2001 • Press and Documentation of Activities

Golan Levin

Press clippings, critical reviews, exhibition catalogs, and ephemera. Ordered chronologically.

002 Berlin Transmediale.01 Festival, 2/2001.
004 Berlin Transmediale.01 Festival, exhibition catalogue, 2/2001.
005 Broeckmann, Andreas et al. “Is software art a genuine artistic material?”  Transmediale.01, 2/2001.
transmediale.01
DIY [do it yourself!]
international media art festival berlin
4 - 11 February 2001
www.transmediale.de

Workshops

Experiments with artistic software, and discussions with:

Christoph Kummerer
Gameboy Hacking - Learn to Re-Programme your Favourite Toy
Monday, 5 February [in German]

Golan Levin
Audio-visual Suite for Synergetic Experiences
Tuesday, 6 February [in English]

Lego Mindstorms
Build and Programme Your Own Robots
Wednesday, 7 February [in German]

Karo Toons (Rothe & Lorenz)
Flash Animations
Friday, 9 February [in German]

Performances

Live audio-visual performances by
Golan Levin & Scott Gibbons (US)
Scribble
Using his own Audiovisual Environment Suite
Sunday, 4 February

Christoph Kummerer & Christoph Weber (A/D)
Pocketnoise
A sound performance based on a hacked Gameboy.
Sunday, 4 February

Julien Maire (F)
Diapositives
A presentation of micro-machines built into slides
Saturday, 10 February
Berlin Transmediale.01 Festival, 2/2001.

**Transmediale.01 DIY (do it yourself!)**
international media art festival berlin
4 - 11 february 2001
www.transmediale.de

---

**Artistic Software Award Competition**
(saturday 10 | 14.30-18.00: artist presentation | 20.30 - 22.30: award ceremony | auditorium)

**DJ I Robot**
Chris Csikszentmihalyi (US) 2000 [www.rpi.edu/~csik/research/]
Die DJ I Robot Software steuert die mechanischen Armbewegungen eines Robots, der einen Plattenti-
teller befähigt, um Töne zu erzeugen. Der DJ I Robot ist ein deutsches Beispiel dafür, wie Ideen in ma-
schneiche Kontrollcodes umgesetzt werden können. Der Code eines DJs ist eindeutig eine Frage des Stils. Der Jury gefällt DJ I Robot, weil er persönliche und subjektive Möglichkeiten der Codierung herbeizieht. Die DJ I Robot Software controls the mechanical arm movements of a robot to work turntables, spinn-
ing them to produce sounds. The DJ I Robot is a clear example of how ideas can be expressed in machine control code. The code for a DJ is clearly a matter of style. The jury likes DJ I Robot beca-
ue it highlights the personal and subjective possibilities of coding.
(on show in the media lounge)

**Audiovisual Environment Suite**
Golan Levin (US) 2000 [www.media.mit.edu/~golan/avses]
Audiovisual Environment Suite is an interactive programma, welches es dem Benutzer erlaubt, erstaunliche, sich endlos wandelnde abstrakte Effeekte in Farbe, Form und Ton zu
erzeugen. Die wahre Stärke der Arbeit sah die Jury in der umfassend experimentellen
Ausführung der Programmierung – ein wunderbares Beispiel, wie „programmieren selbst“
neue Bereiche eröffnen kann.
Audiovisual Environment Suite is an interactive programma that allows the user to create
dazzling, ever-changing abstract effects in colour, shape and sound. The jury felt the real
strength of the work being in the widely experimental nature of the software writing - a beautiful sign of how “writing it yourself” (WIf) can open new territories.

**Vexation 1**
Astroline Schnitt (F) 2000 [www.grodin.org/au/]
Zufall ist - wie bei Vexation 1 dargestellt - ein interessantes Thema der Softswarvkunst.
Während das visuelle Bild, ein gleichbleibender, dem Pong-Spiel entlehnter Puck, der
Vorhersehbarkeit unterworfen ist, ist die Bewegung dieses Pucks innerhalb seines Rahmens alles an andere als vorhersehbar. Die eigenartige Wirkung innerhalb der pro-
grammieren Begrenzungen ist faszinierend und überraschend viel länger, als man denken
würde.
Randomness is an interesting software art issue - as exhibited by Vexation 1. While
the visual image suffers from predictability, an unchanging pong-like puck, the move-
ment of this puck within the frame is anything but predictable. The quirky randomness within the program's restrictions is
engaging and is able to keep you guessing for longer than seems likely.

**Scribble**
Golan Levin and Scott Gibbons (US) [www.media.mit.edu/~golan/avses]
Levin hat als Teil seiner Doktorarbeit am MIT die Software Audiovisual Environment Suite entwickelt. Sie ist ein Set von
fünf interaktiven Systemen, die den Nutzer befähigen, in Echtzeit abstrakte Animationen und Klänge zu erzeugen. Jedes
von diesen fütten Teile ist ein experimenteller Versuch, ein Interface zu gestalten, das weniger flexibel und einfach zu bedie-
nen ist, andererseits aber auch unerschöpfliche und individuelle Ausdrucksmöglichkeiten im Visuellen wie auch im Auditi-
en bietet.
Scribble is now the Umsetzung der Programme in eine Performance an zwei Rechnern. Levin has a Composition ent-
worfen, welche das gesamte künstlerische Spektrum der Audiovisual Environment Suite ausschöpft und so ein einmali-
ges synästhetisches Erlebnis vermittelt.
Levin developed the Software Audiovisual Environment Suite as part of his doctoral thesis at MIT. The Environment is a
set of five interactive systems that enable the user to create abstract animations and sounds in real time. Each of the five
elements is an experimental attempt to create an interface. On the one hand, the interface is intended to be both flexi-
bile and easy to use, and on the other it should provide unlimited possibilities for individual audio and visual expression.
Scribble is a performance that adapts the programs for two computers. Levin has developed a composition that makes full
use of the complete artistic spectrum provided by the Audiovisual Environment and thereby creates a unique synaesthe-
ctic experience.

**Audiovisual Suite for Synaesthetic Experiences**
Golan Levin (US) [workshop i tuesday 6 | 14.00 - 18.00 h | room 131]
Zudem leitet Levin den transmediale Workshop Audiovisual Suite for Synaesthetic Experiences, in dem er zum einen die
Historie von synästhetischen, audiovisuellen Instrumenten aufzeichnet und analysiert, zum anderen neben seiner eigenen
Entwicklungsarbeit auch die interessantesten Ansätze von Künstlerkollegen im Detail - von der Programmierung bis zur
Anwendung - vorstellt.
Levin is also heading the transmediale workshop Audiovisual Suite for Synaesthetic Experiences. He will not only review and
analyse the history of synaesthetic, audio-visual instruments, but he will also present from programming to application
developmental aspects of his own work and interesting approaches of other artists colleagues in detail.

**xaflab3 : sonic wargame**
Anne Wellmer, ang, fischnp + Dave, Opstetien (NL)
Die Musiker verwenden ihre eigenwilligen, selbstgebauten Instrumente in einem vernetzten Wettkampf, benutzen,
samplen und attackieren das Klängmaterial der anderen.
The performers use their unusual self-built instruments in a networked challenge, using, sampling, and attacking each
other's sound material.
The Audiovisual Environment Suite (AVES) is a set of five interactive systems which allow people to create and perform abstract animation and synthetic sound in real time. Each environment is an experimental attempt to design an interface which is simple and easy to learn, yet can also yield interesting, infinitely variable and personally expressive performances in both the visual and aural domains. Ideally, these systems permit their interactants to engage in a flow state of pure experience.

The AVES systems are built around the metaphor of an inexhaustible and dynamic audiovisual “substance,” which is freely deposited and controlled by the user’s gestures. Each instrument situates this substance in a context whose free-form structure inherits from the visual language of abstract painting and animation. The use of low-level synthesis techniques permits the sound and image to be tightly linked, commensurately malleable, and deeply plastic.

The AVES systems inhabit a domain at the juncture of art, design, and the engineering of tools and instruments. As artworks, they extend an established Twentieth Century tradition in which artworks are themselves generative systems for media. As a set of tools, the AVES work represents a vision for creative endeavor on the computer, in which uniquely ephemeral dynamic media blossom from a close collaboration between a system’s user and designer.

http://acg.media.mit.edu/people/golan/avses/

**Andreas Broeckmann**: The Transmediale.01 has organised the first competition that includes an art award for software. This competition acknowledges the artistic work done by artist-programmers who are neither "interactive media artists" nor "net artists", but whose aesthetic material is code and whose form of expression is software programming. One definition suggested for software art is that it encompasses projects in which self-written algorithmic computer software stand-alone programs or script-based applications is not merely a functional tool but in itself an artistic creation. Does software serve a merely instrumental function, or does it offer new, creative cultural perspectives? Is computer code a genuine artistic material like paint or digital images?

**Gerfried Stocker**: Of course, it's still art, but I mean a lot of things are art. We know that playing Mozart at the Vienna Stadthalle is wonderful art. No doubt about it. We know and we call it art when we see the movies at the Berlinale. Art is just too broad as a definition and actually not really interesting enough for this process of development. We have to be confident enough to realise software art is a reaction to the general development of society and culture driven by new technologies, by digitalization, by the conversion of everything into code. There are only very few, limited possibilities for artists really to work with if they want to stay within this process and its dynamics of transformation. Otherwise they step outside and make interesting works of art, like for example Jeffrey Shaw or Bill Viola. Wonderful artists, but examples of what I mean...

**Unrike Gabriel**: We won't talk about them here, because they don't code...

**Andreas Broeckmann**: I think that you underestimate the development of art in the last 30 years. I think what you call radical and new in software art is the kind of radicality that every avant-garde try to claim to. I'm sure that there are three or four other avant-gardes going on at the moment where people claim that same radicality. And it's still art.
Daniela Pieve: I agree and I think that there are several ways to specify the code. I think I agree that saying "Here we need a tree or there we need something that makes decisions" is not enough. For example, in my case I was always really concerned about learning the very abstract formalism. And I took much longer than I intended. Maybe we should also talk about this delegation of the code and the disadvantages which this always brings with it. Of course you lose the quality feedback from the artistic material. I would always admit that and it was a painful experience. If you communicate with the programmers of course these people get more involved in the aesthetic concept by the time everything is ready. The other question is whether we want the same structure as in film production, where division of labour is fully accepted and how software art could be done in the future...

Colan Levin: My own background was originally as a painter and composer. It was only after many bad experiences trying to convince engineers to help me build my work that in the end, out of frustration, I had to learn how to do it myself. I would ask them to help me with things and they would say, "Oh, I'm a hundred dollars an hour." Or "I help you out of pity," or I'd lean over their shoulders and say "It's not right aesthetically," and they would say "I don't understand what you mean." In the end it was just necessary to dig in. It's not only a craft, but the craft is important. One has to be to in touch with it and then the concept is also important of course.

Ulrike Gabriel: I'd like to return to something Gerfried Stocker said. As an artist, you have a certain material you use. In Antoine Schneider's project, Variation II, you saw this black screen with a ball bouncing around there was a certain material used a certain space opened up. The space was very minimal. And the artistic material was used inside this space, with this definition of the parameters. This was adequate. In Gerfried's example, Bill Viola didn't even know what he was using. He did a job he couldn't do as an artist. That's why I have this very basic position: let's stay artists. Just go there, touch the code, use it, use it for your artwork, consider yourself an artist just in the classical sense, because then you keep this process.

Gerfried Stocker: I'm absolutely in favour of any type of software art. We're not at a trade fair here, selling software art, we're in a very intellectual community discussing problems and issues around software art. And we have to become more radical and more demanding. It should not be enough that somebody is writing code himself. That is not really enough to be a software artist. Because some of the works are very similar to this still very important approach of low tech artists: "We get control of our tools and this is important." This is totally important, but it is a different direction. I'm very interested in any type of work that is really using software in a way that reflects a new way of producing and making art in terms of the shift from document to event. Any digital picture, any digital sound, any sample is a stream of code, but it is not really software. It is still a document. And then we come to the event in a form of description of something that is really the algorithm which is reproduced every time. You have this process of converting text, coded into silicon memory and then it becomes something.

Audience/Tim Druderey: There are two things we should distinguish. One of them is that software is not just code, it's the algorithm. If I can code something in C, but I want to put it on the web and I can't program in Java and I get someone to code it in Java for me, does this mean that I handed over my role as an artist? If I formulate the algorithm to get someone else to code it, are they the artists or am I the artist?

Daniela Pieve: I've written down what you just said more or less literally in my notes. I'd like to bring into the discussion the notion of algorithmic art, instead of artistic software.

Colan Levin: I use algorithms in my work and I can't deny that sometimes the output of these algorithms can be lovely but I suspect that if you were to have a competition like this of algorithms themselves in the pure state I think it would be rather boring. You can go out and look at Donald Knuth's five volume set of software algorithms and see for yourself. It's really just a tool that we can use. And you could have a competition, but I think you need a really well-qualified audience to appreciate why "quick sort" is better than "relex sort" or "bubble sort". And why should we give an art prize to those people?

Audience/Tomoko Thiel: The works that we have seen tonight are mostly the sort of tools to generate sensory experiences. Maybe Daniela Pieve's came the closest to making the data and decision structures visible, which the jury suggested was what they were looking for or hoping to find through this competition.
Florian Cramer: Perhaps the sensory experience of what we just saw as demonstrations of software art works sometimes gets in the way of actually seeing the generative processes going on inside. If you look at the presentations tonight you get the impression it's media art as we know it. But imagine we would, for example, have included a self-replicating algorithm. We could have shown you a sample of C++ code or something like that. But we wouldn't have grabbed anything from that. The antithesis were at the threshold between conventional media art, which is very strongly focused on visual and sensorial representation, and a software art that is concerned about concepts, about algorithms. How can this be translated into a sensory experience at all? I'd be very glad if we could some day do without data beams. Data beams are an aspect of media art which I really hate since they exist. And everything boils down to presentations on data beams. I think we are aiming towards the invisible. This is really exciting. We are aiming towards processes which are no longer focused on interface representations. The fact that we now have many examples of software art which take the interface experience but subvert it or turn it upside down, marks something like the threshold, the end of one discourse and perhaps the beginning of a different discourse in electronic arts.

Gerfried Stocker: I'll take up the discussion about aesthetics in software art because it's a very problematic term. It's so strongly related to traditional art but it's a very interesting point really to look at what we're thinking of when we're talking about aesthetics in programming. I'd like to refer to Colin Lewin's work, which I admire very much. And the aesthetics of his code is not in the beautiful pictures or the nice sounds that he's making. What he achieved is to create something like a day. The whole interface becomes completely integrated, an integral part of the software itself, of the code and its expression. And I think this is really one of the greatest achievements that software art can do no matter what it produces, whether in the end it's nice pictures, nice sounds or maybe some virus that is so well done that it's able to bring down the whole of the Internet.

Andreas Brackmann: You'll have to explain, though, why this full integration as an aesthetic model is superior to, for instance, friction.

Gerfried Stocker: Friction might also be very interesting. It's a question of these different levels of writing and inventing the code, creating an interface as technology or the interface in its sense as the surface where communication can happen with the project and the result. This integration is very important, but it's not really creating friction, it's creating nice pictures. But I said the pictures are not important, nor is the friction really important. And that is why I don't accept a definition of software art that only refers to viruses and codes and this subversive aspect and would also not accept an art form which is only referred to nice pictures and nice sounds. The quality itself lies in a completely different area. There is a potential danger that we are looking at the wrong thing.

Rafael Lozano-Hemmer: These things are not mutually exclusive. On the one hand when we're talking about software in a very fetishistic way, like Gerfried Stocker is doing, I share that kind of formal attraction to the fetish of the code. At the same time we have to be fully aware that in no way is the code not socialized. In other words the code, as we all know if you program in different coding languages, is in itself a kind of collaboration with the people who wrote the code. So to the degree that there is no purity in that kind of expression, I'm interested in the fact that code itself is social. And you can say the opposite thing for instance with respect to other works that like mine are more assemblages of other kinds of media or other kinds of social patterns. In that we are working on a code as well, except that it is not a software code. It may be a language code or a political, social, or economic code. I think Colin Lewin's work is extremely social and the way in which my body reacts to his code in terms of the choices he made for the effect of the images and sound is extremely social. To me, that's where the value lies, whether he did it in Director or otherwise, I think it's meritorious and interesting.

Gerfried Stocker: I would agree. That's the very important point of the interface which is one part of this change. And I think it would be very wrong to divide these things. Just to make it clear again, I was specifically referring to the aspect of aesthetics. What could be the aesthetic in the type of art? There are unfortunately criteria in looking at art other than only aesthetics. And software art also has to refer to these other criteria.

Anne Negten: I think there is a difference between artistic code and software art or artware. I think that's something not to be confused. Personally, I'm not that interested in the aesthetics of the code, but what is created through this code or what this code will bring about. Then I'm not referring to the common notion of art again, but really to a much broader experience caused by the code.

Daniela Plavec: I still wonder about the transparency issue. How can we communicate to the audience something of the complexity we are all interested in? Do we want that or are we happy with a very specialized audience? Like Bellabrara says: machine aesthetics and the pure optimized algorithm. So if the mathematician finally says what is great and what is art, then we open up the whole spectrum of what we are talking about mathematics and the beauty of mathematics, too. Since there is a lack of definition of art, I think I would even agree if mathematics was included in the future.
To our knowledge, this is the first award given and solely dedicated to software art. This award is not about what is commonly understood as multimedia - where the focus is on data that can openly been seen, heard and felt. This award is about algorithms; it is about the code which generates, processes and combines what you see, hear and feel.

The mere fact that the transmediale artistic software award is the first of its kind proves that algorithms have a longer history of being overlooked in the perception and criticism of digital art. While the code of a mere image, sound or text file passively relies on other pieces of software in order to be perceivable and editable, program code actually actively manipulates the machine. Perhaps the most fascinating aspect of computing is that code - whether displayed as text or as binary numbers - can be machine-executable, that an innocuous piece of writing may upset, reprogram, crash the system, that the Internet and almost any electronic device technically depends on writing. The word processors we use on computers are typewriters built from writing.

Your word processor behaves typewriter-like because the programmer chooses certain functions of the mechanical typewriter to model in code. It is modest self restriction on the part of the programmer that the word processor adheres to the typewriter model instead of filling your writing with random letters, twisting or erasing bits at will or improvising new writing from your words.

Unless you know the program source code, you can't tell whether your E-Mail software sends your mail to the destination you specified or whether your operating systems actually stores your files and won't delete some of them on every February 29th because it contains some malicious code. Computer viruses might be seen as a critical form of software art because they make so-called users aware that digital code code is virulent. Computer software is not simply a tool.

Every program that pretends to be a tool disguises itself. You expect that 'Save' will save and not erase. This feeling that you understand and
control what the software is doing in the machine can only be based on trust in the programmer.

For us, software art is opposed to the notion of software as a tool; not because we would want to differentiate some kind of high art from some kind of low craftsmanship of programming. Instead, software art has the potential to make us aware that digital code is not harmless, that it is not restricted to simulations of other tools, and that is itself a ground for creative practice.

Software art could be algorithms as an end to themselves, it could subvert perceived paradigms of computer software or create new ones, it could do something interesting or disrupting with your computer, it could be creative writing, it could be science.

Since we talk about algorithmic code when we talk about software art, we also talk about programming languages - perhaps even poetry in programming languages -, and we talk about the difference of source code in programming languages and executable compiled code. Unfortunately, we ran into problems here that might be telling how new the playing field is that this award has opened. With the single exception of Axel Roch’s project, the jury didn’t receive the source code of any of the 49 contributions entered for the competition. Also, no works were entered in which algorithms or program code was part of its own presentation. We even had a high amount of entries where we could judge no software at all, since only video tapes of software-based installations were sent to us. Nevertheless, we are convinced to have entered an exciting field; a field which many artists raised in other new media still have to reckon with, while many other artistic coders we know didn’t send entries.

We could well have imagined to award something very simple - an elegant or a disturbing piece of program code, a sophisticated oneliner that blows one’s mind and perhaps even one’s machine. We have imagined many aspects of artistic software code which we partly found and partly didn’t find in the entries.

What we didn’t or couldn’t see were:
- algorithms
- meta-code
- code-modifying code
- efficient code
- beautiful code
- code as diary (e.g. autobiographic writing, code that contains the history of it’s own creation.
- code libraries
What we saw were:
- synthesizers (i.e. algorithmic generators of data)
- filters (i.e. algorithmic processors of data)
- code as social platform
- code as environmental control
- code as glue
- obscure code
- ironic code
- simulated code
- missing code
- non-working/impossible code
- application code
- self-disguising code
- insignificant code
- code as writing
- disruptive code
- code as attitude

There was no rule of thumb for us except that we excluded work we either didn't find interesting as software, or which we didn't find interesting as artworks, or both. We did not award a first price, but rather saw the entries we shortlisted as signals; signals from where software art departs and where it is tending towards.

Vexation 1
Randomness is an interesting software art issue. There is, at the core, the many problems of the generation of random/pseudo random number sequences. Further, how are those sequences incorporated into the code? To what aspects of the software are they applied? Too much randomness deteriorates into noise - boring. Too little randomness is regular and predictable - also boring. Dancing along the chaotic edge takes practice and greatly influences the pacing, attitude and apparent depth of the software. One particularly appetizing flavor of randomness among the entries is exhibited by "Vexation1" by Antoine Schmitt. - While the visual image suffers from predictability, a unchanging pong-like puck, the movement of this puck within the frame is anything but predictable. The quirky randomness within the program's restrictions is engaging and is able to keep you guessing far longer than seems likely.

Audiovisual Environment Suite
Audiovisual Environment Suite by Golan Levin was a large and involved entry which took a while to sort through. While the software was presented as a nondescript interactive installation, the jury felt the real strength of the work was in the widely experimental nature of the writing.
The Flocculus software in particular was a beautiful sign of how writing it yourself (WIY) can open new territory. Visiting the author's website shows his depth of coding and range of experimentation. To what will he finally apply all this visual language? After making his discovery, the author seems less decided as to what to do with the code, often putting the choice in the user's hands. We think Mr. Levin could be more articulate in defining how the software is presented. As a source of writing, the motion and shape of the Flocculus seems to have already created a rich vocabulary and he could spend more time experimenting with what can be said with it.

DJ I Robot
Okay, okay, we admit the DJ I Robot is cool and the jury would love to see it perform at the Transmediale festival - especially in competition against human DJ's. But the reason we choose to shortlist the DJ I Robot is that it is a clear example of how ideas can be expressed in machine control code.

The area of overlap between the timing of the machine and musical time is an open ground for improvisation in coding. The DJ I Robot software controls the mechanical arm movements of a robot to work turntables, spinning them to produce sounds. To make the DJ good, the timing code must be more than functional, it must also be musical.

Here are some things to consider about the robot's code: What is a good 'DJ algorithm'? How does one create a model for spinning records? Can human DJ movements be sampled and filtered? To what extent is the mood of the party accounted for as feedback in the code's decision tree? Would your DJ algorithm be the same as mine?

The code for a DJ is clearly a matter of style. We like DJ I Robot because it highlights the personal and subjective possibilities of coding.

**HOTLIST**

**JOHN F. SIMON, JR.**

TRANSMEDIALE (WWW.TRANSMEDIALE.DE), a media-art festival and competition that takes place in Berlin this month, is introducing a new category devoted to software art. As a judge of that part of the competition (along with Florian Cramer, a lecturer in comparative literature at Freie Universität Berlin, and Ulrike Gabriel, a media artist, and with the guidance of artistic director Andreas Broeckmann), I was most interested in artists who had trained on computers in art school, outgrown commercial applications, and turned to writing their own code. Their work often critiques the limitations of the software industry and its products. The result is a kind of creative writing that, as German media theorist Friedrich Kittler puts it, “gains the enormous power to do what it says.” Here are four examples:

**Code as Parody:**
Adrian Ward (UK), “Signwave Auto-Illustrator”
www.auto-illustrator.com

This self-described “parody” software challenges the way we expect commercial software, like Adobe Illustrator, to behave. Rather than obediently track your cursor, Auto-Illustrator turns circles into smiley faces; worse, its pencil gets out of line. On his website, Ward poses the question, Who is the artist: the coder or the user? Think it’s the user? Auto-Illustrator will fight you for it.

**Code as Behavior:**
Antoine Schmitt (FR), “Avec Détermination”
www.atrinom.org/en/avdecodetermination

The small animations at Avec Determination are like character sketches. Pairs of leglike lines bounce around a box, alternately “standing,” “resisting,” “behaving,” and “not behaving.” Programmed to act according to the laws of gravity, to the figures’ “intentions,” and to various flavors of randomness, these animations elegantly illustrate the desire to have programming, rather than user participation, be the determining factor in the viewing experience. Schmitt points out that software artworks are different from other autonomous artworks—like Jean Tinguely’s animated assemblages, say—because software is created with words. Thus, “it is able to transform itself by acting on its own description.”

**Code as Attitude:**
Andy Deck (US), “Artocontext”
www.artocontext.com

Andy Deck writes personal alternatives to standard software. His site features an array of projects designed to loosen what we see as the stranglehold that big business and the media have on software design and the flow of information on the Internet. “Open Studio,” for example, is Deck’s take on groupware (i.e., software programs designed to enable corporations to hold meetings online). But instead of the rectangles and flowcharts usually found on such sites, Deck offers concentric circles and brushes that produce bar codes. His tools are not there for global financial planning but for the submission of it. By making variations on standard industry models, Deck’s code has the clear attitude of “his” and not “theirs.”

**Code as Environment:**
Golan Levin (US), “Floo”
acg.media.mit.edu/people/golan/floo

Golan Levin’s projects originate in his work at the MIT Media Laboratory with the Aesthetics and Computation Group. His site allows visitors to work within a program called “Floo” to create wispy digital nebulae. Floo is the Web version of a performance/installation (Audiovisual Environment Suite) presented at last year’s Ars Electronica in Linz, Austria, involving computer-generated audio work as well as visual programs. Levin has a knack for finding interesting visuals through the manipulation of various algorithms. But, as with many of today’s media artists, he might best be served by spending less time on the performative aspect of his work and more on articulating his ideas in code, which is, after all, what it’s all about.

---

Jane Harris

---

John F. Simon, Jr., is an artist living in New York City.

---

**Wunder Boys**

It used to be that only magnates and maniacal rulers could afford enough objects to assemble a “significant collection.” Today, anyone can do it: Peruse the data on the Net, download what intrigues you, and share it with others in your very own “cabinet of curiosities.” After all, what’s a blogger if not a Peter the Great in digital clothes? In theory, the Wunderräum structure should work well on the Web. In reality, most sites that define themselves as Wunderräume are more akin to junk stores than to treasure troves. Look hard enough, though, and you might find gems. Case in point: Marek Walczak and Martin Wattenberg’s ongoing project WonderWalker (wonderwalker.net/art), a viewer-participatory critique of museums and the politics of archiving commissioned by the Walker Art Center in Minneapolis. The idea is for the visitor to insert an “object” (via a link) into a “personal cubbyhole” and then to link that object to at least one other.

Some of the more interesting links include www.randomaccessmemory.org by “Aleks,” a site where one can store one’s memories by date, name, and subject; and www.ashab.freeonline.co.uk/801/Abduct/BettyHill.html by “Rudie,” an article by Barbara Becker posted on an alien-abduction website about “The Betty Hill Conspiracy” (a well-known 1961 abduction case). John Smith’s contribution, www.andersenwindows.com, is an online catalogue for a window maker; postings like this one call into question Wattenberg’s stated belief that people “collect with a higher standard in mind,” knowing that others will look at their choices. Too bad there isn’t a de-accession option in the program—I bet Peter would have had one.
WWW.SINGLECELL.ORG
FROM GOLAN LEVIN

The purpose of this extraordinary website is to investigate “new life forms” that have sprung up on the internet: a sort of online zoology if you like. Singlecell, its makers explain, is “a monthly bestiary of these newfound species: a collection of online life-forms discovered and reared by a diverse group of computational artists and designers”. Actually, it’s an experimental space set up by Design Machine’s Golam Levin for fellow designers to create Flash pieces around the theme: a new “interactive life form” will appear each month, with contributions from Levin himself (Obozok, shown below right, is a sort of amorous, friendly Spacehopper that undulates in harmony with the movements of the mouse), Peter Cho and Joshua Davis, plus UK designers Danny Brown (Mosquitoes, shown, below left) James Tindall and Ed Burton. Yet another fine example of the worldwide web as a space for likeminded creatives to play and learn.

I:CUBE PROMO
FROM LE VILLAGE, PARIS

There’s an awful lot of animation about at the moment; nevertheless the really imaginative styles and techniques still manage to stand out. This promo for iCube track Adore is a case in point. Made entirely on a G4 it features, very simply, a line drawing of a man running to catch a plane. He sprints through the airport, gets on his plane, joins his fellow passengers and takes off. That’s it, but while it doesn’t sound too riveting, the movements of the man, combined with minimalist drawings of airport structures and decorations (Benoit Millet is a graphic designer and architect and it shows) lie in to the pulsating, hypnotic music in a very fluid way which makes for great watching. “The music by iCube inspired us to do something light; an ‘easy watching’ style,” claim its directors modestly – the fact is they’ve actually achieved a lot more. Directors Benoit Millet and Sebastien Debey

BRITISH AIRWAYS POSTERS
FROM M&C SAAITCHE

British Airways now have flat beds in business class. These flat beds (which, I’m sure we all agree, are an excellent idea) have been introduced to carriers flying to Chicago, San Francisco, Hong Kong and Washington. M&C Saatchi therefore had the tricky task of combining two pieces of information in one ad, and to their credit the solution they reached is both informative and dramatic. Using the word “flat” as their point of reference, famous US landmarks are shot from unusual angles and cropped to emphasise the flat and horizontal lines of each composition. The saucy line: “Lie back and think of...” also sporadically appears. The best advertising posters can become public art which, when run in sequence as they are at the Bank Underground station (shown, left), this campaign can lay claim to. British Airways has a reputation for great TV work, these posters show how well that grandeur can translate to other formats. Creatives: Tiger Savage, Mark Goodwin. Photographer: Christopher Griffith

> Magnetics
> Golan Levin & Katrin Grotepass
> 5 April 2001 - 26 April 2001
> Reception Thursday 5 April, 6-9pm
> 
> Moving Image Gallery
> Michele N. Thursz, Director
> www.movingimagegallery.com
> michele@movingimagegallery.com
> 
> 414 Broadway #3
> New York, New York 10013
> 212.966.4741
WHAT WORLD EVENT OCCURRED THE YEAR OF YOUR BIRTH? Watergate.
WHAT MAKES YOU TICK? My friend Kelly says "conveyor belts."
WHAT'S YOUR FAVORITE OBJECT? Stainless steel belt clip.
WHAT'S YOUR FAVORITE CURSE WORD? Shazbot.
WHAT ARE YOUR INFLUENCES? Bauhaus. Alexander's Pattern Language, the Bee Gees.
WHAT DO YOU WANT TO CHANGE? Myself.
WHERE DO YOU WANT TO GO? Boom.
WHAT'S ANOTHER NAME FOR YOUNG GUNS? Never mind the bollocks, here's the Sex Pistols.

Mark Tribe: net.ephemera

Curatorial Project: Group Exhibition
Moving Image Gallery, New York, NY
May 3 - May 31, 2002

This curatorial project grew out of a conversation with Michele Thuesen, director of Moving Image Gallery in New York. We were discussing the difficulties of exhibiting net art in galleries and museums. Most net art is meant to be experienced as a solitary encounter. And many net artists saw their practice as oppositional to art world institutions. Putting net art in the gallery involves a recontextualization that can radically alter the experience and significance of the work.

net.ephemera was an attempt to work around this problem by exhibiting ephemera—drawings, diagrams, notes, receipts and other physical artifacts—related to the making of net-based projects.

Click here to view the net.ephemera exhibition

Colen Levin:
BACTERIAL DEVELOPMENTS AT SINGLECELL

Singlecell.org, the experimental design site that invites well known Web designers to contribute, is showing work by Joshua Davis. The June issue, entitled 'Bacteria', is a simulation of the development and growth of micro-organism colonies, and was run for six days on Davis' own site, playstation.com, at the end of May.

Davis explains: "My piece was built trying to harness random growth patterns over time without bogging down a visitor's CPU. There are two invisible points which continuously place themselves randomly around the screen. The project has a library of shapes to choose from. Once it has chosen five total shape compositions, it finds the two invisible points, determines their distance from each other and uses this value to create the width and height of the composition."

While this may sound complex, Davis believes that Flash, the format used to create Bacteria, was the least powerful option he could have chosen.

He says: "To date I'm the only member of Singlecell using Flash, which is the most interior of the technologies being used in the project and with it comes some serious issues. But I don't think we ever encountered problems. Some of my best tricks are accidents."

The format of the work displayed at Singlecell varies from contributor to contributor, with pieces created in languages such as C, Java, Lingo and Actionscript. In an attempt to bring Web design to life.

The project is run by conceptual artist and composer, Golan Levin, who prefers to maintain his anonymity by being referred to as the Zookeeper. Levin contributed the first installment on the site, and prefers his involvement to be perceived as being as minimal as possible.

He says: "I'd prefer that the Zookeeper isn't perceived as anything more than an administrator. Sure, I've got the keys, but most of the time I'm just changing the water and sweeping out the cages."

The project will continue to the end of the year, and it is as yet undecided where it will go from there. Hopefully, however, the pieces on the site will give other artists, designers and programmers a new perspective on their creativity.

www.singlecell.org
TELEPHONE MAN
GOLAN Levin TALKS ABOUT CONDUCTING A SYMPHONY
OF A “HUNDRED LITTLE X’S.” IN THIS CASE, CELL PHONES

interview by Tanya Bezreh

Golan Levin is all packed for Austria, funding more or less in hand, staff lined up, space in the Ars Electronica Center’s offices secured. He will be in Linz for two months, assembling a concert that will premiere at the September festival. The concert, called DialTones, will be a “telesymphony” (www.telesymphony.com) played entirely with the ring tones of the audience’s cell phones. At last year’s festival, Levin won the Award of Distinction in the Interactive category for Audiovisual Environment Suite, audiovisual software that was performed as a quartet. He’s nervous discussing a concert that hasn’t even happened yet—and one that is dependent on good luck and tight production. Alarmingly, the night we talk, Levin has just checked his e-mail only to read a forwarded AP article with the headline “cell phone symphony,” about an orchestra in Israel that introduced its symphonic highlights with classical-music-ring-tone samples.

Artbyte: Did reading that “cell phone symphony” headline freak you out?
Golan Levin: Of course. One always wants to be the first, even though there are thousands of times that one finds out one is not. When I did my master’s thesis I obsessively dug into the history because I really wanted to make the point that there are thousands of predecessors to everything we do, and all of these pioneers are frequently overlooked. And some aren’t pioneers, some are total cheesemeisters. (Laughs) And the idea is current. It’s not me who is having the idea, so much as the world is having the idea.

AB: Are there cell phone symphony cheesemeisters?
GL: Anything involving a lot of X’s all playing at the same time is a schtick, a gimmick. And it is the challenge of the artist to surpass that, surpass the materials that they’re using and hopefully, ideally, make a profound statement. I want someone to write an article about the historical overview of all the people who’ve done music with a hundred X’s. Laurie Anderson did that car horn symphony in 1972. Obviously, she had precedents as well, like Cage.

AB: Have you ever seen any precedent-setting “quantity performances” live?
GL: I missed the Perry Hoberman thing in ‘99 with the blenders [Thick Shake at Ars Electronica 1999]. I was born in the year that Laurie Anderson did the car horn thing; I’m sorry I missed that. The one that I like the best is Maywa Denki [Japanese avant garde performance duo Masamichi and Nobumichi Tosa], because they went one level of abstraction higher. They did that in Japan where we’re just going to have outlets that we turn on and off, and you can plug whatever you want into the outlets. If you want 100 blenders, put 100 blenders in there. If you want 100 drills, put a 100 drills or 100 saws or 100 whatever. It’s a nice metaapproach because they identified this phenomenon of a hundred X’s.

AB: How did you approach Ars Electronica?
I told Gerfried [Stocker, the Center’s director] about it in October, and he said it sounds expensive. And I said, “It’s a hundred grand.” And he said, “Get the fuck out of here.” Then I went to Berlin in February for the Transmediale. A friend said I should talk to these two underground phone hackers who work for him. They’d been involved in the scene for 20 years, doing interesting things with phones. They laid it out for me. They said, “That’s what you want to do? Here’s how you do it.” I was like, shit, that’s actually affordable. And with that information, I changed the proposal, went back to Gerfried, and said, “OK, I can do it for fifty grand. And he said, “Go raise the money, and we’ll talk.”

AB: So you shopped this proposal around?
GL: Exactly. I was very lucky to get some help from Thundergulch.org. I’ve been very lucky with arts organizations and grant foundations. They’re in the business of giving out money. Cellular phone companies are not in the business of giving out money, however. So it’s taking a longer time.

AB: Which is to say what? There’s still equipment that needs to come through?
GL: There are a few things we need. Think about what it means to have an orchestra of phones ringing simultaneously. The idea of making music with a phone is not original; cell phones have had ring tones for several years. The new medium, in this case, is the mobile switching center [the system that enables calls between mobile phones, as well as calls between mobile and standard phones.] That’s something the average person does not have access to. If you think about the history of new media art, the pioneers are people who have first access to
something that was formerly technologically or financially prohibitive. In the late ’60s, early ’70s, computers were these rare things that only existed in big research labs. Computers were the size of this room. You’d be this remarkably rare artist who got the chance to work with such things. New media art is still defined by that kind of accessibility. Who can personally place 30 simultaneous phone calls? Probably, in five years, everyone can do it. Or ten years. Today, the average person, musician, composer, or artist, does not have access to 100 outgoing phone lines. That’s infrastructural.

AB: What about those robot calls you get on the telephone? Are they calling 30 people at a time?

GL: There’s probably some 16-year-old...a really fucking good hacker, who figures out some way to hack into the net2phone API and starts making random phone calls for giggles. It could happen. Networks are vulnerable, and someone with a similar idea could achieve something. The theme of Ars this year is “takeover.” It reflected the idea of a person, namely me, or my group, taking over an infrastructure that’s not normally available to us.

AB: Do you think it’s going to be carefully composed, or will there be room for improv? What if only 50 people show up, versus a thousand people?

GL: That’s a great question. I think we’ll have a much more accurate idea about that the week of the performance; we’ll know what we can support. There are two main elements of chance that play into the piece. The one that’s up to the technology is unpredictable dialing delays. [The other element] is up to the audience. This is played on their mobile phones. If I give them mobile phones, that’s much less interesting. Come as you are, and you are prepared to be part of this larger thing. And everyone has a cell phone. It’s just a fact. Even if everyone doesn’t have a cell phone, everyone has a cell phone. All you have to do is just show up, because you’re already carrying all the equipment you need. But equipment varies. Some people have older cell phones. Some people have newer cell phones. Some people have shitty cell phones. Some people have very fancy cell phones.

AB: What’s the cell phone or the ring tone that you most fear?

GL: I have to embrace; I can’t fear. I don’t want this to be an ad for Nokia, but Nokia has the best phones. Their phones can accept new ring tones over SMS. Ericsson makes otherwise fine equipment, but for some reason, they never implemented ring tone transmission. So if someone in the audience shows up with an Ericsson, what am I supposed to do? I have to embrace that. I can’t control their handset. If they want to stick it to me, they’re going to put the William Tell Overture on there.

AB: So why are you doing this?

GL: Because I have this weird combination of wanting to, and I think I can. But mostly it’s for the sake of the idea itself. I want to know what it will be like. I know it’s a cheesy answer, but from a phenomenological perspective I think it could be really interesting, really stunning. Iannis Xenakis, the composer, had a great quote: “The problem with electronic music is that it all sounds the same, because it always comes out of the same two speakers.” And this is about radically shifting that, to create an environment where there are 500 speakers around you, and they’re all spatialized. Hell, yeah, I want to know what it’s like to be surrounded by that. It’s not just for the sake of the stunt. The challenge is to transcend the stunt, to surpass that medium, make it more than schtick, make it meaningful, personally or socially.

AB: How are you going to make it meaningful, personally or socially?

GL: That becomes a matter of making moving music, and that is something I don’t think anyone can describe how to do. It has to succeed formally. As a composition, it has to do all the dramaturgy and choreography that a piece of music has. It moves you, lifts you up, brings you down, and takes you to some interesting space. The mise-en-scene has to support that as well and make it an interesting experience for the people who are there. The piece will hopefully be a metaphor for, or alert you to the fact, that all around us there is this symphony already happening. And I’m just gathering it into a place of unusual density.
What is the basic premise of the Singlecell project?

Singlecell is an experimental online bestiary featuring monthly contributions from computational artists and designers from around the globe. The project has several overlapping goals. The first is simply to explore new possibilities for expressive computational art and design online. Generally speaking, people’s concepts of what is possible tend to be limited to the affordances of the commercially available design tools, and to the constraints and objectives presented by commercial web design jobs. So with Singlecell, our goal is to suggest what is really possible when designers have the liberty to make something personal, and the talent or skills to invent their own means in doing so. As it happens, the theme of newly-discovered species of online interactive life-forms is an interesting and agreeable pretext for binding together our work.

But really this is just a metaphor for the second goal of Singlecell, which is to gather together and create a community for a new species of designer: those “computational” artists/designers who write their own software using languages like C, Java, Lingo and ActionScript in order to make their artworks come to life. A third goal of Singlecell is more subtle. There really aren’t many computational designers out there: people who are commensurately skilled in both computer programming and visual communication, and who are interested in sharing their work online. When they do present their works online, it’s often in extremely varied contexts and to varied degrees of polish. By holding its designers to a common theme, and providing each of them with only a month to do it in, the Singlecell bestiary provides a common window into these artists’ individual aesthetics that allows its visitors to more clearly appreciate their similarities and differences, and the possibilities of online computational design as a whole. In sum, Singlecell’s goals are to make a statement about a new way of making things, to bring together a handful of the people who make things in this way, and to provide a view into some of their similarities and differences as individual creators.
Why Singlecell?

There are no major projects on Singlecell, no Gesamtkunstwerks, no career masterpieces. Singlecell is a side-project of side-projects, a compilation of small pieces executed by artists who probably had something larger and more important that they should have been working on. Nevertheless, although Singlecell is a small project with modest goals, it is also an exceptionally clear example, on multiple levels, of a wholly new mode of artistic production. Of course, its focus on the medium of “computational design” is contemporary, made possible by a new breed of designers who are skilled in both programming and the arts. To this, however, is added the way in which these people have come together. Many of the contributors have never met face-to-face. Most of them would never have even known about each other had it not been for the Internet. Quite a few of them might never have known about each other had they not independently agreed to participate in the Singlecell project. Singlecell is interesting because it precisely prohesies the new form of art product, practice, practitioner and community made possible by the Internet revolution.

Warum Singlecell?


Gerfried Stocker
Moving beyond technical feats and "interactive" puzzles, web developers are tackling the visual interface of the world wide web has existed for several years. Newer forms were fashioned into icons, navigated and pried open its multi-disciplinary potential. Graphic designers, sound artists, creative writers have combined the software and sculptures of computer software with the social sculptures of computer networks. Fitting in a variety of art historical lineages, from Joseph Beuys to John Cage, artists working on the internet have incorporated the computer's cultural digital impulses and addressed the implications of an emerging art form. This task attempts to present certain backbone characteristics of what it means to work on the internet and to examine some of the most significant and clearly defined strategies: interactive, multi-user, and stand-alone. Moving beyond the user-driven nature of the internet. A significant interactive system is a "living" computer envi- ronment with a graphical user interface. As an artist and game designer Eric Zimmerman says, "games are among the most ancient and sophisticated forms of designed inter- action," and artists have learned to combine this tradition with digital tools to create compelling games. Natalie Bookchin’s The Lieber records a short story by inaugurates a "Sommergang" with the testing of the JOVD's ASDG, as users struggle to navigate a new medium, the user interface is not yet a comprehensible and uncontrollable streaming gibberish that over- takes the screen, as a virus would a computer. More often than not the user is subjected to a blinding wall of colors and shapes, a user interface that is a reminder of the ubiquitous presence of computers. The keyboard interface is a minimalistic device, a control that, when used properly, can add to the user's enjoyment of the internet. Chat rooms set up around the many tal- ents of the artist in action of Playboys Playdate in a take-up countless hours of endless creations. Bringing animation to such a complex platform is a challeng- ing task, yet contemporary artists and users have found a way to satisfy and further explore that sense of immediate feedback and instant feedback. The successful projects leave it to the users' movements, decisions, and fantasies to animate the screen. In 1995, Eric Zimmerman and World Wide Web's SICYPF 2000 created a new form of interactive animation. With real-time interactive fights in a schoolyard, Office of Art in Education, a user interface, new environments, other artists find quicker ways to ani- mate the screen. Only across the web would it be possible to present the screen in a canvas. Along with Mark Napier's p-Soup, Andy Deck's Open Studio is literally that: a blank canvas on which logged-on users draw and scribble, adding their intervention to the ongoing drawing in real-time to the "collaborative" drawing. The process translates the realities of a real-world creation into animated form. At no point in time can the drawing be static and no movement invisible. John Klima urges online users to share and mix their ideas with small files in a 3-D musical instrument called Glasaed. Interested in hybrid visual-audio-interac- tive experiences, the artist has created a way for users to animate compositions in real-time. In each case, it is in the artist's absence that the animation happens. The internet's most persistent virus is the same as the artist's. The abstraction requirement for interactive art, by making its contribution to a web of predetermined nature of the internet. A significant interactive system is a "living" computer envi- ronment with a graphical user interface.

BASICRAY With the development of information technologies, the question of interfacing content adequately becomes increasingly important. Human-machine interaction in interrelated information contents fundamentally challenges semiotic orders, body politics and identity. BASICRAY is a virtual broadcaster technology. BASICRAY represents the emergence of new bio-electronic semiotics which has been translated into human discourses by providing a hyper real human interface. BASICRAY ("pure light") in Japanese is translated into the visual language of a text and image in the form of a physical object. With real-time animations, the artist can create a piece of art, a semiotic installation right down to every nut and bolt, provide a compelling and realistic VR walkthrough, and send off blueprints to a fabricator for an estimate. At the same time, the artist can create an incalculable object that only exists on a Gateway handheld comput- er. The elements of both can be combined, so the virtual meta components heighten and extend each other.

JOSHUA DAVIS I've been reading a ton of painting essays on Jackson Pollock - who he had this brilliant quote of calling himself a painter, even though the brush never hit the canvas. Similarly, I would consider myself an artist, though I have no control over what this new engine outputs. In one sense I program the paintings, the brushes, the canvas, the strokes, rules and boundaries. Heavily inspired by Grossman, I too am in a constant state of sur- prise and discovery, because the machine may structure forms that I had never had thought to execute.

MARK NAPIER My work is not animation in the traditional sense of that term. It is animated, meaning that it moves over time in a semi-pre-determined fashion as it is read and re-determined by the engine outputs. In one sense I program the paintings, the brushes, the canvas, the strokes, rules and boundaries. Similarly, I work with the machine and structure forms that I had never had thought to execute.

ANDY DECK I became interested in the process as it applied to my personal development. I was interested in the potential of animation and about computer programming. The initial impulse to use computers involved a desire to present all the develop- mental phases that my images went through before completion, but eventually a "virtual" animation of the process was developed. The final product was a series of sixty still images and a short animation that involved the entire process of making the video. The animation is a self-contained animation that involves the entire process of making the video. The animation is a self-contained animation that involves the entire process of making the video.

Come to Carnegie Hall for American Composers Orchestra’s “Technology and the Orchestra,” the culminating event in Orchestra Tech, ACO’s five-day exploration of new music by today’s most wired composers and performers.

Sunday, October 14, 2001 at 8 pm
Pre-concert talk starts @ 6:45 PM on the Carnegie Hall stage with the composers.
Free to ticket holders.
Carnegie Hall, 57th St. @ 7th Ave.
ACO Technology and the Orchestra
Paul Lustig Bankelm, conductor
TRISTAN MURAIL Le Partage des Eaux (US Premiere)
TOD MACHOVER iFrobah (World Premiere, ACO Commission)
MORTON SUBOTNICK Before the Butterfly (World Premiere, Digital Version)
EDGAR VARESE Déserts (with video created by Bill Viola), (NY Premiere)
French style and programs are supported by the Themen & Todes Foundation.
Tickets: $47, $36, $17. Call CarnegieCharge at 212.247.7800, or online at www.carnegiehall.org.

It’s all about technology and the orchestra, and it culminates at Carnegie Hall with premieres by Tod Machover, Music Alive Composer In Residence, and Tristan Murail, plus newly-updated music by techno-pioneers Edgar Varese and Morton Subotnick.

And Hear Today’s Most Hi-Tech Composers and Performers
Orchestra Tech explores some of the 20th century’s earliest electronic experiments as well as today’s sophisticated hyperinstruments, interactive computer music environments, and multimedia. Come hear how technology is redefining the orchestra of the 21st century.

Wednesday, October 10, 2001 at 8 pm
Pre-concert discussion @ 7pm
Miller Theatre, Columbia University, Broadway @ 116th St.
ACO: Orchestral Tech Opening Concert
GILI Rose and Jeffrey Milansky, conductors; Steven Schick, percussion
JOHN OSWALD >>Concerto for Conductor and Orchestra (NY Premiere)
OTTO LUERING >>Synthesis
EDMUNDO CAMPION >>What goes up...
(DY Premieres)
DAVID FILER >>In Between (NY Premiere)
RANDALL WOLF >>Hae Haw

Thursday, October 11, 2001 at 8 pm
Merkin Concert Hall, 129 W. 67th St.
Speculum Musicorum
MARK APPLEBAUM >>Skipo walks up
JAMES MOBERLY >>Sogghorno
JOSHUA FINEBERG >>Empreinte (US Premiere)
MATHIEU ROSENBLOOM >>Kuan Tzu

Friday, October 12, 2001 at 8 pm
Pre-concert discussion @ 7pm
Great Hall, Cooper Union, 7 E. 7th St. @ Third Ave.
Etel: Electric Quartet
with Carol Wincenc, flute
STEVE HACKET >>String Theory
RICARDO DAL FARRA >>Homotects (World Premiere, New Version)
INGRAM MARSHALL >>Fog Trope III
RAND STEIGER >>13 Loops (World Premiere)
MARK WINGATE >>Prophecy
ROGER REYNOLDS >>Arabesque’s Thread (NY Premiere)
Tickets: $15. Call 212.279.4200, or online at www.ticketcentral.org.

Saturday, October 13, 2001 at 8 pm
The Knitting Factory, 74 Leonard St.
Electro-Acoustic Composers Out Front!
Composer-performers exploring new frontiers in composition, improvisation, technology, and virtuosic.
Music by MARC KIMURA, MARTHA MOOKE, DANIEL TRUeman, and GOLAN LEVIN
Tickets $20. Call 212.219.3096, or online at www.virtuous.com.
October 24, 2001

Dear Golan,

Thanks for being with us for Orchestra Tech!

It was a challenging, exhilarating, (and yes, sometimes frustrating) week, but one that I hope you found rewarding.

I was truly impressed with the range of music, talent and goodwill present at the Orchestra Tech concerts and proceedings. I especially want to thank you for your presence and contribution which helped to make the events such a success.

For your information, we have enclosed an Orchestra Tech contact sheet and a short survey. We would appreciate it if you could find the time to complete the survey to help us with our future planning for the Orchestra Tech initiative.

Also enclosed is a recording of your performance. The performance has, with your permission, been cleared for possible noncommercial public radio broadcast and for A-V streaming via American Music Center’s www.newmusicbox.org site.

Again, many thanks for being a part of the project.

With all best wishes,

Michael Geller
Executive Director
Dialogue with Expanded Images

2001.10.26 FRI. - 2001.11.4 SUN. 10:00 AM - 6:00 PM

ソフトビアジャパンセンター
（岐阜県大垣市）JR大垣駅よりバスで10分 [入場無料]

ガイドライン

接待作家：ケイシー・リース / ティファニー・ホルムズ / ジム・キャンベル / カミーユ・アッテ・パック / アーノ・ヤール /
ヴォルフガング・ミュンヒ・ダ・カール/ ジェイ・リーン・ビル・キース / アルス・エレクトロニカ・センター 未来研究所 /
児玉幸子＋竹野美奈子 / ゴーラン・レビン / 岩田洋夫

Arista: Casey Reas / Tiffany Holmes / Jim Campbell / Camille Utterback / Arnon Yaar /
Wolfgang Muench + Kiyoshi Furukawa / Jay Lee + Bill Keays / Ars Electronica Center /
Sachiko Kodama + Minako Takeno / Golan Levin / Hiroo Iwata

10.27 SAT. アーティスト・トーク / パフォーマンス「風来居」 カール・ストーン
10.28 SUN. 国際シンポジウム / パフォーマンス「イメージ・シンフォニー」 ゴーラン・レビン
11.3 SAT. - 4 SUN. 作品体験イベント「浮遊する視線」 岩田洋夫

主催：世界メディア文化フォーラム実行委員会（岐阜県・大垣市）
企画・運営：IAMAS（情報科学技術大学院大学・情報科学技術研究所）

協賛：NTT インターコミュニケーション・センター(ICC)、ソニー株式会社、株式会社ワコム、エプソン株式会社
問い合わせ：IAMAS事務局 〒503-0014 岐阜県大垣市清家町3-3-95 TEL 0584-75-6400 FAX 0584-75-6637
E-mail: info@iamas.ac.jp URL: http://www.iamas.ac.jp/interaction/
ソフトビアジャンプセンター（ソピアホール）

2001年10月26日（金）～11月4日（日）
10:00 AM - 6:00 PM  [入場無料]

岩田洋夫

浮遊する船艙上のビデオ：カメラから見える
下洋の風景のなかで、自分自身の存在ま
でたくさんの瞬間を捉える動画と映像
メソッド、その奇妙な存在感の不思議。
Golan Levin

Golan Levin is an artist and composer interested in developing artifacts and experiences which exploit new modes of audiovisual expression. His work has focused on the design of systems for the creation and performance of simultaneous image and sound, as part of a more general examination of communications protocols for individual engagement and social dialogue. Levin is the recipient of an Award of Distinction in the Prix Ars Electronica 2000 for his AVES interactive software and its accompanying audiovisual performance. Scribbles, Levin received undergraduate and graduate degrees from the MIT Media Laboratory, where he studied with John Maeda in the Aesthetics and Computation Group. Prior to this, he worked as a research scientist and interaction designer at Interval Research Corporation for four years. He currently resides in New York City.

Audiovisual Environment Suite

The Audiovisual Environment Suite (AVES) is a set of five interactive systems which allow people to create and perform abstract animation and synthetic sound in real time. Each environment is an experimental attempt to design an interface which is supple and easy to learn, yet can also yield interesting, infinitely variable and personally expressive performances in both the visual and aural domains. Indeed, these systems permit their interactants to engage in a flow state of pure experience.

The AVES systems are built around the metaphor of an “inhabitable” and dynamic audiovisual “habitat,” which is freely deposited and controlled by the user’s gestures. Each instrument situates this substance in a context whose free-form structure inherits from the visual language of abstract painting and animation. The use of low-level synthesis techniques permits the sound and image to be tightly linked, commensurately malleable, and deeply plastic.

The AVES systems inhabit a domain at the juncture of art, design, and the engineering of tools and instruments. As artworks, they extend an established Twentieth century tradition in which artworks are themselves generative systems for other media. As a set of tools, the AVES work represents a vision for creative endeavor on the computer, in which uniquely ephemeral dynamic media blossom from a close collaboration between a system’s user and designer.


DEBUG: Es gibt wenige Beispiele für das Verschmelzen von Film und Narrative-Techniken. Was macht es so schwierig, einen neuen narrativen Rahmen auf der Basis eines Films für das Internet zu entwickeln?


DEBUG: Wie siehst du die Zukunft von bewegten Bildern? Wie werden die Gesellschaft erobert?

Golan Levin: Bewegte Bilder haben uns längst erobert. Schon letztes Jahrhundert. Dieses Jahrhundert ist die Auflage, sie zurückzuerobern.


Text von: jan rikus hillmann | aeonflux@debug-digital.de
aus: DB 56
As technology plays greater role in lives, artists incorporate it into works
By ANICK JESDANUN
Associated Press Writer
December 10, 2001

NEW YORK

The cell phones wouldn't stop ringing during Golan Levin's concert. An unnerving annoyance?

Not at all. The cell phones WERE the concert.

On the Austrian stage, the New York artist's computers were calling 200 phones in the audience, orchestrating an experimental cell-phone symphony out of the rings, beeps and musical medleys.

As technology plays a greater role in our lives, it's only natural that artists would incorporate it into their work. And this is no fringe phenomenon -- digital art is gaining acceptance in the mainstream art world, appearing in at least two major museum exhibits this year.

"Like the pencil or the video camera or whatever, artists have always been inclined to use tools that were available," said Benjamin Weil, curator of media arts at the San Francisco Museum of Modern Art.

For Levin, 29, the Sept. 2 concert in Linz, Austria, was a chance to explore "the love-hate relationship with the cell phone." Instead of asking concertgoers to turn phones off, he asked them to leave them on and to give him their numbers.

Levin works with software the way traditional artists work with canvases, clay or pianos.

"The computer is just as much a medium of art," he said.

Last month, Bitforms opened in New York as a small gallery devoted entirely to digital art.

On display is Levin's latest project: using computer algorithms to generate characters for languages that imaginary civilizations might have. He has a Web site from which fonts of the characters can be downloaded.

Another artist created a computer program that randomly generated lines and curves of various colors. The computer took 30 snapshots of the images, and prints of three were on display.

The gallery also features digital works on traditional medium, such as computer-generated drawings printed on paper and painted over using watercolors. One piece is "digitally inspired" -- it's a traditional painting of a computer icon enlarged to the point that it's unrecognizable.
Steve Sacks, the gallery's owner, says he created it to highlight the best of the digital works and to show the ranges of possibilities.

"It's evolved to the point where artists are getting better at taking advantage of the tools and making better art," Sacks said. "We've reached the level of seeing more museum-quality work."

Still, few digital artists make a living solely off their work. Many also teach or work in graphic design and rely heavily on grants.

Levin's cell-phone project took a year to complete -- nearly 10 months of it raising money and arranging donations of phone services. Levin notes the medium tends to be more expensive than others, requiring equipment and people with special skills.

Digital art remains experimental in many ways.

It is taught in art schools, "but it's so new that no one really has a firm grasp of what it actually is," said Mark Amerika, a digital artist and professor at the University of Colorado.

"It's a very exciting phase," said Barbara London, a curator at the Museum of Modern Art in New York. "Because it's less (defined), artists can take risks and learn what didn't work."

Artists have been exploring digital art since the 1960s, but only in the past few years has it become widely practical because of better technology and prices, said Peter Lunenfeld, a professor of media design at the Art Center College of Design in Pasadena, Calif.

The movement got a boost this year with the "BitStreams" exhibit at the Whitney Museum of American Art in New York and "010101" at the San Francisco Museum of Modern Art.

Projects commissioned or displayed include "e-poltergeist," which generates random sound bites and launches Web pages as if a ghost were in the machine, and "ecosystm," which uses programming code to translate real-time financial data into images of birds and trees.

Elsewhere, a digital music project called Cathedral ran online and offline Nov. 30-Dec. 2. One aspect involved a car in Poland hooked to the satellite-based global positioning system. As the artist drove the car, based on input from the online audience, various sounds were triggered.

Earlier this year, New York music composer David Anderson created an "Internet opera," using a computer to stretch tones and create musical scales that an advanced civilization might have.

Some of the more cutting-edge projects give performers greater creative control, said Joel Chadabe of the Electronic Music Foundation.

Traditionally, musicians perform what composers intend. With computers, Chadabe said, it's possible to completely change the direction of a multi-instrument piece based on a performer's improvisation.

Not satisfied with the paintbrush, British artist Tom Kemp turned to TealPaint software for creating images on the small display of Palm handhelds. He also moves postcards and other images around while they are scanned into a computer to create new works.
By tweaking what software manufacturers had in mind, he said, "I end up with a brush which could not exist in the real world. That's what I like about digital work: the ability to create tools, hence work, which would be otherwise unimaginable."

Some projects sound more like pranks, though. One, called "Paintball," shot a brightly colored paintball onto a white screen every time someone called a phone number to complain that it's not art. A Web site tracked the wall over a week.

Digital art comes with its challenges.

Unlike film that is developed or clay that dries, a digital work is never finished. Artists also need to keep up with the latest software releases and know that their work may be technologically obsolete in a few years.

And because it's easy to distribute some types digital art, such as Internet-based projects called Net.Art, plenty of bad art gets mixed with the good.

More important for museum curators and collectors seeking the avant-garde, digital art moves away from object-oriented traditions found in painting and sculpting.

How do you buy, store and display digital art? If you buy the entire computer system used, you then have to worry about technical support.

Sacks, the owner of the digital-arts gallery, is undeterred. The 21 pieces on display through Jan. 12 carry price tags ranging from $1,600 to $9,750.

He hasn't sold any yet and is directing efforts at convincing collectors that digital art can be bought and sold as a package -- computer, software, wiring and all.