



CENTER
FUTURELAB
FESTIVAL
PRIX
ARCHIVES

Festival Archive
Prix Archive
People & Projects
Futurelab Archive
Center Archive
Picture Archive
Contact

Press Lounge
Publications
Sitemap
Network & Partners

English Deu
InfoCollecto
Search

Ars Electronica Archive → Catalog Archive



ENDO NANO [PART 01]

Software – Man and Milestones

→ **Ars Electronica 1992**

→ **Festival-Program 1992**

Back to:

→ **Festival 1979-2005**

>1992

[Benjamin Heidersberger](#)

It has rarely been the case in the entire history of mankind that a tool managed to win recognition as quickly as the personal computer did. The exhibition about the history and myths behind this wondrous machine and its heroes covers the time spanning from its infancy until its recognition as an indispensable working media. In the course of researching for the exhibition, we often encountered astonishment; history that wasn't even 20 years old was to be put on exhibition. What was even more astounding was the fact that even this history is already disappearing in the depths of self-fabricated company myths and journalistic ornamentation.

The exhibition is deliberately restricted to the personal computer and its software. Although developments with mainframes, mini and microcomputers have been repeated, it is the latter which is closest to us in its everyday occurrence and philosophy.

It remains to be seen as to whether the operability of the machines, which is becoming easier and easier, will ultimately lead to their disappearance in the sense of being absorbed into the ordinary. But, here you will now see milestones and machines once again.

Benjamin Heidersberger

A STAR IS BORN

When agents in Hollywood want to bring a star, a director and a producer together, they need a micro-film plot.

Time: 1984. Scene: pastel-coloured "we know each other" restaurant. The tables are allocated after months of waiting or thanks to being repeatedly mentioned in the important circles of society; European cuisine, Chef de Cuisine: Wolfgang I-have-forgotten-the-last-name; Palm trees indicate: We are in Hollywood.

The Agent and the Producer meet for a power lunch. We will save the offensive courtesies and the vital gossip.

Agent: Hacker! What a concept!

Producer: Hacker? Pac Man and Space Invaders? No science fiction films! Nobody wants it.

Agent (*disappointed*): No science fiction: science fact! Here are the articles: *New York Times*, *Time*, *Newsweek*. We are living in the age of the computer!

Producer (*bored*): Who wants to see spotty youths in front of a computer?

Agent: And if they save the world?

Producer (*annoyed*): Rubbish!

Agent: A computer freak is driving the world to the brink of an atomic war. Our computers and the Russian ones, too, are about to launch their rockets (gesticulates around in the air with his hands): Smoking rocket silos on the brink of

launching!

Producer: Nonsense, hackers only play Space Invaders.

Agent: The newspaper says that hackers are a danger to National Security.

Producer: Espionage is good, but we need a hero and a love story.

Agent: (*seeing a chance*): And a villain. Computers are good villains – and they don't ask for a million fee. How about: college kid wants to steal the very latest computer games via the telephone line and instead hacks into the computer system at the Pentagon. The super computer gets wind of an attack and opens the rocket silos. The army and the politicians are unable to hold back the talking electron brain – the world is facing Armageddon. But, just at the very last minute, the hacker breaks the code. The megabrain at the Pentagon realizes that it's all been just a simulation game, the rockets are deactivated. The end.

Producer: Good that it was the young hacker who revealed the weak spot of the computer and not the Russians or a terrorist.

Agent (*calling the waiter*): Bring me the cheque!

FROM THE PUNCH CARD TO THE PICTURE TUBE

Once upon a time there were computers like civil servants, puffed up pieces of apparatus which worked according to the book. Every input had to be first of all transcribed onto a form. System managers, absurd Kafkaesque warders, received the forms, fed the almighty machines with them and upon completion of the clerical work, conveyed the result back down to the impatiently waiting applicant.

The forms were generally small, pale-yellow coloured cards with a lot of holes which were able to precisely code all the corner data of the human life: surname, first name, address, sex, salary, bank account number – all a matter of the hole being in the right place.

And so the civil servant computers gobbled up piles of punch cards and made personnel administration, census, population counts or car registrations possible.

If Mr. or Mrs. Normal Person got to see a computer in a film, on television, in magazines or in books, what they saw was angular machine cabinets with a lot of flashing lights and reels which were turning like mad. In the foreground stood the high priests of the machines: serious looking men in white laboratory coats, generally busily changing some magnetic tape.

In science fiction, the generally accepted idea of the electric Leviathan was pushed to the extreme: Arthur C. Clarke's book "2001 – A Space Odyssey" which was made into the psychedelic cult film by Stanley Kubrick, turned the computer into a Mephisto: HAL (IBM, always one letter ahead in the alphabet), the villain with the friendly voice. This diabolic image stood its ground for quite a long time: from 1941 – the year the German Konrad Zuse built the first of these civil servant computers – until 1982.

The technical revolution began in obscurity, in 1969: computer engineers at the Texan company Datapoint, appointed the companies Intel and Texas Instruments to develop a single cell computer: to accommodate the most significant functions of a computer as compactly as possible on a chip the size of a finger nail.

Intel succeeded in accomplishing this technical feat but the new chip was too slow for the contractor Datapoint. The management at Intel did not want to throw their development away and, consequently, a year later put their module with the trade name "4004" on the market. This marked the birth of the micro-processor. The basic vital functions of these single cell computers were the same as the dinosaur ones, the large-scale computers.

This immediately set the inventive-minded throughout America playing with the new component. A number of designs on the chip basis landed in the evolutionary dustbin: Frankenstein computers, neither commercially nor technically successful, computers kits which were only purchased by passionate hobbyists. Nevertheless, some of the hobbyists were successful and were able to leave the evolutionary music of the hacker scene behind them. In California, Stephan Wozniak developed the first "Apple" – small enough for the writing desk and kitchen table, with a keyboard and the familiar goggle-box as a screen. In 1977 "Steve Woz" and his friend Steven Jobs proudly presented "Apple II".

The team Wozniak& Jobs became a myth; these two young Americans became the heroes of the success story, that Apple's PR company had been reciting to journalists and computer fan circles: a quick sprint in training shoes from the garage to the billion dollar company.

Challenge the masters of the market "IBM" with cleverness, groove and enthusiasm.

And so the David and Goliath myth of the computer branch came to be, which from that very day, was to accompany the recognition of the "personal computer" and the new aggressive manufacturers. Of course, the purchasers of the bulky civil servant computers were, in the meantime, able to operate their computers in the large laboratories and administrative offices with keyboards and monitor displays, but only during specific office hours – computer time was expensive time.

A few years later, IBM, the company with the bureaucratic image also finally embarked upon a new course. In 1981, five years after Commodore had also entered the market, what was then the sixth largest company in the world, also put a small, dainty PC on the market.

By 1982, the personal computer had already completely taken up all the ideas of a computer: The US news magazine "Time" pronounced the computer "Man of the Year", and on the front page there was a plaster cast figure in front of a PC, hands distrustfully on its lap "HELLO / MAY I HELP YOU?". Who could say no?

Ten years later: The Japanese presented the first studies of the body-computer which could be worn on the wrist. Computers that looked like slates and could only be operated by slate pencils, were to replace the indispensable Filofax; Telephone and computer produce communication aids which latch on, at any time, to the global network by means of mobile radio net. Digression: the hacker in his natural habitat, quote from the small Pocket Dictionary of Media Cliches, 29th Edition, Hamburg 1992.

Hacker, the (häkoer) generally a non-sporty young man with a pale complexion, stubble, greasy hair and big circles under his eyes. The H. likes to sit in small stuffy rooms. The focal point of his life is the computer, preferably connected by the telephone network to the army's or some international concern's large-scale computers. The goal of the H. is to crack the security systems of external computers in order to pass on military or commercial secrets to the H. public.

During such activities, the H. rarely sleeps and rarely speaks. He like to stick walkman plugs in his ears. The H. nourishes himself on pizza, hamburgers, Coca-Cola, beer, potato crisps and appetite reducing and performance boosting psycho-pharmaceutical products. In his very brief leisure time he prefers to preoccupy himself with questions such as "Is there still a God in this age of simulation? Is it a sin to design robots and artificial intelligence? Will I ever find a wife?" Social life is sometimes spent by the H. in special H. clubs.

As regards the etymology: "to hack", English verb (middle English: hacken): to destroy something with irregular coarse strokes, break open; "hack", English noun (derived from middle English "hackney"): run-down carriage horse which can be hired, metaphorically speaking: politician who can be bribed, penny-a-liners, taxi drivers, prison warders; from this derives the English slang verb "to hack": to tout oneself as cheap all-round labour, to mess something up; "hacked", English adjective to designate a certain style of working; banal, worn-out, cheap, routine, commercial (source: the American Heritage Dictionary of the English Language, College Edition).

In every day language the H. is regarded as being the spontaneous programmer who tries to solve code problems by playing around – until he gets it, or the machine gets him ... !

The first appearance of the H: 1961 the Model Railway Club at the East Coast University of Massachusetts' Institute of Technology (MIT) became a computer club. There, the first H.'s appeared for nocturnal programming sessions with the object of writing the perfect program, and creating a megabrain.

In the heyday of the computer myth, between 1979 and 1989, the H. was regarded as being an alchemist, inaugurated into the secret science of the computer.

Spreading of H.ism: The West-coast H.'s enriched the myth of the sweet aroma of the flower power movement. Pot-smoking bay-lefties got together in the People's Computer Company and set the tone of the modern Robin Hood battle-cry "computer power to the people": Free the flow of information from the clutches of the bureaucratic moloch IBM! However, as the stars of the H. scene went into business and only let "Bolivian marching powder" get up their noses and no longer wanted to talk to every quest at the party, H.ism moved over to good-value, free tele-communication networks. The battle-cry: Everyone has a right to know international party gossip! A popular branch of West-coast H.ism in Europe is the

"Whole Earth Review" published by Steward Brand.

In Europe, there are various national folkloristic versions of the H.; in Germany the image of the H. is typified by the image promoted by the Chaos Computer Club during the cynical 1980's. At present, the H. is a shadowy figure, that flits through the glosses of the German "Zeitgeist" magazine.

THE AESTHETICS OF PROGRAMMING

It is not only the hobby programmer who is confronted with the conflict between brilliant drafts and bitter experimentation. Even professional programmers have to cope with this. The difference between the two is the fact that professionals are involved in a commercial, labour-divisional context. Secondly, professional programmers are generally involved with larger programs. And, with a certain programming scope, the method of simply trying things out becomes more risky. It becomes more and more important to know what effects are being generated inside the computer with every step taken.

The work involved in designing programs and writing programs has something of the beauty of mathematics about it (everything complies). On the other hand, it is subject to a duplicable functional test (that thing must work, if not that was a bad piece of work done, a botch up), and finally, it is always subject to the pressure of merciless commercial success criteria. The deadlines wrecklessly proposed to the management or the contractor by a programming team must be adhered to. If that is not possible, the department is sided. In the worst case, the entire firm can face the bankruptcy courts.

Programmers prefer to talk about the fun side of their work, about the adventure of abstraction, triumph, of getting huge programs. Informatics people, in all modesty, tend to point out the ingenious nature of their work. In actual fact, with all inventor mythology, it is simply a matter of developing the better machine. This is difficult enough, just as hard work as cleansing the Augean stables. It is a matter of formulating, as simple and restricted and as adequate a model of reality, as possible.

C. Wayne Ratliff, for example, reports that he found the idea for dBase, (the most wide-spread data bank system which was put on the market in 1978) as he was developing a mathematical process for his football tipping group. "I was trying to determine the winning team by going through newspaper for newspaper. A dreadful process. At some point, I decided that this wasn't possible without a computer. A week later I had completely forgotten about football. I had decided that the world needed a databank manager with a natural language".

Ratliff always orientated himself very closely to the question of what his target groups needed. With this goal in mind, Ratliff qualifies that dBase is "not perfect": "If I had let my heart rule my head" he explains, "and not followed the advise of others (do it bigger and do it faster, do it smaller, go for 16 bit, make it multi-user capable, use more languages), dBase would have been almost perfect. I simply tried to move in too many directions in order to satisfy everyone, at least just a little. This was ultimately my only mistake".

Perfectionism is an occupational illness that all programmers suffer from. The larger the program, the more important it is for the programmer to stick to his ideals: pure reduction, precise order, simplification, structuring, organization. The attraction of just sitting down and writing page for page of code is, as many informatics people say, immense. But even the largest of programs must remain easy to survey. If not, what could result would be a pieced-together chaotic heap of data: "Hack". The ideal of every programmer is brilliance, elegance and aesthetics.

"The better the technician, the more aesthetic and artistic the machine and the greater the chances that it will work" says Bob Frankston, who wrote the successful program VisiCalc" (table calculations) together with Dan Bricklin. "Good technology is similar to good art" says Frankston. The first draft of VisiCalc was written by the Harvard student Dan Bricklin during his student days, at the end of the 1979's. In 1979, he founded the company "Software Arts" together with Bob Frankston, to finish writing the program. Dan worked by day and Bob by night. "At that time we had no idea at all, as to how electronic table calculations would be received" says Bricklin.

The program was a huge success. Despite this, neither Dan Bricklin nor Bon Frankston became particularly rich or famous. The company, Software Arts, was disintegrated in a legal dispute with its obstetrician, who had lent the first equipment and assumed the role of publisher.

What programming is, formulates Bricklin rather relaxed: "A part of it is manual work and a part is science. As with many things, practice makes perfect. It is not a rigid procedure, but more a manual art. Programmers with a sound education generally have an advantage over those with none. Some people have a hand for it, others

don't. But it is always good to thoroughly involve yourself in a sector. If we are making products for commercial application and the thing must be finished, one shouldn't take it all too seriously". Bill Gates, who has the steepest career behind him of all informatics people, formulates the demands made on "top programmers", the racehorses of this branch, in a much more stringent way. In the mid 70's, as a Harvard student, he wrote a Basic-interpretor for the first commercial microcomputer, the "MITS-Altair", together with his former high school buddy, Paul Allen. Once the compiler was finished, the friends founded the company, "Microsoft". Gates, as did many good programmers who spent the nights of their youth in front of a computer, went into management. Today, he is 36 years old and one of the richest men in the USA.

"The days when each program was a masterpiece are over" said Gates in an interview made in the middle of the 1980's. But nevertheless, he abides by the working principles of those early days: "the best software comes to be when one individual can exactly imagine how the program works. To do this, you must love this program and concentrate on keeping it simple, unbelievably simple."

Gates also emphasized that he thinks nothing of the "prima donna cult" "where someone, just because he is good, does not want to comment on his code, or does not want to speak to other people, or wants to force his ideas on everyone else". Gates continues. "We need people who mutually respect each other. I believe that most big programmers like being together with other big programmers. If they have thought out an unbelievable algorithm, they like being with their contemporaries who understand and acknowledge what they have thought out. Because, if you turn up with such an idea and have the model firmly implanted in your head, that is a lonely business. If you have assumed that a process would be complicated and then you find a way of simplifying it, it's a fantastic feeling. But one needs the feedback of others". A considerable amount of programming work is spent in fighting bugs" (derived from the American everyday word "to bug": disturb, annoy, a verb that originates from "bug"; beetle, bacillus). "Debugging", the correction of errors, is a nerve-racking work in the development of computers. The hardware engineers ("Hardy Boys", after a popular detective novel series for boys) have also to deal with real beetles, spiders, dust and the likes from the realms of small organisms, when debugging. But, normally, the machine goes on strike due to design errors. Software programmers have to deal with errors in reasoning and unclear solutions during debugging, a feature which can never be avoided – even if the programmer were as "perfect" as a computer.

EVERYDAY LIFE ON THE COMPUTER

Software is the result of an arduous, often enjoyable but generally hectic process which calls for expert fantasy and tends towards bureaucratic organization, at one and the same time. The quality of more extensive software depends very greatly on the cooperation of the programmers, on the exchange of information between all those involved, on the qualification of these people, and on the possibilities of getting into new areas. As Biame Stroustrup, who developed the programming tool C++, says that the quality of the product is ultimately determined by the "culture" in the software workshops. And this, particularly in smaller companies, is often bad. In Silicon Valley, the cauldron of the computer industry, highly qualified programmers are constantly hopping from one better paid job to the other, and have very little interest in getting involved with the techniques and technology of one individual company. This in turn causes managers to make quicker use of the capabilities of their members of staff.

The myth that one could hack through and make quick money here, meanwhile shows contra-productive effects. The days when the nimble nightshift could be pacified with rapidly climbing stock shares, are over. As a rule, programmers get medium-scale salaries, just as all other skilled workers do.

Consequently, software is no longer being produced under ideal conditions. To be able to sell well, this shouldn't be noticeable. Customers want software which is, quite simply, a handy tool: uncomplicated, easy to survey, multicultural like rock music and as friendly as an airline stewardess who finds nothing human strange.

However, most programs required by offices are still complicated and involve the user in a host of functions which cannot always be quite as rationally ordered as they should. And, as very few office employees do in fact have the time to play Alice in Wonderland, most EDP users do, in fact, get into the habit of restricting themselves to the most important functions – even when the software is furnished with the most wonderful gadgets. Office and household appliances with computer controls still have far too many buttons. They look as though their insides have been turned inside out.

Informatics people who work on new systems have few scruples in admitting to the deficiencies of the current computer generation. Many professional programmers prophesied years ago that computers would become much simpler to operate, to the point that they would become just as spectacular as an ordinary telephone.

The object behind programming work was, from the very onset, since Blaise Pascal's first draft of a computing machine some 350 years ago: to simplify complex working processes. The transfer of intellectual activities into the functions of a machine does not, of course, hold true for the PC. On the contrary: after a staggering spell of fast development work over the past 20 years, the computer branch is facing the next big challenge at the moment. The demand for PCs has declined greatly, prices have reached absolute "rock bottom" levels. Manufacturers have discovered that integrated systems which connect up all the departments in a large company are the next technical challenge. And, as this technology has not yet matured, existing computers should be interlinked, in the first instance. The German managing director of IBM, Bernhard Dorn spoke to "CeBit" magazine about the cobbler-work performed on existing systems: "Actually we would have to throw everything away, as it is trash. It would take five years until a new DP landscape could be created. But, nobody could afford this stand still".

THE SHORT HISTORY OF THE PC

It is the software that turns the computer into a useful tool. Software should help us take care of tasks, or, in the form of a computer game, help us kill the time we have saved. But only when there is no longer anything specialist and unfamiliar between us and our intentions, can we really use software. The "user interface" of the computer (input and output media) is only ideal when it assists man's activities on the computer, free from error.

The current form of the user interface consists of input equipment such as keyboard, mouse, joystick, plotter, light pencil, language recognition equipment and output equipment: sound output and optical equipment such as printer and screen. Elements of the optical processing of information are: Bitmap presentation of words and graphics, more or less structured menu management and windows (windows to simultaneously show the data of several different programs on the screen). The graphic elements range from lines and boxes to video pictures. Many office programs work with icons (pictograms), which symbolize the computer functions as office utensils (a kind of crib between the writing desk and the computer; terminus technica: desktop metaphor).

The outlines of these individually operable multifunctional computers were developed in the 1960's. In actual fact, the entire concept was prepared in 1945 by Vannevar Bush, a scientific advisor to President Roosevelt. But Bush was not able to build the thing he called "Memex", a keyboard controlled microfilm unit to automatically classify information. He could only describe it. The first step towards interactive graphic user surfaces was made by Ivan Sutherland at the beginning of the 1960's, with the sketchpad, a graphic program which could process forms drawn on the monitor with a light pen. Almost at the same time, Douglas Engelbart – a former radar technician in the US Navy – developed the "NLS" system, an interactive computer with mouse, picture tube screen and graphic representation of information in the form of search trees, at Stanford Research Institute.

Engelbart developed this first "personal computer" under the assignment of the Advanced Research Projects Agency (ARPA) which was founded in 1962 -and after the "Sputnik Shock", was the "think tank" which was lavishly financed by the Pentagon in the USA. Under the leadership of J.C.R. Licklider, the development of directly controllable computers was urged at ARPA, involving the use of immense resources – for informatics people, this was an experimental paradise. The "ARPA-dream" lasted until the beginning of the 1970's, until the beginning of the Nixon era, which radically withdrew the Pentagon's experimental funds and the distribution of resources again went directly to the development of military technology.

In place of ARPA, there was now the copier company "Xerox". Stimulated by the rapid growth of the company "Digital Equipment", Xerox decided to enter into the mini computer business and, in 1970, founded the research eldorado "Xerox Palo Alto Research Center" (PARC). In 1972, the prototype of all subsequent PC generations was presented: the "Alto", a mini computer with a mobile memory part (a forerunner of the floppy disk), bitmap-monitor indication and mouse. Nine years later in 1981, Xerox Star presented, (developed with PARC), an all-round office software system which was equipped with all the elements known to us today. "Star" was the first software package whose design was directed entirely towards office automation. The target public were persons who administrated information and only used the computer from time to time, and as it was termed in Xerox retrospective "are interested in getting their work done, but are not at all interested in the computer".

The "Star" system was a financial flop, above all because at the beginning of the 1980's there was no market for expensive all-purpose machines (\$15,000). Xerox bet on the wrong horse. At the beginning of 1975, the first small computer was on the market which was based on the Intel-chip; the "Altair 8800" (company MITS), a module that one still really couldn't use. But the Altair, which, as the legend goes,

got its name from the Star-Trek saga, became the bright star in the PC sky. Two years later, Apple, Commodore and Radio Shack had small personal computers on the market which still couldn't do very much, but which sold fantastically well. A lot of the design principles from "Star" went into Apple's "Lisa" (1983, a flop) and Macintosh (1984, Apple's success model which was aggressively forced on the market) and were also taken up by Microsoft.

There had now been a long paternity suit raging between Xerox, Apple and Microsoft for the copyrights of the design principles developed then. In actual fact, the know-how transfer arose from the fact that many developers changed company. Alan Kay, for example, who designed the sacred PARC "DynaBook" concept and "Smalltalk", passed through all the alleys in the scene like a jack-of-all-trades: ARPA, Xerox, Atari, Apple, MIT Media Lab. Apples' LisaWrite and Microsoft Word originate from the former Xerox Bravo members of staff (Tom Malloy and Charles Simonyi).

From PARC came the successors of Sutherland's Sketchpad, graphics programs like "Draw" (by Patrick Beaudelaire and Bob Sproull) and "Doodle" (by Dan Silva), and finally also the laser printing technique -the key to desktop publishing: some programmers from the PARC group "Interpress" founded the company "Adobe Systems" and developed "Postscript", that has in the meantime become a standard. At the same time, in the company "Atex", text processing systems were being developed for newspapers and magazines. And finally, in 1985, when the first extensive Desktop Publishing system, "Page Maker", was presented by Paul Branernd, ex "Atex" member of staff and "Aldus" founder, the office PC tool box was complete.

GAMES, FUN AND MADNESS

Amongst all programmers, games writers are the ones most orientated towards the customers' wishes. Computer games must be very attractive and very simple to succeed against all the other play things which collect in a child's room. American children grow up with an close relationship to the screen (it's magic). In Germany, studies have revealed (DJI Munich) that most children lose their initial burning desire to play with the desk computer, after a short time. And, but a mere few children actually fulfill their parents' hopes and use the box to write basic programs. Video games are more interesting, especially the ones where you can't see that they are computer- controlled.

Software designers who busy themselves with the creation of the "interface" (the user surface on the computer), are increasingly taking up the ideas of games writers. Their question is: How do we create the dialogue level of business computers to make working with them fun? Working with the computer should be fun, at least that is what Xerox always maintained: "I look forward to the office".

Video games are aimed at a public which has nothing remotely to do with computers. Most games designers, however, do come from the computer scene and make the theme of their games desire and grief at work, again and again and again: the old game of "law and chance", of getting different elements in order and working away at them under the pressure of time; recognizing features of objects, ordering them, creating stable balances (in the simulation game "SimCity", for example); fighting against the all-knowing machine. Gaining territory, avoiding viruses, storing in good time so that nothing gets lost in event of a crash (a ritual in almost all Sierra Adventures); the foolishness of overtaxed combination artists (e.g. in Lucas Adventures); orientation in the unknown closed space (e.g. Collosal Cave, the legendary very first adventure game for the computer which is constantly being re-written: ... it is rumored that some who enter were never seen again...").

Many games programmers are particularly fascinated by the idea of motivating the computer to make statements which are completely alien to the machine. Even in many music programs, the confrontation with the machine still plays a more important role than the music (it's so hard!). The devoted goal of the virtuous software designer is to replace the mechanical, the complete integration of the tool into the creative process. However, there are a number of programmers who are decidedly interested in the autopoiesis of a machine cracking up and making itself independent.

A particularly well loved joke among programmers pre-occupied with artificial intelligence, are the psychopath programs. At MIT, several programmers amused themselves with this to simulate scenes from psychiatrists' surgeries. Marvin Minsky and his friends gloated when one of their AI programmers even managed to dupe a few psychologists with a program which imitated a paranoid person. "Once you type something in", Minsky explains, "then it replies in a fairly alien, irrelevant way and says things like: I don't trust you. After a while it says: I'm not talking to you any more. Then it simply says nothing".

The program was, to put it precisely, very primitive, but the psychiatrists - Minsky reports - were impressed by the performance of the Artificial Intelligence research team. And, all that without complicated neuronal networks.

The most well known "doctor-program" originated in 1966: Joseph Weizenbaum's language analysis program "Eliza", which was named after the pupil in Shaw's Pygmalion because it also held "educated" conversations with no meaning, by taking up and varying catchphrases from a human dialogue partner. When Weizenbaum discovered that the Eliza program was taken seriously, he wrote his well-known deduction entitled "The Power of the Computer and the Lack of Power of Common Sense". The -in the meantime - retired MIT professor now says "Eliza is Public Domain. Normally I don't want any more to do with it when I am approached about it, but it is a part of history".

A relatively new game with the aesthetics of a computer making itself independent, is "Screensaver". The screen, as well-informed software sales people say, breaks down if the picture tube light spots are not re-distributed from time to time: The picture burns in. Actually, it would be sufficient to turn down the brightness of the screen during longer breaks. But why should we have it that easy, when we have eight megabyte memories at our disposal, of which a minor percentage would be sufficient for the mere work? Screensavers are senseless and beautiful.

The "After Dark" Screensaver (Berkely Systems, 1989 / 1990) for the Macintosh, motivates the computer to some surprising pictures as soon as the computer user, as is correctly and beaurocratically termed, neglects the use of the computer and only sits, thinking, in front of the box. In the case of "Fish", the screen nicely transforms into an aquarium, fish, jelly fish, sea horses float weightlessly across the monitor.

"Puzzle" first of all knocks the monitor screen down into cubes and then pushes them back and forwards until the graphic surface becomes completely senseless.

Such automatic gadgets that dissolve the screen order, get their attraction from the superficial similarity to virus programs which, however, most unfortunately, sometimes upset data beneath the graphic surface. Viruses are programs which can be copied into existing host programs and become active when this program is called up, and sometimes only harmlessly lull some funny song ("Yankee-Doodle Virus" e.g.), at the very worst, they can destroy the entire contents of the fixed disk. Viruses are a horror for the small owner who has already paid more than enough himself, for the software, and who does not want his data to be ruined by some hacker held back in his career. But, the software manufacturers do have the guaranteed active anti-virus recipe on hand. Avoid bootlegs! Be careful with games and copies from university computers! Only load licensed original software onto the personal computer! In order to emphasize their recommendation, most manufacturers include some games (pedagogically worthwhile) in the multi-functional office software package, with large-scale purchases.

The entire devotion of the virus hackers to their mission of "upsetting the system" can be seen by the graphics which go into action as soon as the fixed disk is infected: A poetic aesthetics of disappearance is staged - the on-the-quiet pleasure obtained by the troublemaker in destroying order. Romantics also speak of the "Biotop" principle, because as soon as the computer viruses have been put in circulation by their creator, they multiply like their relatives in biology, sexless into eternity, or at least until they are discovered by one of the many sophisticated anti-virus programs.

In the case of almost every virus program, the data visible on the screen remains intact, for the meantime; it will only be modified a little before it finally disappears: "Zeroeat" eats up all the zeros. "Drop" makes characters slowly and not very spectacularly, yet quite nonchalantly, drop out of their context into the lines below. "Joker" makes everything gay and with a data change, imparts some villainous message "Have you ever danced with the devil under the weak light of the moon? Pray for your disk!" "Whirl" makes the characters visible on the monitor whirl across it - the monitor now having grown dark - like a whirlwind, and then it breaks down into blackness, into the hole of bits and Bytes.

SOFTWARE THAT MAKES SOFTWARE DISAPPEAR

Software, as it is presented today, has developed in the electric field between work, machine and game, computer games are becoming more and more sophisticated, more like cinema films, while the interactive surface of office software is being more and more reduced and simplified. In both sectors there is the tendency to integrate complex control processes into the inside of the machine. To have technology disappear from sight. Complicated "Fuzzy-logic Modules", for example, only serve to have the Tokyo underground railway user forget the technical details of a train ride, as far as that is possible.

Even in science, the former methods of reconstructing a section of reality in the form of a model, are being replaced by new methods which, among other things, are seeking the assistance of new computer systems. The working media is new. The change in method is not new. It only relies on new auxiliary media.

In the exhibition "Creative Software – Man and Milestones", the following are among the programs which can be tried out, played with and experimented with: VisiCalc, Bravo, Lisa Office System, dPaint Amiga, Basic, Logo, PacMan, Star, Collosal Cave, Eliza, After Dark, (de-activated) virus programs, SimCity, Life and the multimedia game "Piazza Virtuale" which will be staged from the Kassel "documenta".
Eva Weber, Klaus Madzia

Quellen: West of Eden. The End of Innocence at Apple Computer, Frank Rose, Penguin 1989; Hackers. Heroes of the Computer Revolution, Steven Levy, Dell 1984; TheValue of Micropower, Adam Osborne, 1974; Readings in Human Computer Interaction, Hrsg. Ronald M. Baecker, William A.S. Buxton, Moilgan Kaufmann Publishers 1987; Logo. Programming with Turtle Graphics, IBM 1983; European Software Festival, Eva Weber, Vogel Verlag 1991; Die Zitate von Dan Bricklin, Bob Frankston, Bill Gates sind dem Buch Programmers at Work, Interviews by Susan Lammers entnommen, Microsoft Press, 1986, 1987 in deutscher Sprache beim Markt & Technik Verlag unter dem Titel Faszination Programmieren.

© AEC Ars Electronica Center Linz Museumsgesellschaft mbH, [Impressum](#), info@aec.at