-81. *See Killer List of Video Games*, INTERNATIONAL ARCADE MUSEUM, *available at* http://www.arcade-museum.com (last visited Feb. 3, 2004) (provides a reference of arcade games and manufacturers).

90. Telephone Interview with Will Carlin (Feb. 27, 2003) (notes on file with author).

91. *Id.*

92. Sometimes "it's downright plagiarism." *Id.*

93. *Id.*

determine how the game responds to the controls. Carlin says a designer could video tape a car skidding around a corner and then analyze the skid marks in the video to get an idea of the game physics involved and how to recreate that effect.[94]

Video game software engineers, hardware engineers, and game designers all engage in different types of reverse engineering. Reverse engineering is important at each level. The industry has grown in leaps and bounds in large part due to competitor's' ability to understand, analyze, and build on other's work. If reverse engineering is outlawed, we should expect a big drop-off in the number of new games produced every year.[95]

III. PRE-DMCA CASE LAW FOR REVERSE ENGINEERING

The case law involving reverse engineering in the video game industry has focused primarily on copyright issues. This section examines three of the primary pre-DMCA cases that address reverse engineering of video game platforms.

A. Atari v. Nintendo

The controversy between Atari and Nintendo lays out most of the pre-DMCA framework for the legal analysis of reverse engineering in the video game industry. In the late 1980s, the 8-bit Nintendo Entertainment System (NES) had an 80% market share.[96] The security mechanism on the NES, called 10NES, prevented games from running on the system unless they contained a special chip and software.[97] Nintendo used the security mechanism to push game developers into licensing contracts.[98] Atari began reverse engineering the NES in 1986, the same year it was introduced in the US.[99]

Atari first tried to reverse engineer the security mechanism by "monitoring the communication" between the game cartridge and the game console.[100] However, this approach did not give them enough information. Next, in an attempt to re-create a listing of the object code, Atari "chemically peeled layers from the NES chips to allow microscopic examination of the object code."[101] However, this too failed as Atari's

94. *Id.*
95. Carlin, *supra* note 90.
96. Atari Games Corp. v. Nintendo of Am., Inc., 897 F.2d 1572, 1574 (Fed. Cir. 1990).
97. KENT, *supra* note 53, at 372.
98. *Atari I,* 975 F.2d at 836-37 (Fed. Cir. 1992).
99. KENT, *supra* note 53, at xiv. *Atari I,* 975 F.2d at 836.
100. *Atari I,* 975 F.2d at 836.
101. *Id.* Microscopic examination will not literally reveal the object code. *See* KENT, *supra* note 53, at 372.

engineers were not able to sufficiently reconstruct the code from the peeled layers of the chips.[102] Atari finally turned to their lawyers. As part of the copyright process, Nintendo had filed a listing of their object code with the Copyright Office. This listing contained the information Atari had unsuccessfully sought through reverse engineering. Atari's lawyers made up a fictional lawsuit, claimed that Nintendo was suing them for copyright infringement, submitted false affidavits to the Copyright Office, and got a copy of the listing.[103]

Atari was soon thereafter successful.[104] Atari developed its own security chip and program, which they named Rabbit, to mimic the 10NES.[105] In 1988, they began producing their own games "without Nintendo's strict license conditions."[106]

Nintendo and Atari sued each other. One of the issues was Atari's right to reverse engineer Nintendo's security mechanism. The court stated that except for the taint from their purloined copy of the 10NES program, Atari's reverse engineering was a fair use in so far as it was necessary to understand the 10NES.[107] "When the nature of a work requires intermediate copying to understand the ideas and processes in a copyrighted work, that nature supports a fair use for intermediate copying. Thus, reverse engineering object code to discern the unprotectable ideas in a computer program is a fair use."[108] However, because their copy of the 10NES program was fraudulently obtained, Atari lost this defense.[109]

Another related issue was whether Atari could copy program code that was not currently needed, but that might be needed in the future if Nintendo upgraded their security.[110] The court refused to extend fair use to a preemptive right to copy.[111] Atari further argued that the signal stream itself was not copyrightable.[112] Here the district court agreed, and found that the signal stream did not overcome the originality

102. *Atari I*, 975 F.2d at 836.

103. *Id.*; KENT, *supra* note 53, at 373.

104. It is not clear how useful the stolen information was. *See* KENT, *supra* note 53, at 373 (discussing how Atari's clean room operation was close to breaking the 10NES at this time, and also Ed Logg's quote implying that no one used the information from the Copyright Office and merely that "some paralegal f---ed up!").

105. *Atari I*, 975 F.2d at 836 (Fed. Cir. 1992).

106. *Id.* at 836-37.

107. *Id.* at 843.

108. *Id.*

109. *Id.* ("To invoke the fair use exception, an individual must possess an authorized copy of a literary work.").

110. *Atari II*, 30 U.S.P.Q.2d at 1406-07.

111. *Id.* at 1407.

112. *Id.* at 1403.

requirement.[113] The ruling that the signal stream was not itself copyrightable would limit "Nintendo's rights in the 10NES program in [] two ways."[114] First, Atari, as a competitor, may copy those portions of the program that are necessary to access the unprotected signal stream for interoperability, and may include that code in their final version.[115] Second, they may make intermediate copies of the entire program in order to reverse engineer necessary sequences of the unprotected signal stream.[116]

However, the favorable ruling regarding the signal stream was not enough to overcome Atari's fraud.[117] Atari lost the dispute.[118] As *Atari v. Nintendo* was winding down, *Sega v. Accolade*, which addressed some of the same issues, was just beginning.

B. Sega v. Accolade

Accolade used a two-step clean room process to create video games compatible with the Sega Genesis game console.[119] The first step was to reverse engineer the system and create a development manual. Accolade purchased a Genesis video game console and three game cartridges.[120] Then they wired up the system so they could examine the data moving between the cartridge and the console during game play.[121] The engineers dumped the code from the cartridges, disassembled it, printed it, and studied it.[122] The engineers then loaded a mix of their own code and modified code from the purchased cartridges onto the console and tested it until they discovered how to unlock the Genesis.[123] "At the end of the reverse engineering process, Accolade created a development manual that incorporated the information it had discovered about the requirements for a Genesis-compatible game."[124] The manual did not contain any Sega code, but only contained "functional descriptions of the

113. *Id.* at 1405 (citing Feist Publ'ns v. Rural Tel. Serv. Co., 111 S. Ct. 1282, 1289 (1991)).
114. *Atari II*, 30 U.S.P.Q.2d. at 1408-09.
115. *Id.* at 1408-09.
116. *Id.*
117. "To the extent, however, Nintendo is likely to show misappropriation and copying of the unauthorized Copyright Office copy, it is likely to succeed on the merits of its infringement claim." *Atari I*, 975 F.2d at 836 (Fed. Cir. 1992).
118. The court granted Nintendo's summary judgment motion regarding the copyright infringement claim, finding elements in Atari's Rabbit program "firmly establish illicit copying." *Atari II*, 30 U.S.P.Q.2d at 1406-07.
119. Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d at 1514 (9th Cir. 1992).
120. *Id.* at 1514-15.
121. *Id.*
122. *Id.* at 1515.
123. *Id.*
124. *Id.*

interface requirements."[125]

The second step was to use the development manual to create its own games for the Genesis.[126] In 1990, Accolade released *Ishido*, a game that it had developed and released for the Macintosh and IBM PC.[127] In 1991, Sega began manufacturing a new version of the Genesis console with which *Ishido* would not work.[128] Accolade embarked on a second round of reverse engineering. The engineers found a small piece of code that was ignored by the original Genesis, but which was necessary to unlock the new Genesis.[129]

Sega filed a claim of copyright infringement against Accolade, not for the resulting product, but for Accolade's intermediate copying during their reverse engineering process.[130] The district court found for Sega primarily because Accolade's use was commercial—Sega had lost sales—and Accolade apparently had an alternative that did not require the intermediate copying of code.[131] The district court ordered Accolade to recall all of its infringing games within 10 business days.[132]

On appeal Accolade made four arguments relating to the copyright infringement claim: (1) intermediate copying is not infringement; (2) disassembly of object code to gain understanding of the ideas and functional concepts is lawful; (3) disassembly is authorized by section 117 which allows computer programs to be read into memory; and (4) disassembly in order to gain understanding of ideas and functional concepts is a fair use.[133] The court dismissed the first three arguments,[134] but accepted the fourth and dissolved the district court's order:[135]

> [D]isassembly of copyrighted object code is, as a matter of law, a fair use of the copyrighted work if such disassembly provides the only means of access to those elements of the code that are not protected by copyright and the copier has a legitimate reason for seeking such access.[136]

---

125. Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d at 1515 (9th Cir. 1992).
126. *Id.*
127. *Id.*
128. *Id.* The new console had an updated security system.
129. *Id.* at 1515-16.
130. *Id.* at 1516.
131. Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d at 1517 (9th Cir. 1992). Sega claimed that there was an alternative way to make interoperable cartridges and was willing to show Accolade's attorneys, but not Accolade's engineers, the cartridges that accomplished this.
132. *Id.*
133. *Id.* at 1517-18.
134. *Id.* at 1519 (noting that "intermediate copying of computer object code may infringe . . . regardless of whether the end product of the copying also infringes . . .").
135. Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d at 1518 (9th Cir. 1992).
136. *Id.*

This decision, along with *Atari v. Nintendo,* validated reverse engineering as a fair use defense. Fair use is explored further in the following case.

C. Sony v. Connectix

Sony v. Connectix took a closer look at fair use and addressed how reverse engineering could be used by a competitor[137] to create a compatible platform. Sony made the PlayStation video game console as well as games for the console.[138] PlayStation games were released on standard compact disks.[139] Connectix made a program, the Virtual Game Station (VGS), for the Apple Macintosh computer that allowed PlayStation games to be played on the Macintosh "even if you don't yet have a Sony PlayStation console."[140]

In the process of creating the VGS, Connectix analyzed the BIOS of the PlayStation.[141] The BIOS is a copyrighted program that acts as a low level interface between the software and the hardware.[142] Connectix engineers used the copyrighted BIOS for reference and testing only—none of the copyrighted BIOS appeared in the final VGS product.[143] The court found that Connectix's use of the copyrighted BIOS was a fair use.[144]

The "fair use doctrine preserves public access to the ideas and functional elements embedded in copyrighted computer software programs."[145] In determining whether a use qualifies under the doctrine, the Copyright Act lists the factors for consideration:

> (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.[146]

---

137. Sony Computer Entm't, Inc. v. Connectix Corp., 203 F.3d 596, 599 (9th Cir. 2000). Connectix and Sony were not ordinary competitors in the platform market. Connectix provided an alternate PlayStation-compatible platform that may have extended the sales of PlayStation games and thus increased the value of the PlayStation to Sony.

138. *Id.*

139. *Id.*

140. *Id.* at 599, 601. They were also working on a Microsoft Windows version.

141. Sony Computer Entm't, Inc. v. Connectix Corp., 203 F.3d at 601 (9th Cir. 2000).

142. *Id.* at 599-600.

143. *Id.* at 600.

144. *Id.* at 602.

145. *Id.* at 603.

146. Sony Computer Entm't, Inc. v. Connectix Corp., 203 F.3d at 602 (9th Cir. 2000) (quoting 17 U.S.C. § 107 (2000)).

The first factor the court looked to was the nature of the copyrighted work. Because Sony's BIOS contained unprotectable elements that could not be examined without copying, the BIOS was accorded a "lower degree of protection than more traditional literary works."[147] Since the copying was necessary to examine those unprotectable elements, factor (2), the nature of the copyrighted work, weighed heavily in favor of Connectix.[148] As to the other factors, because the final product did not itself contain infringing material, factor (3), amount and substantiality, contributed very little to the analysis.[149] Factor (1), purpose and character, weighed in favor of Connectix because their product was "modestly transformative," did not merely supplant the PlayStation, and was a "wholly new product" notwithstanding the similarity of uses and functions.[150] Finally, because the VGS was "a legitimate competitor in the market for platforms on which Sony and Sony-licensed games [could] be played," factor (4), any economic loss incurred by Sony, "[did] not compel a finding of no fair use."[151]

These three cases show how reverse engineering acts as a fair-use balance to copyright law. The *Connectix* decision was published February 10, 2000. The new anti-circumvention rules of the DMCA went into effect a few months later on October 28, 2000.[152] Before Section IV examines how the DMCA alters the balance, the following part looks at the pre-DMCA reverse engineering balance.

D. The Reverse Engineering Balance

The above pre-DMCA decisions established a balance between a copyright holder's right to exclude, and a third party's right to access work through reverse engineering. This part looks closer at the balance focusing on two important factors: first, that manufacturers control the cost of reverse engineering; and second, that reverse engineering promotes competition by opening up access to platforms.

147. *Id.* at 603 (quoting Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d at 1514 (9th Cir. 1992)).
148. *Id.*
149. *Id.* at 606.
150. *Id.* at 606-07.
151. *Id.* at 607.
152. 17 U.S.C. § 1201(a)(1)(A) (2000) ("The prohibition contained in the preceding sentence shall take effect at the end of the 2-year period beginning on the date of the enactment of this chapter [enacted Oct. 28, 1998].").

1. Manufacturers Control Reverse Engineering Cost

Manufacturers control how difficult their platform is to reverse engineer. In designing the 8-bit NES, Nintendo used a combination of a hardware chip in each game cartridge and software embedded in their platform as a key and lock. This combination of hardware and software, largely because it included a hardware element, presented a significant access barrier to Atari. Compare this to the key and lock Sega used in their system: Sega used a generic software key, included in every game cartridge, to unlock the Genesis platform.[153] Sega's lock and key did not require an additional hardware chip in each game cartridge, as did Nintendo's system, and most likely was less costly to both Sega to manufacture and a competitor to reverse engineer.

Manufacturers control the cost of reverse engineering, both in terms of their own design and manufacturing costs, and also in terms of how difficult and costly the system is to reverse engineer. As the complexity of the lock and key grows, so does the effort required to successfully reverse engineer a system. Because manufacturers have the ability to control access to their platform without help from the legal system, legal protection can be redundant.

2. Platform Access

Platform access is often at the focal point of the reverse engineering debate.[154] Reverse engineering gives the third party a choice: negotiate with the platform manufacturer, or attempt to reverse engineer access. In addition to promoting fair licensing by placing an upper limit on the terms of acceptable licenses, i.e. the cost of reverse engineering,[155] this choice opens up access and markets for small developers without the resources or sophistication needed to negotiate a deal. This type of vertical access plays an important role in the industry's growth. Without access to standard platforms, small entrepreneurs are not able to participate in a large segment of the video game industry. For these

153. Sega's system also triggers Sega's trademark. Their lock and key was named the Trademark Security System. Sega appears to have weighed the tradeoffs, and made a decision to implement a security system with low up-front costs that depended ultimately on trademark law.

154. Vertical platform access was at issue in *Atari II* and *Sega v. Accolade*. Horizontal platform access was at issue in *Sony v. Connectix*. In the video game industry, horizontal platform access merits less discussion because of two important factors. First, as Judge Fern Smith notes in *Atari II*, there is a significant time lag needed to successfully reverse engineer a system. *Atari II*, 30 U.S.P.Q.2d 1401. And second, emulation of a system will almost always require next generation technology. Both of these factors give the manufacturer of a successful platform time to recover investments.

155. *See* Weiser, *supra* note 29, at 548.

entrepreneurs, to the extent the law blocks reverse engineering, the law blocks access to the market. Losing this group of entrepreneurs will likely lead to an industry-wide loss of creativity, game content diversity, innovation, and competition. Because manufacturers themselves have the ability to dial in their own level of protection, redundant legal protections should be added carefully, lest they deter competition.

3. Healthy Balance

The reverse engineering balance before the DMCA was healthy. Pamela Samuelson and Suzanne Scotchmer studied the software industry generally, and concluded that reverse engineering had not hurt the industry.[156] Because "decompilation and disassembly are time-consuming and resource-intensive, these forms of reverse engineering [have] not . . . significantly undermine[d] incentives to invest in platforms."[157]

The courts also recognized that a balance was needed. Without the reverse engineering allowance, a copyright holder's rights are similar to those rights granted by the more stringent patent process. If a competitor "wishes to obtain a lawful monopoly on the functional concepts in its software, it must satisfy the more stringent standards of the patent laws."[158] Judge Fern M. Smith states the doctrine for allowing reverse engineering:

> By requiring independent game developers to carefully study a particular security system and discern which program instructions are truly necessary for present compatibility, console manufacturers will have a limited period of time in which to control the market for compatible games. In this time period, some third party game developers are likely to enter license agreements with Nintendo, particularly if they have limited resources. After a relatively short period of time, however, other developers will enter the game market with independently produced, but still compatible games. In addition, if third party developers who entered license agreements later find the license agreements too onerous, there still exists the option of reverse engineering the security system after the expiration of their license agreement. Thus, a fair use defense which allows copying for present compatibility balances the incentives for both game developers and console manufacturers.[159]

156. Samuelson & Scotchmer, *supra* note 1, at 1612-13.

157. *Id.* at 1622.

158. Sony Computer Entm't, Inc. v. Connectix Corp., 203 F.3d at 605 (9th Cir. 2000) (citing Bonito Boats, Inc. v. Thunder Craft Boats, Inc., 489 U.S. 141 (1989)).

159. *Atari II*, 30 U.S.P.Q.2d at 1407.

The three cases described above, *Atari v. Nintendo*, *Sega v. Accolade*, and *Sony v. Connectix*, address reverse engineering platform access and arrive at a healthy, competitive balance that allows some access while protecting incentives for manufacturers. The next section discusses how the DMCA throws off this balance.

IV. THE DMCA AND REVERSE ENGINEERING

The DMCA was signed into law October 28, 1998.[160] In addition to implementing the World Intellectual Property Organization Copyright Treaty,[161] the DMCA was enacted to support the "adaptation of the law of copyright to the digital age."[162] The DMCA significantly changes the law relating to reverse engineering and throws off the balance between a copyright holder's rights, fair use, and competition. The anti-circumvention provisions of the DMCA provide a way to seal off technology by severely restricting access through reverse engineering. This section looks at the text of the DMCA in light of its early judicial interpretations and discusses how it will likely affect the video game industry.

A. Anti-Circumvention and Reverse Engineering

In the digital age, reverse engineering and circumvention are two sides of the same coin. To the extent the DMCA prevents and limits circumvention, it prevents and limits reverse engineering. The anti-circumvention provisions of the DMCA prohibit the circumvention of a "technological measure that effectively controls access to a work protected under this title."[163] The effect is that any digital work protected under copyright has a new form of legal protection: anti-circumvention.[164] Section 1201 limits anti-circumvention in two important ways.[165] First, as stated in section 1201(a)(1)(A), anti-circumvention actions are limited to works that are "protected under this title."[166] This limitation, in conjunction with section 102, prevents anti-

160. Digital Millennium Copyright Act, Pub. L. No. 105-304, 112 Stat 2860 (1998).

161. *Id.*

162. Universal City Studios v. Reimerdes, 111 F. Supp. 2d 294, 316 (S.D.N.Y. 2000).

163. 17 U.S.C. § 1201(a)(1)(A) (2000).

164. As long as a "technological measure" can be added to "control access" – this is a trivially low bar.

165. 17 U.S.C. § 1201(c)(1) states that "[n]othing in this section shall affect rights, remedies, limitations, or defenses to copyright infringement, including fair use, under this title." If, as has been done in *Reimerdes*, anti-circumvention and copyright infringement are two independent causes of action, then this section says nothing about the interaction between fair use and anti-circumvention. *Reimerdes*, 111 F. Supp. 2d. 294. *Reimerdes* interpreted this silence as the elimination of fair use. *Id.* at 321-22.

166. 17 U.S.C. § 1201(a)(1)(A) (2000).

circumvention from reaching outside the statutory subject matter of copyright.[167] Second, section 1201(f) limits anti-circumvention indirectly by allowing reverse engineering for only limited purposes.

1. One Difficulty and Danger of the DMCA

One difficulty and danger of the DMCA is that it fails to address what happens when an anti-circumvention technology is used to control access to both proper and improper copyright subject matter. For example, copyright protects movies and video game content,[168] but does not protect the method of operation of the movie player or game console.[169] Two important questions are: (1) what if you cannot separate the content from the platform;[170] and (2) what if analyzing the method of operation of the platform necessarily entails circumventing copyright protection technology? These questions were asked and answered in the video game context by *Atari*, *Accolade*, and *Connectix.* The answer, consistent with section 102(b), was that copyright could not be used to block access to the platform. These cases were decided before the DMCA and the questions must be asked and answered again.

2. The Act of Circumvention and Anti-Circumvention Technology

The anti-circumvention requirements that trigger the DMCA are easily satisfied. The anti-circumvention provisions can be applied when a person "circumvents a technological measure that effectively controls access to a work protected under this title."[171] There are two key ideas: the act (what has to be done to count as circumvention) and the technology (what counts as an anti-circumvention device). The statutory definition of the act of circumvention amounts to bypassing the technological measure without the copyright owner's permission.[172] The technological measure that counts as an anti-circumvention device also

167. Six years before the DMCA was legislated, the Federal Circuit declared that "[a]n author cannot acquire patent-like protection by putting an idea, process, or method of operation in an unintelligible format and asserting copyright infringement against those who try to understand that idea, process, or method of operation." *Atari I*, 975 F.2d at 837.

168. 17 U.S.C. § 102(a)(6) (2000).

169. *Id.* at § 102(b).

170. That is, what if the media and the player interact in such a way that it is impossible to draw a line between the two?

171. *Id.* at § 1201(a)(1)(A).

172. "[T]o 'circumvent a technological measure' means to descramble a scrambled work, to decrypt an encrypted work, or otherwise to avoid, bypass, remove, deactivate, or impair a technological measure, without the authority of the copyright owner." *Id.* at § 1201(a)(3)(A). This is a kind of digital trespass.

amounts to nothing more than the permission of the copyright owner.[173] Both hinge on having or not having permission. Significantly, this means circumvention violations can be brought against anyone who accesses[174] digital work, in a manner that is not in the ordinary course of operation,[175] without the permission of the copyright owner. Thus, circumvention as outlawed by the DMCA is almost indistinguishable from our first definition of reverse engineering.[176]

3. Reverse Engineering Restrictions

Reverse engineering is written into the DMCA as a subsection of, and exception to, the circumvention prohibitions.[177] When copyright can be used to prevent access to functional elements or methods of operation, copyright risks going beyond its statutory subject matter. It is here that reverse engineering provides a necessary "safety valve."[178] However, the text of the DMCA limits and qualifies reverse engineering to the point where it is questionable if it exists as a useful option at all.

The DMCA limits reverse engineering by restricting the act, the means, and the publication of results. The act is limited by who can do the reverse engineering ("a person"), what their purpose must be ("for the sole purpose of . . . interoperability"), how much they may reverse engineer (only the "elements of the program that are necessary"), what kind of devices their results must be directed toward ("an independently created computer program"), what kind of information they are allowed to look for (only information that has "not previously been readily available to the person"), and how to do the work (such that the acts "do not constitute infringement under this title").[179]

The DMCA also restricts the means that can be used (only those "necessary to achieve such interoperability"), and how the means may be used (such that they "do not constitute infringement under this title").[180]

The DMCA further restricts the publication of the reverse engineering results by who can publish ("the person" who did the reverse

173. "[A] technological measure 'effectively controls access to a work' if the measure, in the ordinary course of its operation, requires the application of information, or a process or a treatment, with the authority of the copyright owner, to gain access to the work." *Id.* at § 1201(a)(3)(B).

174. The word "access" is this phrase is an attempt at a neutral way of saying view, listen to, play, examine, or use content which had been scrambled, encrypted, or somehow protected.

175. 17 U.S.C. § 1201(a)(3)(B) (2000).

176. *See supra* note 1.

177. Title 17 Section 1201 is entitled "Circumvention of copyright protection systems". Subsection (f) provides "Reverse engineering" exceptions to "infringement under this title". 17 U.S.C. § 1201 (2000).

178. Universal City Studios v. Reimerdes, 111 F. Supp. 2d at 322 (S.D.N.Y. 2000).

179. 17 U.S.C. § 1201(f)(1) (2000).

180. *Id.* at § 1201(f)(2).

engineering), why they can publish ("solely for the purpose of enabling interoperability"), what kind of interoperability the publication must be directed toward ("an independently created computer program"), and how the publication may occur (such that publication does "not constitute infringement under this title or violate applicable law other than this section").[181]

At a minimum, these restrictions turn reverse engineering into a crime if your purpose is anything other than interoperability with an independently created computer program.[182] Even then the restrictions make reverse engineering a dangerous practice.[183] The following case, *Universal City Studios v. Reimerdes*, although not about video games, presents an early interpretation of a few of the reverse engineering provisions of the DMCA.[184]

B. Universal City Studios v. Reimerdes

In late September 1999, Jon Johansen, a 15-year-old Norwegian, and two others reverse engineered a licensed DVD player and discovered the keys to the encryption algorithm.[185] They used this information to create DeCSS,[186] a small program that decrypts DVD movies so they could play them on their own Linux player.[187] In November 1999, the defendants, including Reimerdes, posted DeCSS on the Internet for download and also provided links to other DeCSS-based software.[188] Universal City Studios sued Reimerdes under the DMCA's anti-circumvention sections. This case primarily deals with section 1201(a)(2) of the DMCA, the anti-trafficking provision that "bans offering or providing technology that may be used to circumvent technological means of controlling access to copyrighted works."[189] Reimerdes raised a number of defenses, all of which failed. The defense of fair use, which "has been viewed by courts as a safety valve" to temper copyright rights,[190] failed because fair use is a defense to copyright infringement, and

181. *Id.* at § 1201(f)(3).

182. "A 'computer program' is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." *Id.* at § 101.

183. It is not clear at this point, what kind of useful reverse engineering is allowed.

184. *See* Universal City Studios v. Reimerdes, 111 F. Supp. 2d. 294.

185. *Id.* at 311. Jon Johanson was recently tried in Norway and cleared of all charges. *See DECSS Author Jon Johansen found Innocent in Norwegian Court*, 2600 NEWS (Jan. 7, 2003), *at* http://www.2600.com/news/view/article/1485. Johansen's lawyer is quoted saying "when you have bought a film legally, you have access to its content, [i]t is irrelevant how you get that access. You have bought the movie after all." *Id.*

186. DVDs are encrypted with an algorithm called CSS.

187. Universal City Studios v. Reimerdes, 111 F. Supp. 2d at 311.

188. *Id.* at 312

189. *Id.* at 319.

190. *Id.* at 322.

defendants "are not here sued for copyright infringement."[191] They were sued for anti-circumvention trafficking, not copyright infringement, and the court held that Congress provided no fair use defense to this action.[192]

The defense based on the reverse engineering provisions of the DMCA also failed. Defendants contend that DeCSS is necessary to achieve interoperability and create a DVD player on Linux.[193] The court found that section 1201(f)(3) allows only the individual who did the reverse engineering, i.e. Jon Johansen, to make DeCSS available to others.[194] And even then, the court speculated that Johansen could not post DeCSS because there would be other possible uses for the code, and therefore the posting would not be done "solely to achieve interoperability with Linux or anything else."[195]

Reimerdes shifts power to copyright holders by reading the anti-circumvention provisions of the DMCA to eliminate fair use and eviscerate it's reverse engineering allowances.

C. Video Game Reverse Engineering Under the DMCA

The DMCA will hurt the video game industry by curtailing reverse engineering. The DMCA's anti-circumvention provisions can be easily brought to bear on any digital work because of the ease with which anti-circumvention technology can be wrapped around that work. The DMCA does not require that digital works be well protected—simple scrambling of data is enough. *Reimerdes* found that even a "weak cipher" effectively controls access to copyrighted works.[196] The *Reimerdes* test for what counts as an access control device is circular, and easily met: "if its function is to control access," then it effectively controls access.[197]

Once anti-circumvention technology is added, that work can only be reverse engineered for the narrow purpose of interoperability defined in section 1201(f) of the DMCA. As discussed above, reverse engineering in the video game field is used for more than just

191. *Id.* at 322.
192. Universal City Studios v. Reimerdes, 111 F. Supp. 2d at 322.
193. *Id.* at 320. There was no Linux-based DVD player at the time.
194. *Id.* at 320.
195. *Id.* The court reasoned that because DeCSS runs on Windows machines, DeCSS could not be used solely for Linux interoperability. *Id.* This is a dangerous idea because algorithms, source code, and programs can be compiled and run on any number of platforms and by their nature are rarely locked into a single platform.
196. *Id.* at 317. The court in *RealNetworks, Inc. v. Streambox, Inc.*, found a preliminary "Secret Handshake" qualified as a DMCA anti-circumvention device. 2000 U.S. Dist. LEXIS 1889 (W.D. Wash. 2000).
197. Universal City Studios v. Reimerdes, 111 F. Supp. 2d at 318.

interoperability—reverse engineering opens up both horizontal and vertical access and enables technological, design, and user interface leap-frogging.[198] This leap-frogging produces fast paced innovation that has created an industry where no single company has remained dominant.[199] Great games drive the industry, and the next great game can come from anywhere. Some games are successful because they push technology forward, other games are successful because they push a design forward. The industry is better because this generation of games improved directly on the previous generation. The legal environment plays an important role in keeping the industry competitive and growing. The DMCA stifles reverse engineering and will slow the industry's growth.

Also troubling is the possible ability of the DMCA to extend protection to un-copyrightable elements. Copyright protection does not extend to the ideas, processes, procedures, and some of the other elements in a work.[200] However, with the addition of simple anti-circumvention technology, the DMCA might be used to block access to unprotected elements much the same as Cohen and Lemley warn patents could be used to block access to "unpatented components."[201]

To add to this trouble, compare older video game hardware[202] to Loffredo's new hardware, discussed above, which uses programmable logic parts.[203] This new technology creates a problem: that which has historically been tangible hardware, visible to the eye, susceptible to probing, experimentation, and included many un-copyrightable elements,[204] is now entirely a software "bitstream"[205] which is easily scrambled and wrapped in anti-circumvention technology. Should hardware be subjected to different copyright treatment because it is programmable? According to section 102 of the Copyright Act, the

198. Successful game designs and user interfaces quickly become widespread and improved on.

199. *Changing the game*, THE ECONOMIST, Dec. 6th, 2003, *available at* 2003 WL 58585083 (since its inception, the industry has operated in approximately 5 year cycles with no single manufacturer or game maker able to hold onto the top spot). Conversely, the DMCA could aid industry consolidation.

200. "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." 17 U.S.C. § 102(b) (2000).

201. Cohen & Lemley, *supra* note 13, at 26.

202. Older hardware contained many standard commodity parts such as resistors, transistors, capacitors, and standard integrated circuits.

203. Loffredo, *supra* note 73.

204. Although some elements of older hardware have been protected under copyright, "mask works," for example, many standard parts such as resistors, transistors, capacitors, and standard integrated circuit chips have not been subject to copyright protection.

205. *Id.*

answer is no.[206] Arriving at that result applying the DMCA is sure to be a difficult task.

The DMCA has undone the reverse engineering allowances of *Atari*, *Accolade*, and *Connectix*, where the court carefully balanced copyright holder's rights, fair use, and competition. The DMCA creates a new cause of action, anti-circumvention, that trumps fair use, avoids balancing, and ignores competition. *Atari*, *Accolade*, and *Connectix* articulate a pro-competitive reverse engineering doctrine that allows access to platforms and the ability to create alternative compatible platforms. The DMCA wipes out this balance.

CONCLUSION

Reverse engineering is used by game designers, software and hardware engineers, and is essential to the health of a competitive video game industry. The DMCA's anti-circumvention provisions upset the balance between the rights of copyright holders, fair use, and competition. Because of the importance of reverse engineering, Congress should amend the DMCA to expand allowances for reverse engineering practices.

206. 17 U.S.C. § 102(b) states that un-copyrightable subject matter can not be made into copyrightable subject matter by changing "the form in which it is described, explained, illustrated, or embodied."

APPENDIX A
UNDERSTANDING CODE

This appendix is presented to illustrate the connection between source code, object code, and disassembled object code.

1) Here is a simple program, written in C. This is source code. Source code is normally not shipped with a product and is often carefully guarded. If run on most computers, this program would print the message "hello, world":

```
/*  This is the "first program" you ever write.  */
/*  Brian W. Kerninghan and Dennis M. Ritchie,*/
/*  The C Programming Language page 5 (2d ed. 1998). */
#include <stdio.h>
main()
{
        printf("hello, world\n");
}
```

2) Here is an example of the object code generated from the above source code. This is what is shipped in games. This is the kind of code (although for a different system) that reverse engineers deciphered in *Sega v. Accolade* and *Sony v. Connectix*. Most consider this, incorrectly, to be 1s and 0s and unreadable. The last line is data that most programmers can read. It contains the words "hello, world." For example, 68 = 'h' 65 = 'e' 6C = 'l' etc.

```
00401010  55  8B  EC 83  EC 40  53  56  57  8D 7D C0 B9  10  00  00  00  B8
00401022  CC CC CC CC F3  AB 68  1C 00  42  00  E8 2E 00  00  00  83  C4
00401034  04  5F  5E  5B  83  C4 40  3B  EC E8  9E 00  00  00  8B  E5  5D C3
0042001C 68  65  6C 6C  6F  2C 20  77  6F  72  6C 64  0A 00  00  00  00  00
```

3) Here is the disassembly of that same object code. The disassembler has taken the object code and reformatted it to improve its readability. All programmers who know x86 assembler can read and understand this code. This is the same program as in 1) and 2).

```
00401010    push    ebp
00401011    mov     ebp,esp
00401013    sub     esp,40h
00401016    push    ebx
00401017    push    esi
00401018    push    edi
00401019    lea     edi,[ebp-40h]
0040101C    mov     ecx,10h
00401021    mov     eax,0CCCCCCCCh
00401026    rep     stosdword ptr [edi]
00401028    push    offset string "hello, world\n" (0042001c)
0040102D    call    printf (00401060)
00401032    add     esp,4
00401035    pop     edi
00401036    pop     esi
00401037    pop     ebx
00401038    add     esp,40h
0040103B    cmp     ebp,esp
0040103D    call    __chkesp (004010e0)
00401042    mov     esp,ebp
00401044    pop     ebp
00401045    ret
```