NOVEMBER 1983

VIDEO SYSTEMS

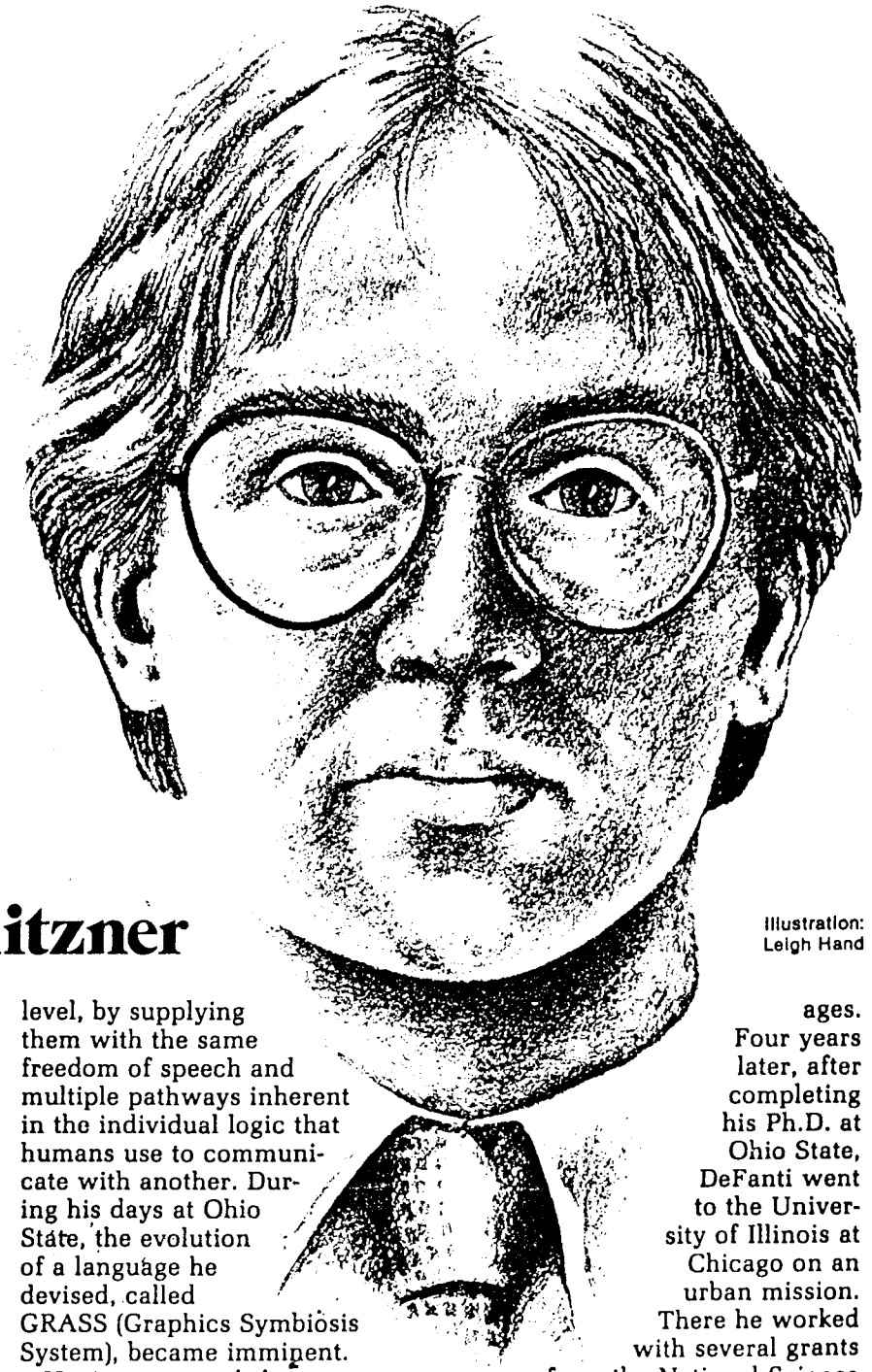
The magazine for video professionals

$10.00

7th Annual VideoIndex '83

Hardware Buyers' Guide Issue

Inventor: Tom DeFan



By Melanie Mitzner

Illustration: Leigh Hand

In his youth, Tom DeFanti did one of three things—worked on projects for a science fair, watched inordinate amounts of television or traveled as far as his bicycle would take him. Nevertheless, he was studious and took his math and science studies seriously.

After receiving his undergraduate degree in math and education from Queens College, DeFanti set out for Ohio State University to earn his master's degree and Ph.D. in computer and information science. There, he developed computer systems for educational use, and, as a research associate in the art department, concentrated on the development of software for artists who were all too frequently falling by the wayside because of the machines' inaccessibility.

DeFanti was determined to find a way in which people could communicate effectively on a visual level, by supplying them with the same freedom of speech and multiple pathways inherent in the individual logic that humans use to communicate with another. During his days at Ohio State, the evolution of a language he devised, called GRASS (Graphics Symbiosis System), became immipent.

Having assumed that user control of real-time animation and computer graphics systems would result in the kind of interactivity that was feedback intensive, DeFanti set about loosening up the structure of computer languages. He gave up the programmer's typically purist approach and merged the real-time control structure of television with a flexible, hybrid, picture-based computer language that could generate, process and manipulate images.

Four years later, after completing his Ph.D. at Ohio State, DeFanti went to the University of Illinois at Chicago on an urban mission. There he worked with several grants from the National Science Foundation, the National Endowment for the Arts, Carnegie and Sloan to develop alternative teaching methods, similar to CAI (computer-assisted instruction). These methods would then be applied to raise the level of education in Chicago area high schools, which was notoriously low at the time. While fulfilling that mission, he continued research and development on GRASS.

Seven years after he first began

Creating graphics with a special computer language

Tom DeFanti

developing the computer language, he managed to draw five conclusions that would lead him directly to the source of his labor. He concluded that a purist approach to electronic language was hopeless. He realized that it was necessary to have the ability to merge independently developed sequences without having to write new software to do it. Prioritizing and still maintaining flexibility was essential.

User extensibility was also important so that new concepts could be realized without additional input from the original software designer. And finally, multiple pathways to reach the same visual end was crucial because no two graphic artists, or educators, communicate the same. Although these objectives were ambitious ones, DeFanti was determined to achieve them.

Working on $70,000 Vector General display oscilloscopes and PDP 11/45s, DeFanti further developed the language of GRASS—a language that could either teach a person how to use it to create applications programs or place some dynamic graphics capabilities in the hands of computer novices, such as artists.

With GRASS, DeFanti proceeded to develop teaching materials that he still uses in his courses. GRASS was most frequently used by artists at the university to generate graphics for museum exhibitions worldwide and, at the same time, earn tenure because of the international recognition they received. In 1976, Larry Cuba produced the briefing-room scene in *Star Wars*, using DeFanti's GRASS system, one of the first major Hollywood uses of computer graphics.

Until 1977, GRASS remained in the hands of educators. Then Larry Leske, one of DeFanti's graduate students, went to work for Dave Nutting Associates, a wholly-owned research company for Bally Corporation. At the time, DeFanti was still hacking away, trying to build some hardware to accommodate his graphics software when Leske told him that Nutting had built a box with a keyboard for a home arcade game. They realized that this box could be modified to satisfy DeFanti's requirements, so a relationship began between DeFanti and this research arm of Bally Corporation. DeFanti soon reciprocated when Bally asked him to develop a graphics language for a home computer that was easy to manipulate and versatile enough to sustain attention.

"They weren't going to offer any courses or give out any phone numbers that consumers could call to debug their program," DeFanti recalled. "They wanted something very simple to operate. I asked them if they cared how

many FOR-NEXT loops (in BASIC, that's a measure for how fast the system operates) it could do in a second and they said that wasn't important. So Nola Donato, a grad student of mine, the infamous games programmer Jay Fenton and I began to develop what would later be called Zgrass."

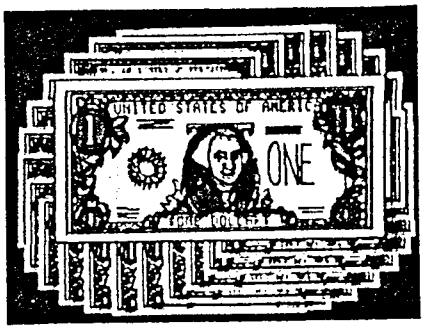
At that time, university policy dictated that any research done on the premises for non-print media would officially become the property of the university, which meant that the threesome had to work feverishly on their own time to develop the language. Much of their R&D took place in a cabin situated in the Wisconsin woods.

"We had a program up in a year," DeFanti said, "which was a real struggle because between the three of us, we had to share one really primitive development system, a Z-80 with only 16K memory. There was much acceptance of the system we developed because it was really the first flexible home computer that concentrated on graphics. However, just before production start-up, Bally decided that for them there was no future in home computers because they were not a consumer product company, but specialized in arcade games and entertainment enterprises. So the project was canned. Fortunately, we had access to the silicon chips they developed for the system and they agreed to let us form companies using the technology as long as they weren't directly competitive with Bally."

It was conceivable that Zgrass could lead to a subculture on the frontier of computer graphics, because DeFanti had worked to modify what was formerly the extremely complicated GRASS language into an elegantly simple computer graphics medium. What made Zgrass different from its counterparts was its fundamental graphics components. Other CAI-developed software had been fundamentally oriented toward words.

Zgrass has all of the string processing and arithmetic capabilities of other languages but the graphical constructs are also deeply embedded into the program. One of its greatest assets is that it was designed as a tool package with which to construct innumerable programs to fill specific needs.

BASIC, a language often used by programmers, is extremely inflexible in comparison. You simply cannot pull elements off of a disk, execute them and get rid of them. But Zgrass has a flexible disk and memory utilization structure that acts much like an operating system. It can run for long periods of time and will keep track of all of the changes as it

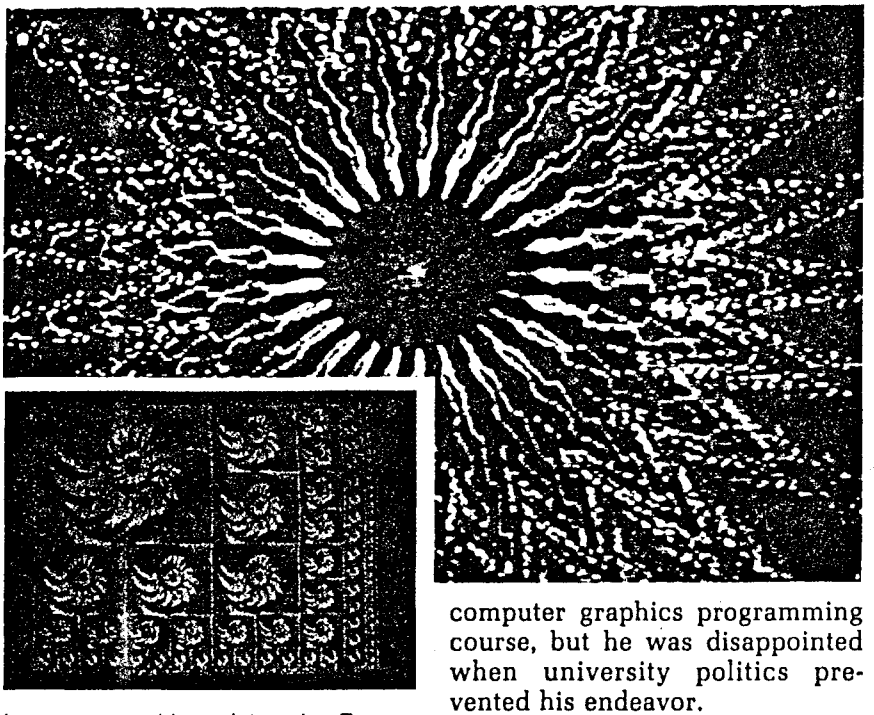


Electronic graphic images from Zgrass developed by DeFanti, which were made to accommodate broadcast TV. *(Designs by Jane Veeder)*

continues to bring new elements in and out of the picture. Also, DeFanti developed time-based constructs that allow you to move elements around independently, so that you could write a program to do one animation and a program to do another, then have them operate simultaneously without rewriting the programs.

DeFanti describes Zgrass as an "interpretive language with a compiled option that, after the system is debugged and operating properly, one can compile to speed it up." Generally, languages are either interpretive and slow or compiled and fast, lacking highly interactive features and developmental capabilities that are required to generate graphics. Zgrass gives you the best of both.

Under a National Science Foundation grant, DeFanti hoped to use Zgrass to develop the first



Images created by artists using Zgrass. DeFanti helped to create a graduate program that is informally linked between the School of Art & Design and the College of Engineering at the University of Illinois in Chicago. DeFanti hopes that this emergence will improve the quality of computer graphics as it is known today. *(Designs by Jane Veeder, Dan Sandin and Tom DeFanti)*

computer graphics programming course, but he was disappointed when university politics prevented his endeavor.

In spite of the revolutionary achievements DeFanti had realized in the development of Zgrass, marketing proved to be his greatest obstacle. After Bally's interest waned, he had approached other hardware manufacturers.

"Hardware people are notorious for demanding software that is compatible with their systems," DeFanti said. "Even though I tried to market Zgrass, along with the inexpensive $20 chip set from Bally, they refused to put it into their circuits. They want you to take your software and make it like the rest of the stuff over there that conforms to their hardware. We resisted, which, from a financial point of view, was not the best strategy."

DeFanti also observed a psychological block, saying, "The second industrial revolution is based on taking raw goods to manufactured goods. They build factories to do that. You don't need a factory to produce software. For a typical manufactured product that

grosses $100 million, you might spend $1 million getting it into the market and about 1 percent on R&D. With software, it's more like 90 percent R&D and zip for manufacturing."

Determined to keep all their efforts from getting lost in the shuffle, DeFanti hired some students to take the systems from the Bally Arcade resolution of 160/102 to the commercial video game resolution of 320/204. With help from Dave Nutting Associates, boards were wire-wrapped and units built for use in DeFanti's classes. During this wire-wrapping session, a small company evolved, taking the widget that DeFanti had wire-wrapped and developing it into a product called Datamax UV-1. The object behind that product was to upgrade the graphics and animation capabilities of the cable TV market.

"We thought the world would be happy to see animation, readable type and pretty graphics on cable TV," DeFanti said. "But we soon discovered that nobody cared because cable TV doesn't exist in a competitive market. Once they get the franchise, they roll over and drop dead."

Perhaps dismayed, but not completely discouraged, DeFanti's determination led him to the company Astrovision. They were interested in buying the rights to the Bally system. Further modifications ensued until, much to DeFanti's chagrin, that path to Astrovision turned out to be a dead end when the company declared bankruptcy in the summer of 1982.

However, DeFanti persisted. He refused to succumb to ill-fated, business relationships. Because it was too difficult to market Zgrass with the chip set to low-end hardware manufacturers, he decided to upgrade the language for more powerful microcomputers. In fact, flexibility was such a built-in characteristic that he was prepared to adapt the language for just about any 68000 or PDP 11-type processor by writing in the "C" langauge.

DeFanti is now in the process of developing a language that will "hopefully become the CPM for graphics because, in reality, it functions as an operating system, but it just happens to have graphic elements built into it as its fundamental bottom."

This new version of Zgrass, presently referred to as RT/1, will operate on 16-bit computers. DeFanti is also pleased that he has been retained to develop RT/1 for the specific purpose of creating artists' work stations for video game design. Terry Disz, Dan Sadowski, Stephen Joyce, John Lindberg and Peter Caldes are programming RT/1. Jane Veeder and Copper Giloth, artists, are coding the artists' work stations.

He explained, "We really attacked the time-based elements. We're also using a lot of interactive controls—a digitizing tablet, dials that can be assigned to any variable in the program so that, at any point, you can attach an element to a dial, manipulate it, find the right place for it, then write the program."

What this implies is a dramatic shortening of the feedback loop. On many other systems, one must conceive of the image, input a number, compile the data, look at

it, edit it, compile it again, inspect it once more, then write it in the program—a process that can take up to 10 minutes for a single trivial change.

DeFanti compares his method to the manner in which a piano is tuned, using a tuning fork and adjusting the pitch to the sound waves. "What we're really involved in is learning how to tune graphics, with dials...tune pictures," DeFanti said. "And nobody has ever been able to do that before. We're trying to do to visuals what people have been doing to music for 10,000 years."

In addition to the application of RT/1 for video game design, DeFanti also hopes to adapt it for the home computer market. "There are people," he said, "who want to have a much better language than BASIC, which is virtually the only thing available. PASCAL is a language for computer scientists but not a very good one for interactive graphics development, so I think in terms of the home market as being a publishing medium for RT/1."

Meanwhile, he has an excellent environment in which to debug the system. Not only do his facilities at Real-Time Design provide him with basic research needs, but because he helped to create a graduate program that is informally linked between the School of Art & Design and the College of Engineering where he is tenured, he will witness the application of RT/1 by the most diverse users conceivable.

"The program is rather unique in that we teach artists how to survive in a digital/analog world and we teach engineers aesthetics, then we throw them all together," DeFanti said. "It's an unfunded program and we only accept five new students a year. I did this in association with Dan Sanden and Guenther Tetz of the art department because we feel that artists ought to be in charge of computer graphics. In many computer graphics businesses, the artists more or less came up from the graphic design department where they did press type slides. They are considered slaves; they are not really treated with respect for their craft. This attitude has to change and there have to be more sophisticated systems developed to improve the quality of computer graphics as we know it today.

"Artists should be the ones to get out there and specify, configure and direct the construction of these systems. To achieve this end, we've been teaching them how to get out there and deal with technical parameters like wirewrapping and programming."

When DeFanti was asked to advise software designers about the rigors of product development, he said, "Not having been so successful, I don't know if I'm really an authority on that." However, after assuring him that failure was always a prerequisite for invention, he said, "Don't expect people to accept your inventions because if you maintain that as your goal, you'll really be disappointed. As for financing, there are several ways you can go. Federal grants are difficult to get, but if you have a good track record, good credentials and you develop a relationship with the people on the panel, it becomes a possibility.

"Remember, they have to show their management that you'll be doing something that will have national impact. These days, however, little money for graphics is coming from the federal government except for research relating to the military. In fact, the government has to support education better than they are now. I'm teaching the *history* of computer science, not computer science. You know why? Because we're teaching on computers that are 10 years old. We simply don't have new equipment unless a manufacturer gives it to us. Private industry is your best bet.

"Acting as a contract research arm for a large corporation definitely has its advantages. They pay researchers what they deserve instead of trying to fit them into the company's salary structure. Also, there's virtually no politics to contend with other than those in the beginning of contract solicitation. The Tax Recovery Act of 1981 has made contract research very attractive because the reimbursements of R&D investments makes it equitable for large corporations. The only thing you have to watch out for is if a company wants you to do an exclusive for them—I'd shy away from that. You should try to keep some part of the market for yourself because often, they make decisions that have nothing to do with the quality of your product, so you could end up developing a great program and witness the abandonment of your hard-earned labor."

As far as deciding whether to manufacture or license your product, DeFanti is adamant about the latter. Previous encounters are a true reflection of the difficulty he has had trying to market his own software, which is an essential responsibility should you decide to "manufacture" it yourself.

"Licensing," he said, "gives you the security of an already established marketing entity through the company to which you are licensing your product."

DeFanti was asked whether he had any ideas for a new product, assuming there were no limitations to resources or market viability. "I'm not interested in unlimited resources," he said, "because that means you have unlimited management problems. It's all I can do to manage SIGGRAPH, as chairman of the organization, teach at the university, work on software and direct my staff of 15 people at Real-Time Design. But I am interested in getting the whole design and implementation of custom hardware—silicon chips—to be a software, not a hardware, problem. Experiments at Stanford have shown that the software students do a better job of designing hardware than the hardware students because the hardware students try to save parts, which cost almost nothing, and the software students are doing involved data structures inside of the chip.

"Of course, this kind of design work involves computer graphics, so you've got a technology that is now designing itself, one that is being used to design itself. That's very exciting to me. After all, what's more intoxicating than technology as an extension of human cognition?"