AMERICAN COMMITTEE FOR
INTEROPERABLE SYSTEMS
2550 Garcia Avenue
Mail Stop PAL1-521
Mountain View, California 94043

(415) 336-2482

UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

SEGA ENTERPRISES LTD., a Japanese corporation, Plaintiff, v. ACCOLADE, INC., a California corporation, Defendant.	NO. No. C 91 3871 BAC BRIEF *AMICUS CURIAE* OF AMERICAN COMMITTEE FOR INTEROPERABLE SYSTEMS Date: March 13, 1992 Time: 2:30 p.m. Place: Courtroom of the Honorable Barbara A. Caulfield
ACCOLADE, INC., a California corporation, Counterclaimant, v. SEGA ENTERPRISES LTD., a Japanese corporation, and SEGA of AMERICA, INC., Counterdefendants.	OF COUNSEL: Peter M.C. Choy, Deputy General Counsel, Sun Microsystems, Inc. Norton Cutler, Assistant Chief Counsel, Litigation, NCR Corporation Gregory Handschuh, Vice President, Legal Affairs, Amdahl Corporation John S. Solakian, Corporate Patent Counsel, Bull HN Information Systems Inc. W. Russell Wayman, Corporate Vice President and General Counsel, Storage Technology Corporation

Page

TABLE OF CONTENTS

	Page
INTEREST OF AMICUS CURIAEAMERICAN COMMITTEE FOR INTEROPERABLE SYSTEMS	1
ARGUMENT	3
A. Interoperability Is Critical To The Computer Industry.	3
B. Interim Copying Is A Permissible Step In Reverse Engineering Software To Achieve Interoperability.	4
C. Prohibiting Interim Copying Would Effectively Eliminate The Ability To Determine The Unprotected Ideas Underlying Software In Contravention Of The Idea-Expression Dichotomy And The Underlying Purposes Of Copyright Law.	7
CONCLUSION	11

Page(s)

TABLE OF AUTHORITIES

Cases

Baker v. Selden, 101 U.S. 99, 102 (1879) 10

Bonito Boats, Inc. v. Thunder Craft Boats, Inc., 489 U.S. 141 (1991) 1, 9

Consumers Union of the United States, Inc. v. General Signal Corp., 724 F.2d 1044, 1050 (2d Cir. 1983), *reh'g denied*, 730 F.2d 47 (2d Cir. 1984), *cert. denied*, 469 U.S. 823 (1984) 8

E.F. Johnson Co. v. Uniden Corp. of America, 623 F. Supp. 1485 (D.C. Minn. 1985) 2

Feist Publications, Inc. v. Rural Telephone Service Co., 111 S.Ct. 1282 (1991) 1, 9

Harper & Row Publishers, Inc. v. Nation Enterprises, 471 U.S. 539, 549 (1985) 9

Kern River Gas Transmission Co. v. Coastal Corp., 899 F.2d 1458, 1463 (5th Cir. 1990), *cert. denied*, 111 S.Ct. 374 (1990) 1

Kewanee Oil Co. v. Bicron, 416 U.S. 470, 476 (1974) ... 5

NEC Corp. v. Intel, 10 U.S.P.Q. 1177 (N.D. Cal. 1989) 2

Vault Corp. v. Quaid Software, Ltd., 847 F.2d 255 (5th Cir. 1988) 9

Statutes

17 U.S.C. § 102(b) 3, 7

17 U.S.C. § 107 8

17 U.S.C. § 117 8

35 U.S.C. §§ 101-102 10

Other Authorities

Last Frontier Conference Report on Copyright Protection of Computer Software, 30 Jurimetrics J. 15, 24-25 (1989) 2

CCH Copyright Law Rptr. ¶ 15,797 7

Page(s)

P. Goldstein, *Copyright*, ¶ 5.1.1.4 at 88-90 (1991 Supp.) 2

INTEREST OF *AMICUS CURIAE*

AMERICAN COMMITTEE FOR INTEROPERABLE SYSTEMS

The American Committee for Interoperable Systems ("ACIS")[1] is an informal organization of companies that develop innovative software and hardware products that interoperate[2] with computer systems developed by other companies. The members of ACIS believe that computer programs deserve strong intellectual property protection to provide developers with sufficient incentives to create new computer programs. At the same time, ACIS members are concerned that the improper extension of intellectual property protection generally, and copyright law in particular, will impede innovation and inhibit fair competition in the computer industry. ACIS therefore seeks the adoption of legal standards that reflect copyright law's long-established "balanc[ing] of] the competing concerns of providing incentive to authors to create and of fostering competition in such creation." *Kern River Gas Transmission Co. v. Coastal Corp.*, 899 F.2d 1458, 1463 (5th Cir. 1990), *cert. denied*, 111 S.Ct. 374 (1990). *See also Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*, 489 U.S. 141 (1989) and *Feist*

---

[1]The members of ACIS include: Amdahl Corporation, Bull HN Information Systems Inc., Chips and Technologies, Inc., Clearpoint Research Corporation, Comdisco, Inc., Emulex Corporation, Forecross Corporation, Informix Software, Inc., Johnson-Laird, Inc., Kapor Enterprises, Inc., Landmark Systems Corporation, NCR Corporation, Phoenix Technologies, Ltd., Seagate Technology, Inc., Software Association of Oregon, Storage Technology Corporation, Sun Microsystems, Inc., Systems Center, Inc., 3Com Corporation, Unisys Corporation, and Zenith Data Systems Corporation.

[2]The term "interoperability," as used in the computer industry and in this brief, refers to the need for the software or hardware of one manufacturer to operate with that of another manufacturer.

Publications, Inc. v. Rural Telephone Service Co., 111 S.Ct. 1282 (1991).

ACIS is concerned that certain of the legal assertions made by the plaintiff, Sega Enterprises Ltd. ("Plaintiff"), in its Memorandum in Support of Motion for Preliminary Injunction are antithetical to this careful balancing. Specifically, ACIS disagrees that interim copies made during reverse engineering of computer programs to determine how they interface with computer hardware and other software is copyright infringement *per se*, without a showing of substantial similarity between the original computer programs and the allegedly infringing commercial end product. Plaintiff's proposition ignores the unique characteristics of computer programs which must be taken into account when applying the traditional legal doctrines of fair use and the idea-expression dichotomy in the computer context.

ACIS has reviewed the briefs of both parties to this lawsuit and the cases cited in those briefs. ACIS believes that the legal analysis on the issue of interim copying set forth by the defendant, Accolade, Inc. ("Defendant"), in its Opposition to Plaintiff's Motion is an accurate interpretation of existing case law, statutes and scholarly treatment of the issue, and supports the conclusion that interim copying is permissible under current copyright laws. *See e.g., NEC Corp. v. Intel*, 10 U.S.P.Q. 1177 (N.D. Cal. 1989); *E.F. Johnson Co. v. Uniden Corp. of America*, 623 F. Supp. 1485 (D.C. Minn. 1985); P. Goldstein, *Copyright*, ¶ 5.1.1.4 at 88-90 (1991 Supp.); *Last Frontier Conference Report on Copyright Protection of Computer Software*, 30 Jurimetrics J. 15, 24-25 (1989). For present purposes, ACIS takes no position as to the appropriate

application of the legal analysis put forth by Defendant to the facts of this case.[3] The purpose of this brief is to set forth the policy arguments which strongly support a conclusion that the making of interim copies in order to read, study and analyze computer programs should not be, and is not, a violation of the copyright laws.

ARGUMENT

A. Interoperability Is Critical To The Computer Industry.

ACIS was born out of the recognition of its members that the life blood of the computer industry is the ability of any competitor to make and distribute interoperable software and hardware. Computer programs function only by interacting with the computer environment in which they are meant to operate. This environment is extremely unforgiving. Unless the computer program conforms to the precise rules for interacting with the other elements of the system, no interaction between the program and the system is possible. These rules of interaction--or interface specifications, in computer parlance--are essential to interoperability between the program and the system. Such underlying rules are unprotectable elements under 117 U.S.C. § 102(b).

Nothing in the copyright laws requires the author of a computer program to have a license from the hardware manufacturer in order to publish programs which operate with that hardware. If the

---

[3] ACIS also takes no position with respect to any issue raised in this litigation that is not specifically addressed by this brief.

developer of the first interoperable program for a particular computer system could use copyright law to prevent all subsequent developers from writing any programs for that system, the first developer could severely, if not entirely, curtail the creation of competing products. Similar monopolies would exist everywhere computer programs are used in a computer system, e.g., applications software, systems software such as the operating system, and the software embedded in peripherals such as terminals, printers and memory devices. Such a broad monopoly is not consonant with the purpose of the copyright laws, and would impede the development of interoperable products and the free flow of the ideas underlying the software. In sum, overprotection would stifle innovation, stunt legitimate competition by other firms and, through monopoly pricing, impose heavy costs on users.

B. Interim Copying Is A Permissible Step In Reverse Engineering Software To Achieve Interoperability.

The unprotected ideas underlying the vast majority of traditional copyrighted literary works, such as plays and novels, are readily apparent simply from viewing or reading the works in the form in which they are published. Computer programs, on the other hand, are published in a form which, though understandable to the hardware on which such programs operate, is not practically comprehensible to humans. Publishers of computer programs for particular hardware generally do not release to the public all of the information necessary for an understanding of how the programs interoperate with the hardware. Nor can such an understanding be gleaned from merely watching the program operate. Thus, the ideas underlying the program are not cognizable, as a practical matter,

without further effort by the programmer to transform the program into a readable form.

To obtain access to the underlying ideas for purposes of efficiently and competitively designing interoperable programs, programmers must "reverse engineer"[4] the commercially distributed versions of computer programs and study the data revealed through such reverse engineering to obtain an understanding of how the programs operate with the target system. There are several means of reverse engineering computer programs. Some of these means involve observation of the program's behavior from the outside and can be performed by loading, running or viewing the program on a computer. However, in order to achieve interoperability, it is often necessary to make intermediate copies of the program in order to transform it into human readable form and study the code and its structure.

Programmers write software using "source code", high level computer languages which use alphanumeric characters understandable to humans. This source code is then "assembled" or "compiled" into binary or machine language, often called "object code", which consists of nothing more than a series of "1's" and "0's". To read and understand the object code, it is necessary to convert it from the binary form in which it is stored into alphanumeric characters

---

[4] "Reverse engineering" is the practice of examining a computer program to find out how it functions and how it is designed, so that other products can be designed for use either in conjunction with it, or as a replacement for it. Reverse engineering has been expressly acknowledged as "a fair and honest" method of discovering information regarding how a product is made or works. *See Kewanee Oil Co. v. Bicron,* 416 U.S. 470, 476 (1974).

which can be read and understood by humans.[5] Since many programs are thousands of lines long and the typical monitor screen only shows forty lines or less, it is nearly always necessary to print the converted version of the program out on paper and analyze and reformat it, all in an effort to understand the functions of the code, and the ideas and other information necessary to design an original program which interacts with the same hardware or software as the studied program.[6]

---

[5] The conversion process is complicated by the fact that when source code is assembled into object code, much of the original explanatory information that was written into the source code for use by the programmer to explain what the program is doing, but which is not essential to the computer, is eliminated from the program. Thus, when object code is converted, either through disassembly or some other means, the programmer obtains only "assembly language" instructions, which are at a lower level than source code and which direct the computer at its most basic level (e.g., telling the computer to move data from one register or storage location to another ("MOV A, X"), or to add ("ADD A, B") numbers). The reverse engineer then must study and experiment with the assembly language instructions in an attempt to divine, without the assistance of the original programmer's source code comments, labels and notes, what the program, as a whole, is doing.

[6] Plaintiff seems to suggest that as an alternative to making interim copies, it is possible to remove the top from a semiconductor chip in order to read the contents of it. Chemically peeling the top off a memory chip, however, merely exposes the silicon chip topography to view. In order to obtain the contents of the circuit so exposed, it is necessary to photograph the circuit, "read" the state of the transistors and capacitors to decide which represent 1's and which 0's, decode the order in which they are stored, write down and type into a computer the thousands and thousands of resulting 1's and 0's, disassemble the list of 1's and 0's into assembly language code statements, and then study and examine the assembly language to determine what the program, as a whole, is doing. Thus, peeling the chip does not eliminate the need to make interim copies.

C. **Prohibiting Interim Copying Would Effectively Eliminate The Ability To Determine The Unprotected Ideas Underlying Software In Contravention Of The Idea-Expression Dichotomy And The Underlying Purposes Of Copyright Law.**

Rather than seeking to demonstrate substantial similarity between Defendant's programs and its own, Plaintiff instead argues that interim copies of Plaintiff's programs made solely to read and understand how such programs operate with the computer system constitute infringing copies in and of themselves, a *per se* violation of the copyright laws. Such a conclusion does not comport with the underlying purposes of the copyright laws, and expands the scope of the protection afforded by the copyright laws to computer programs far beyond constitutional and statutory limits.

The idea-expression dichotomy--that is, the proposition that copyright protection extends only to the expression of ideas and not to the ideas themselves--is fundamental to copyright law. *See* 17 U.S.C. § 102(b) ("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. . . ."). It also is axiomatic, as explained above, that because of the form of the expression in which they are published, computer programs must, in all but the simplest or most exhaustively documented instances, be reverse engineered simply to reach the ideas underlying the expression.

The reverse engineering process, itself, does not take anything from the work being examined other than ideas and functions embodied in the work which are not protected by copyright law.[7] Use

---

[7]The House Report accompanying the 1976 copyright law states:

[Footnote continued on next page]

of this process does not justify using the protected expression of the examined work in another commercially distributed work. However, consistent with the copyright laws, the reverse engineer may extract from the copyrighted expression, by the study of its ideas and how it functions, the basis from which to make his or her second work function. So long as the actual expression of the second work is not substantially similar to that of the first work, there is no copyright infringement.

Whether the making of interim copies is justified on the basis of the copyright doctrine of fair use set out in 17 U.S.C. § 107;[8] or as an adaptation allowed under 17 U.S.C. § 117; or on the

---

[Footnote continued from previous page]

> "Some concern has been expressed lest copyright in computer programs should extend protection to the methodology or processes adopted by the programmer, rather than merely to the 'writing' expressing his ideas. Section 102(b) is intended, among other things, to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law."
>
> H.R. 94-1476, as reprinted in CCH Copyright Law Rptr. ¶ 15,797.

[8]The fair use doctrine is applicable in the context of reverse engineering despite the fact that the ultimate goal of reverse engineering may be to develop a commercial product. The fact that the ultimate goal of writing a novel may be to profit from its sale does not prohibit the author of the novel from making "fair use" copies of reference sources used in writing the novel. Similarly, the reverse engineering of a copyrighted computer program does not by itself impact on the market for the program, nor does it take away revenue from the sale of the program. It is only the fact of development and marketing of a new, original program, in competition with the first, that has an indirect commercial effect. As the Second Circuit pointed out in *Consumers Union of the United States, Inc. v. General Signal Corp.*, 724 F.2d 1044, 1050 (2d Cir. 1983), *reh'g denied*, 730 F.2d 47 (2d Cir. 1984), *cert. denied*, 469 U.S. 823 (1984), "[t]he Copyright Act was

[Footnote continued on next page]

basis of the public policy that requires that others be able to read and use copyrighted works to extract the information contained in them,[9] the result is the same. It is not illegal *per se* to make and study interim copies of computer programs to achieve interoperability. This result is dictated by the traditional idea-expression dichotomy which is fundamental to copyright law, as well as practical necessity in the computer context.

Recent cases teach that, since all forms of intellectual property protection are a species of government-granted monopoly, such protection must not exceed exactly that set out in the laws. *Bonito Boats*, 489 U.S. at 168 ("offering patent-like protection for ideas . . . conflicts with the 'strong federal policy favoring free competition'") (citations omitted); *see also Vault Corp. v. Quaid Software, Ltd.*, 847 F.2d 255 (5th Cir. 1988) (copyright law prohibits the taking of the expression of an idea, but does not

---

[Footnote continued from previous page]

not designed to prevent . . . indirect negative effects of copying."

[9]As the Supreme Court stated recently in *Feist*:

> The primary objective of copyright is not to reward the labor of authors, but '[t]o promote the Progress of Science and useful Arts.' [U.S. Const.] Art. I, § 8, cl. 8. To this end, copyright assures authors the right to their original expression, <u>but encourages others to build freely upon the ideas and information conveyed by a work</u>.

111 S. Ct. at 1290 (citations omitted) (emphasis added). *See also Harper & Row Publishers, Inc. v. Nation Enterprises*, 471 U.S. 539, 549 (1985), citing H. Ball, Law of Copyright and Literary Property 260 (1944) ("author's consent to a reasonable use of his copyrighted works has always been implied by the courts as a necessary incident of the constitutional policy of promoting the progress of science and the useful arts").

grant a monopoly on the use of the unprotected idea). Increasing the scope of protection increases the extent of the monopoly, without concomitant benefit to society. Were this Court to rule that it is an actionable infringement to make copies in order to translate the machine language version of computer programs, the original programmer would in effect be granted a patent on the program, but without the safeguards implemented in the patent system to avoid stifling competition and the free flow of ideas.[10] Protecting the art, as well as the expression of the art, under copyright law "would be a surprise and a fraud upon the public." *Baker v. Selden*, 101 U.S. 99, 102 (1880). Prohibiting interim copying as a means of examining copyrighted programs has the anticompetitive effect of a patent, without the balancing access to the ideas underlying the program to further competitive development of innovative and more cost-effective programs.

CONCLUSION

The principle of interoperability goes to the essence of the free enterprise system, furthering free competition in the computer industry. Because overprotection of computer programs causes substantial industry and consumer welfare costs, courts must interpret intellectual property laws in the computer context to balance the need for strong protection of an author's works with the

---

[10] Such safeguards include the requirement that a patent applicant establish to the satisfaction of the Patent Office that its system is new, novel, useful and inventive, and disclose the best mode of operation of the invention. 35 U.S.C. §§ 101-102, 111-112. In addition, the claims in a patent provide clearer guidance as to the scope of the monopoly granted, whereas the protection afforded by a copyright is subject to the uncertainties of judicial interpretation.

need to further the ability of any third party to make and market computer products for use on or with the products of another company.

Prohibiting the making of interim copies would grant *de facto* protection to unprotected ideas and functional elements concealed within the object code of computer programs. Without reverse engineering, no one other than the copyright holder would have any lawful way of discovering and permissibly using the ideas and unprotected elements. This Court should affirm that the copyright laws permit interim copying in order to reach underlying ideas and to achieve interoperability. Granting one company the exclusive right to develop interoperable computer programs for a particular computer system will lead to higher prices, reduced output, and less innovation, not to the promotion of "the Progress of Science and the useful Arts."

DATED: March 10, 1992

Respectfully submitted,

AMERICAN COMMITTEE FOR
INTEROPERABLE SYSTEMS

By: _____
Peter M.C. Choy, Esq.
Chairman

2550 Garcia Avenue
Mail Stop PAL1-521
Mountain View, CA 94043