Read_Me Festival 1.2 Jury Statement
Amy Alexander, Cue P. Doll, Florian Cramer, RTMark, Alexei Shulgin.

The jury voted to award prizes to three projects: DeskSwap, ScreenSaver, and Textension. The term "software art" is a decidedly broad category, and each of the awarded projects takes a very different approach to it. The festival guidelines originally called for the awarding of first, second and third prizes. However, the jury felt that ranking such disparate projects with respect to one another would be artificial. Therefore, in recognition of the fact that "software art" is not simply one genre but encompasses a variety of approaches, the jury has decided to dispense with the rankings and award each of the three selected projects equivalent prizes. Since read_me 1.2 is one of the pioneering festivals of software art we felt it necessary to open up the field rather than to prematurely narrow it down. We consider software art to be art whose material is algorithmic instruction code and/or which addresses cultural concepts of software. For us this implies not restricting software art to PC user applications, nor even just to executable machine code. Each of the three winning projects fits our concept of software art in a different way. Since we wanted to communicate the scope and potential of software art as broadly as possible, we gave, in addition to the three prizes, a total of five honorary mentions: to Re (ad.htm, Tracenoizer, Carnivore, Portret of President and WinGluk Builder. It should be said that very few of the pieces submitted had any political or activist usefulness, although several pretended to. While the jury appreciated the diversity of the works entered, we were somewhat dismayed by the scarcity of political content.

Winners: in alphabetical order

DESKSWAP by Mark Daggett (http://www.deskswap.com)
is a program that critically considers the problem of the standardization of personal computer users' workspaces. It allows you to compare your desktop with desktops of other people, living in different countries and speaking unknown languages. Each time you get terrified by the consequences of globalization that manifest themselves in the predetermined aesthetic solutions of your surroundings: sofa from IKEA, wallpaper from Microsoft. The voyeuristic aspect of the project provides a certain relief, which you experience looking at other people's desktops: everything is ok, people are using their computers for the same rubbish as you - same programs, same files and folders. But - maybe "serious" users just don't have time to play around with strange programs like Deskswap? Deskswap is made in a very simple and elegant way; it doesn't pretend to be more than it is. It is effective, interesting and very user-friendly. The program is used with great pleasure by "normal" people (not just by media art curators). That's because Deskswap offers the possibility of communication in our time of global alienation.

SCREEN SAVER by Eldar Karhalev and Ivan Khimin
(http://www.macros-center.ru/read_me/inde5.htm#screen)
Of the three awarded pieces, "Screen Saver" is the most challenging to the concept of software and software art. At first glance, it doesn't seem to be software in its own right. The piece consists of a simple step-by-step instruction for configuring the screen saver of the Microsoft Windows operating system. As a result, the PC is turned into a display of a giant rectangle which slowly moves from the left to the right corner of the screen and back, slightly modulating its color in the process. This is a simple, elegant and beautiful piece. It could be called a black square of digital art, but that wouldn't explain why it is interesting as software. "Screen Saver" is software in at least two respects: On the one hand, it shows that software art can be post- or meta-software which, instead of being coded from scratch, manipulates existing software, managing to turn it upside down even without much technical sophistication. It reprograms Windows without employing programmer's skills. On the other hand, its formal instruction for misconfiguring the software is itself a software code. "Screen Saver" thus shows that software doesn't have to be written in computer programming languages. In an age of code abundance thanks to personal computers and the Internet, Software Art no longer needs to design algorithms from scratch, but can be

disassemblings, contaminations and tweaks of code found in the public. This makes contemporary software art distinct from the computer-generative art of the 1950s to 1980s. "Screen Saver" exemplifies this postmodern condition of software art in an almost paradigmatic simplicity. It brings up such questions as: Are there software readymades? Can non-programmers reprogram systems? Which does limit or extend which, and what does prevail in the end; the manipulation or the object manipulated, the artistic hack or Microsoft Windows?

Another proof of "Screen Saver" being software is the fact that, although curious for the jury, its original authors have split over different opinions and forked the codebase into two separate projects (similar to programs like Emacs and XEmacs). The second was entered under the name ".scr" to the competition; it differs from "Screen Saver" only in its instruction to choose a different font in the Windows screensaver setup. As a result, the rectangle doesn't slide from left to right, but bounces in all four directions. We found this result inferior to the more minimalist and hypnotic "Screen Saver". As in any program code, one changed instruction can make a big difference. We therefore feel it is justified that we award only "Screen Saver", not ".scr".
One final note: the jury noticed that "Screen Saver" breaks under Windows XP. The rectangle becomes much smaller and only bounces in the middle portion of the screen, thus destroying the effect. Like much great art, "Screen Saver" is a real period piece.

TEXTENSION by Joshua Nimoy
(http://www.macros-center.ru/read_me/inde10.htm#text)
In terms of aesthetic enjoyment, Textension is a clear winner. It is delightful, exciting, fantastic to play with. It points in many directions at once, suggesting that hypertext could be fun and beautiful and profound in all kinds of new ways that it isn't today. Interestingly, the way to this development is pointed out by the typewriter, which produced beautiful things through the physical action of metal. Textension is the first piece of software to pick up effectively this very lost thread.
Note: The jury is sad that mode #9 does not have a "save" feature, in which branching constructions could be stored by an author and reread by readers, in a perpetuation of the author/reader model of literature; zoom and rotate features would of course then also be nice.

HONORARY MENTIONS:

PORTRET OF PRESIDENT by Vladislav Tselicshev
(http://chells.boom.ru/)
It is a small application that installs a portrait of President Putin in an oval frame on the desktop of a computer user. The political and critical point of the project is obvious - the author proposes that you decorate your desktop (=workspace) the way Russian bosses have done for centuries: they decorate the walls of their offices with portraits of higher bosses to show their loyalty. The transfer of such loyal behaviour into the virtual sphere is logical; it's inhabited by the same humans with all their merits and shortcomings. Also, when a PC user customizes her desktop she tells about herself to the people around her. The jury would like to point out the simplicity and elegance of this work as well as the program's ease of use and its political orientation.

RE (AD.HTM by mez breeze (http://www.hotkey.net.au/~netwurker/)
An honorary mention goes to "Re (ad.htm" by the Australian artist mez, This entry created a lot of discussion in the jury, and quite dissimilar individual rankings and opinions. "Re (ad.htm" consists of a selection of writings or, to use the artist's terminology, "wurks" that had been posted to several net cultural and arts-related mailing lists. They are highly condensed pieces written in "mezangelle", an invented hybrid language which mixes syntactical snippets of programming languages, network protocols and markup code with the English language. The resulting texts can be read in multiple, often contradictory ways due to their elaborate use of ambiguity and compound ('portmanteau') words noted in rectangular brackets, thus resembling regular and Boolean expressions in commandline programs and programming languages. In contrast to a merely ornamental code chic, this hybrid language is used to expose and deconstruct the

epistemological politics engendered into seemingly "neutral", technical codes. It is poetically dense, involving and difficult, but also humorous. Of course, it is not technically executable code, although the bracketed expressions expand into multiple combinatory output sequences. But above all the mezangelle targets fictitious, fantastic compilers, creating a dream-like imagination of metonymic contiguity between human bodies and machines. Sure, this topic has been spelled out in popular culture and media theory multiple times, but mez succeeds to free it from all cyber-kitsch by tackling it from within, in structure.
"Re (ad.htm" of course provokes the question whether it can be legitimately considered software art even more than "Screen Saver". But it clearly is art whose material is formal instruction code and which addresses cultural concepts of software. Imaginary, pretended and otherwise broken or pseudo-code in fact has a long tradition in poetic software programming, starting with the Algol poems of the French Oulipo group in the 1960s and not ending with the Perl poetry popular among hackers since the early 1990s. In the non-digital realm, Russian Futurists, concrete and L=A=N=G=U=A=G=E poets approached programming code poetry when they experimented with the formal elements of conventional language. mezangelle, which historically departed rather from the net.art tradition of experimental ASCII art, differs from the
former in various respects: It neither is a concrete poetry-style conceptualist clean-room design of code, nor is it naive haikus or love poems like most Perl poetry. So mezangelle does for code poetry what 1990s net.art did for ASCII Art when it turns an idea that itself was brilliant, but carried out naively, into something contemporary and sophisticated.

TRACENOIZER by LAN (http://www.tracenoizer.org/)
Other projects have worked with the idea of introducing noise into surveillance processes for the purpose of allowing individuals to hide themselves. The actual effectiveness of such techniques is often questionable. Such was the case with TraceNoizer. As far as the jury can tell, TraceNoizer is not literally effective at introducing noise into our data identities; after several weeks we still couldn't find our data clones in search engines at all. TraceNoizer's interest to the jury, however, was its use of algorithmic processes as critique. In TraceNoizer, static data becomes a dynamic process; the omniscient search engine database is transformed into something like a video feedback loop. Each generation of TraceNoizer cloned webpages is fed back into itself and (at least in theory) back into the search engines, generating new pages that echo their originals - and their subjects - more vaguely with each successive generation. The noise added to the database is not external, but the search engine turned on itself. Search engines use exclusionary systems to determine and dictate data "relevance" - from Google's incestuous PageRank technology to other search engines' blatant payola practices. Given this fact, TraceNoizer's system of having data reproduce by looking up its own ass seems an appropriate and entertaining response.

WINGLUK BUILDER by CooLer
(http://www.macros-center.ru/read_me/inde6.htm#gluck)
WinGluk Builder belongs to a cracker culture of "revenge software," i.e. creating programs that affect the normal work of an operating system and give the impression that your computer is broken or infected by a terrible virus. Despite the saboteur character of the program the jury decided to nominate it for the following reasons:
- The program is focused on understanding the computer as an object with certain physical and aesthetic qualities and tries to reveal these qualities.
- It uses a computer against its purpose, overcoming the predetermination imposed by the pragmatic software creators.
- It takes a critical attitude towards hacker-cracker culture: using Wingluk Builder, everyone can feel like an impressive virus creator by pressing a couple of buttons.
-The project implies the possibility of integrating other "viruses" into the program (it has thorough instructions on how to do that)
- An attempt to create a community around itself.
- The project ironically comments on the interface of Windows applications - it looks exactly like a proper program with an uninstall feature, a help file and all the other features of a decent program that humorously

contradicts its own purpose.

- And last but not least: the program in fact is not that "evil" - it can't destroy your computer or erase your data. It rather gives you an opportunity to reflect on the possible results of hackers' activity, on the attention with which you should use your computer, as well as on the fact that your digital friend does not necessarily have to be a boring hybrid of a mailbox and a DVD player, but sometimes can perform strange and funny things.

- And the lat but not least: the program in fact is not that "evil" - it can't destroy your computer or erase your data. It rather gives you an opportunity to reflect on the possible results of hackers' activity, about attention with which you should use your computer, as well as about the fact that your digital friend does not necessarily have to be a boring hybrid of a mailbox and a DVD player but sometimes can perform strange and funny things.

CARNIVORE by RSG (http://rhizome.org/carnivore/)

Bosses currently use all kinds of elaborate software to spy on their workers. Products like MailCensor (http://www.mailcensor.com) encourage bosses to check for "unauthorized transmission of Email containing confidential data" and "provide a safe and productive work environment for employees, by filtering out offensive/inappropriate email from the Internet."

On some networks, software can be installed by users to spy on their bosses as well. Packet sniffers, used by systems administrators to diagnose network problems, can often be used or modifed to do just that. Some packet-sniffing software is expensive, some free:

http://www.tucows.com/, search on sniffer

http://www.softpile.com/search.phtml?query=sniffer&pp=10&in=title

The trouble is, most of this software wouldn't be easy for a non-technical user to convert into a tool for gathering useful information. Those products that are easy to use for corporate spying tend to have pricetags that are easy for bosses and companies to afford but not for employees.

Among currently available sniffing products, the jury likes Ethereal (http://www.ethereal.com), a free, cross-platform diagnostic tool that can be used fairly easily by employees to spy on their boss's e-mail, websurfing and other network communications.

An upcoming version of Rhizome's Carnivore is planned to make it easier for an art audience to get involved in corporate spying. The jury hopes it will do this. Since Carnivore is open source software, other people with the appropriate programming expertise can also write such modifications themselves. For now, Carnivore only runs on specialized servers, and it doesn't gather data in a human-readable form.

The relationship of Rhizome's Carnivore to the FBI's spying tool of the same name seems to be a matter of concept and hipness-value, but it is not explained and is not very obvious.