



Piksel09 :: f[re](e){op}[en]able

Welcome to Piksel09 - the 7th annual Piksel festival!

This years theme - f[re](e){op}[en]able - is a play on the words free, open and able. This is our way of celebrating the 7th festival with a meta theme which in a poetic way express the fundamental topics that have been the main focus of Piksel from the start - artistic practice built on technological freedom!

The development, and therefore use, of digital technology today is mainly controlled by multinational corporations. Despite the prospects of technology expanding the means of artistic expression, the commercial demands of the software industries severely limit them instead. Piksel is focusing on the open source movement as a strategy for regaining artistic control of the technology, but also a means to bring attention to the close connections between art, politics, technology and economy.

Piksel is an international event for artists and developers working with free/libre and open source technologies within the arts. Part workshop, part festival, it is organised in Bergen, Norway, and involves participants from more than a dozen countries exchanging ideas, coding, presenting art and software projects, doing workshops, performances and discussions on the aesthetics and politics of free technology and art.

Piksel09 is done in collaboration with Gallery 3,14 and Lydgalleriet hosting this years exhibitions, Bergen Kunstmuseum hosting the presentations and panels, and Bergen Kunsthall/Landmark hosting the evening events.

Piksel - Free as in Art!

Presentations

- 4 The ToonLoop Live Stop Motion Software
- 5 Virtual Entity
- 6 REBUNTU
- 7 The Art of Seduction OR Practical Jedi Mindtricks OR Escaping the Matrix
- 9 Microcodes presentation
- 13 Re-ware
- 14 Milkymist, an open hardware VJ platform
- 15 Noise & Capitalism
- 17 Open Source Software Tools for creativity
- 18 Pure Data Rhythm & Bass Machine
- 19 From here to eternity with FOSS
- 20 Gate peepin'
- 22 OHANDA – Open Hardware and Design Alliance
- 23 Flock
- 24 Piksel Plenum – Open Artistic Production
- 25 oneliner: presentation
- 26 Respirator presentation
- 27 Py-Cessing
- 28 APODIO
- 33 Multi-Touch 360

Live events

- 34 Action Potential
- 36 KUNST UND MUSIK MIT DEM TAGESLICHTPROJEKTOR
- 37 DC12V ::Teatrino Elettrico::
- 38 Noise invaders
- 40 Paper Cut Tales
- 41 Mouth(s)' lecture(s)
- 42 Colour Projections
- 44 Mattin
- 45 d.R.e.G.S. – Late Night Lobby Nuizzzzance
- 46 noish VS automata
- 47 Respirator
- 48 He boxed regularly and was strong and very brave and always a perfect gentleman
- 49 MSST
- 50 Psychoid
- 52 Performance for circuit bent toys and meat controllers
- 53 The DIY drone synthesizer
- 54 Souffles I
- 55 [i/o]
- 56 THENOISER VS ZERO POINT ENERGY



All material in this catalog is licensed under a Creative Commons BY-SA license, unless the individual authors have specified otherwise.

Exhibition

- 57 Microcodes
- 58 Battle
- 59 capacitive body
- 60 TL-Display
- 62 Reality Checking Device
- 63 void extension
- 64 Unanchored
- 65 Health & Safety Violation #14
- 66 oneliner
- 69 intrinsic
- 70 The serv seq
- 71 Musica Vista 1.0
- 73 Taiknam hat
- 74 augen-auf-schlag
- 75 Generative Audio Prototypes
- 77 Tribo Pænix
- 78 Flick Flock

Workshops

- 79 W.O.M. – Workshop Osc Machine
- 81 LP Atari Punk Console
- 83 Qeve – Free your visuals
- 84 Hackteria | bioelectronix for artists
- 86 Navalha
- 87 CHAOSLAB
- 91 ToonLoop Live Animation Workshop
- 92 Enhancing Pure Data Interactivity with Computer Vision (Open CV)
- 93 Building simple analogue light-controlled theremins
- 94 VGA Signals
- 95 Minia, Open hardware sensor board

-
- 96 Software of the year
 - 97 Participants
 - 98 Crew
 - 99 Partners

The ToonLoop Live Stop Motion Software

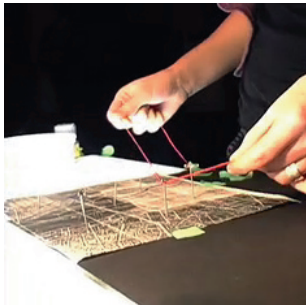
Alexandre Quessy (CA)

ToonLoop is a set of software tools for live stop motion animation. It is intended to help teaching new medias to children and to give a professional tool for movie creators. The performer can use a MIDI pedal to add frames to a constantly looping animation. ToonLoop is an idea and work of Alexandre Quessy with the help of Tristan Matthews. It is similar to the work of Pierre Hébert and Norman McLaren from the NFB of Canada who draw on film in live performances.

<http://www.toonloop.com> and <http://code.google.com/p/toonloop> for up-to-date informations.

Our current technologies are : Python, SDL (pygame), OpenGL and GLSL shaders. We use the JPEG and Motion-JPEG formats with the Quicktime component, or AVI. Operating systems : GNU/Linux and Mac OS X for the Python current version. There is also a Java/Processing version for Mac OS X and Windows. We use the Twisted and Nevow asynchronous networking Python packages. It supports the onionskin and chromakey effects. It can be controlled using either the keyboard or MIDI devices. It can be rendered full screen. Video clips can be exported as movie files.

<http://toonloop.googlecode.com/svn/trunk/py/INSTALL.txt> for installation instructions for GNU/Linux.



Alexandre Quessy is an audio-visual artist and developer from Montréal, Québec, Canada. He currently works at SAT as a programmer and is a graduate student in Communication at UQAM.

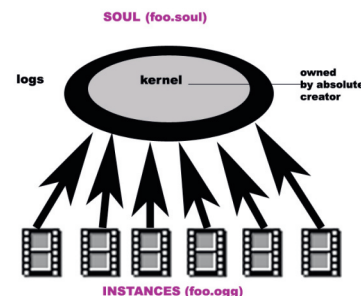
Virtual Entity

Eleonora Oreggia (IT)

Virtual Entity is a philosophical research starting from the assumption that the concepts of authenticity, ownership, uniqueness and seriality are, within the digital domain, no longer valid whereas they are not redefined.

The practical aspect of this research is a new software being specifically developed to release, license, and catalogue digital files. This system, transforming the traditional approach towards metadata, is based on the idea that any file is an independent creation living its own life and experiencing various levels of transformation and progressive generation (of meaning, shape, and entities) in the course of its virtual existence. This way digital resources, interpreted as cultural units, are considered the main actors of the web.

Virtual Entity constructs in parallel a theoretical mythical world and its functional technical counterpart; it is, in this sense, a physical and metaphysical software.



Eleonora Oreggia is a media artist born in Milan and based in Europe. She received a Master Degree with honor in Philosophy from the University of Bologna (DAMS), with a thesis in Semiotics of Art. She worked few years as editor and researcher at Netherlands Institute for Media Art (NIMK) in Amsterdam, and she has been Researcher in Design at Jan van Eyck Academie in Maastricht in 2008/2009. Her work, consisting of audiovisual pieces, software art, interactive installations and real-time performances, has been shown in many institutes, galleries and festivals over Europe, Asia and South America - and over the Net. "My work, mixing different practices, talks about social and individual identities, and their interfacing with reality and imagination". She is currently PhD candidate at Goldsmiths, University of London, department of Cultural Studies.



REBUNTU : Linux that kills itself and can kill You

Danja Vasiliev (RU)

re:buntu is an illustration of a system making decisions by itself, committing to self-maintenance and self-destruction. The system is continuously reloading a copy of itself within itself until the memory runs out. When no free memory is left the system needs to decide which one of the copies of itself it shall kill in order to reproduce itself again, again and again...

re:buntu is an OS intervention.

The kernel picks out a program process with the lowest "pid" and highest "oom_score" run by "non-root" user and terminates it to free up some memory to run new, more demanding program.

It did not take long and suddenly Linux (yes, the kernel) is everywhere around us. On our daily routes when we listen to the GPS navigation system, in our telephone conversations, in our precious on-line half-lives. It controls motors of big machines and renders augmented realities. Many of us use Linux kernel at home - the degree of some ultimate intimacy.

But are we actually in charge of the situation, in control of the way Linux kernel manipulates our data and deals with the programs we run? It would seem logical that the User must have the ultimate control over the system she interacts with, - remember the movie Tron?

- "He's not any kind of program, Sark. He's a User!" Yes, the User is the intelligence and thus shall be placed above any software, even above the Linux kernel.

However, the situation is not in our favor. For some numerous reasons the developers of Linux kernel decided to have something called "OOM-Killer" lurking around computer memory and deciding on its own when to crash User programs - all in order to keep System routines running. Simply put, when you open something huge on your computer and that something huge suddenly crashes - it is very likely that Linux kernel decided that you shouldn't open such huge files. Okay, not a big deal, can reboot and try again, but what about all those unmanned machines and devices working under the control of the very similar Linux kernel? It seems like we are getting closer and closer to the moment of happy singularity when our tools will not ask us anymore questions and will shut-off any root access...

Danja Vasiliev

Working with network technologies and Open Source software. Using computer as a raw model/base for experiments and trying to merge together the digital and meat spaces of our existence. Critical to pervasive cyborgination, while working with the same technologies - he is looking for balance between the two worlds.

<http://k0a1a.net/rebuntu>

The Art of Seduction OR Practical Jedi Mindtricks OR Escaping the Matrix

Richard Spindler (AU)

I am Richard Spindler, a Diplom-Ingenieur in the field of Computer Science from Austria. I have been at the Piksel Festival for several years now, presenting my Open Source Software Open Movie Editor, and also the underlying Gmerlin Framework, and the surrounding Linux Video Developers community.

I am an Open Source Developer and Advocate, this year I was speaking about Open Source and Free Software Video Applications at the Linux Audio Conference in Parma for example, and I am also interested in Open Street Map, giving talks about it on the yearly LinuxDay Event in Dornbirn, Austria.

As a Software Developer I was always interested in the concepts and foundations of user Interface Design, and furthermore the concept of Man-Machine Communication, which led me to become interested in the concept of communication between humans themselves.

Therefore, this year I would like to speak not about Software as in Computer Programs, but rather about a little research project of mine, that could probably be called Brainware. Most of my work was heavily inspired by the experiences and ideas that I carried home from past Piksel Festivals.

What I will talk about are practical Experiments that I did during this year, in respect to programming brains, both your own brain and other peoples brain, I will show movie clips, give some interesting ideas on how to understand them, and about how to apply this knowledge to ones own surroundings. I will offer advice and tricks how to tune out of the world that surrounds us and how to trick oneself into relying onto your natural instincts in a controlled and technocratic world. I will talk about how to cultivate habits that make an individual stronger to resist harmful influences from advertising and civilizational delusions and distractions. Also I will present methods and advice about how to create systems and structures to provide oneself with positive influence, as well as how to recognize and use existing infrastructure in your immediate surroundings for positive inspiration and support. I will also give a short intro into fnoords, what they are, and how to recognize them.

Additionally I will offer a diverse selection of internet links with further reading material to offer additional information for those that are more interested and to provide vast resources and inspiration for everyone to start working on their own experiments in this field. Also pointers to movies and books that might come in helpful as well will be provided.

Some of the stuff that I am presenting will be tough and difficult, shocking and disturbing, but from what I've seen over the last years, I believe that the Piksel audience is openminded enough to see through my petty metaphors and to understand what I will be explaining.

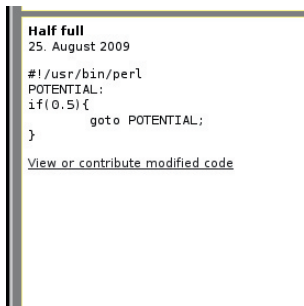
Also I believe that some of the material that I will be presenting might be familiar to many pikslers, as I've seen people also working with similar concepts in their pieces. My talk will be disillusioning, enlightening, surprising, confusing but entertaining and completely crazy. Have fun.



Microcodes presentation

Pall Thayer (IS)

Microcodes are very small code-based artworks. Each one is a fully contained work of art. The conceptual meaning of each piece is revealed through a combination of the title, the code and the results of running them on a computer. In emphasizing the conceptual, artistic potential of software code, these works make no attempt at being useful in the sense that tools may be useful. Their sole purpose is to evoke thoughtful contemplation and, in doing so, perhaps to enlighten the viewer. As works of art these are the creative work of Pall Thayer. As programs they may be copied, distributed, modified and used under the terms of the GNU General Public License v.3 or (at your option) any later version. The work may be seen at <http://pallit.lhi.is/microcodes>
<http://www.this.is/pallit>



Pall Thayer is an Icelandic software and code artist. He has been involved with new media art for over 10 years, creating work that ranges from interactive, audio-visual installations and internet-based work to pure code being presented as artwork. His work has been exhibited widely at festivals and galleries around the world such as Transmediale (Berlin), FILE (Sao Paulo), Pace Digital Gallery (NYC) and Mejan Labs (Stockholm) to name but a few. Pall currently serves as director of Lorna, the Icelandic organization for electronic arts.

The Microcode Primer

A guide for non-coders towards a conceptual appreciation of code

based on codes at <http://pallit.lhi.is/microcodes>

Over the years, we've seen how the "private" languages of certain fields of practice within the arts have slowly permeated the more public language used to express and discuss various facets of these fields. For some odd reason it appears as though the practice of writing software code has been exempted from this. Perhaps because artists don't always make their code visible or because artwork based on software code hasn't been around long enough and gained enough attention.

My recent series of "Microcodes" are intended to be critically examined at the code level as well as at the level of the running process. The code informs the conceptual ideas behind each piece while the running process lends it a more "poetic air". Therefore, I am providing this short primer that is intended to give the viewer some insight into the meaning of written software code so that they can extract valuable information from it that will help them to better understand what each of the "Microcodes" is about.

This is not by any means intended as a tutorial on how to write software code, only as a guide on how to read software code in a meaningful way.

Let's begin by examining the shortest of the "Microcodes". "Sleep" is of course a direct reference to Andy Warhol's movie of the same title. Andy had intended to make the movie eight hours long to reflect the average time that people sleep at night.

```
#!/usr/bin/perl
sleep((8*60)*60);
```

The first line of this code is the same in all of the codes. It simply tells the computer what programming language this is written in. In the case of the "Microcodes", the language is Perl. The second line tells the program to sleep by using the built-in function "sleep". Most languages use functions the same way. The name of the function is written first and then "arguments" are supplied in parentheses after it. When you see something like this a google search for "Perl sleep" will give you many webpages full of information about the sleep function. Different functions accept different arguments. The sleep function accepts a number of seconds to sleep. In this case I could have simply written 28800 (the total number of seconds in eight hours) but I wanted to emphasize the fact that it is eight hours, therefore $8(\text{hours}) * 60(\text{minutes})$ gives us the number of minutes in eight hours and that number $* 60(\text{seconds})$ again gives us the total number of seconds in eight hours. Mathematical operations and other functions can be used as arguments for a function.

The important thing here is that the computer needs to keep track of the time during a "sleep" operation to know when to stop sleeping. So while the program sleeps, it's not doing just nothing. It is "actively" sleeping just as Andy's movie showed that when a person sleeps, he's not doing "nothing". He is in the active process of sleeping.

Now let's take a look at a slightly more complex "Microcode". "Active monochrome" is the low-tech, digital approach to monochromatic painting. Whether we interpret monochrome painting as stillness, pause or violent upheaval against the multi-faceted norm in painting, the coded version provides the same sort of experience.

```
#!/usr/bin/perl
```

```
print "\e[?25l";
system 'clear';
while(1){
    $$SIG{INT}=sub{print "\e[?25h";exit;};
}
```

All of the “Microcodes” do what they do in a terminal window. This is a very basic computing environment. Instead of using a graphical user interface, everything is performed textually. Programs are run by typing their name and parameters are set by typing commands. A terminal window can always be cleared by running the “clear” program (typing “man clear” in a terminal window will provide you with information about the “clear” program). In this case this is done by using the “system” function to tell the computer to run a separate program. But even then a blinking cursor will be displayed. The way to get rid of this blinking cursor is to issue what's called a “terminal escape code” which consists of a cryptic sequence containing “\e[?” and some characters. Such sequences can be used to manipulate the color of the text or the background, change the terminal window's properties (i.e. Size or position) or several other properties. This program begins by printing such an escape sequence to the terminal to hide the cursor. Next it uses the “system” function to run the “clear” program that clears the terminal of any text. Following these actions it enters an “infinite loop” or a loop that has no defined end. “while” is what's known as a “conditional”. It means that whatever is inside the curly brackets that follow it is to be performed if certain conditions are found to be true. For instance, if we said, “while(1 == 0)”, whatever follows in curly brackets will never be executed because one can never equal zero. It's also important to understand, in this context, that the numerical value zero means “false” to a computer program. Likewise, any number that is not zero means true, even if it's a negative number. Therefore “while(1)” is a condition that will always be true and the program will repeat the process within the curly brackets over and over until the program is stopped by force. If this program stops running, it will show the normal terminal prompt immediately and therefore no longer display a monochrome. It has to work hard at maintaining the monochrome effect which in many ways is a rather violent disruption to the normal interface of the terminal window. It removes the terminal's functionality and forces the viewer to acknowledge it on a strictly aesthetic level. The code that is inside the while loop's curly brackets only gets performed if something halts the program (typing CTRL-c will stop it). Before exiting it will return the hidden cursor to its normal state.

The Perl programming language, like many others, is modular. It can be extended by using modules that provide functions that are not included in Perl's built in functions. Several of the “Microcodes” use a module called LWP::Simple which provides easy-to-use functions for retrieving information from the world wide web. Here is “Social spaces”:

```
#!/usr/bin/perl
use LWP::Simple;
$social_text = get('http://twitter.com/statuses/public_timeline.rss');
@social_space = $social_text =~ /(s)/g;
foreach(@social_space){
    print $_;
}
```

This code contains several methods that are common in computer programming. First off is the extension of the programming language's capabilities. “use LWP::Simple” tells the program to read a file in the computer that contains code that defines a number of functions. Just seeing that this program uses the LWP::Simple module, tells us that this program interacts with the web. There's no other reason to use this particular module. Google-ing “Perl LWP::Simple” will help you find more information about the LWP::Simple module than you could ever need. Another thing that this program contains that hasn't been introduced before is variables. Variables in computer programming are like adjustable symbols that can stand for anything the programmer wants them

to. Most often variables are likened to buckets that can contain things. In Perl, variables always take the form \$ and then the name of the variable. The name can be anything that the programmer chooses (within certain boundaries). I've chosen to use the name `social_text` so my variable is written `$social_text`. What it contains isn't immediately clear because it's not a fixed value or text. It's contents are set by using the "get" function which is one of the functions provided by LWP::Simple. As an argument, it takes the URL for a webpage. The URL I am using points to the syndication feed on twitter.com which contains the most recent entries by users of the site. So when I run the program, my variable (`$social_text`) will contain the text from that file. If you want to see what it looks like you can simply copy the URL and past it into your browser's address bar. So a variable's value can be set by some sort of an operation within the program (that's what makes them variable). For instance, if I have a line that looks like this:

```
$my_value = 1+1;
```

`$my_value` will not contain "1+1". It will contain "2" because mathematical operations are performed automatically (almost always).

Next, we have a very different sort of variable that is prepended with an @ symbol instead of \$. This is called an array. A variable can only hold a single value at a time. That value might be several pages of text but it's still only a single value. An array can hold several different values that we can call on separately by providing so-called indexes. I could set the values of an array like this:

```
@my_array = ('Pall', 'Sylvia', 'Olive', 'Markus', 'Patricia');
```

The different elements are numbered from 0 up. So if I follow with:

```
print $my_array[2]
```

It will print 'Olive' to the screen. Note that when calling on a single element of an array, its name is written with a \$ instead of @. This is exclusive to the Perl programming language.

The operation used to set the values for the `@social_space` array is somewhat complex and will take a little bit of explaining. The array's values are set to the outcome of the operation:

```
$social_text =~ /(\\s)/g;
```

The combination of equal sign and tilda (~) means that we want to match elements within the text using what's known as "regular expressions" or "regex". Regular expressions can be very complex and attempting a detailed explanation of them is way outside of the scope of this primer so I'll only explain what this particular regular expression does. In Perl the "conditions" of the regular expression are always between slashes (/ the regex goes here /). If the condition is located inside parentheses it means that you want to match that condition. The "g" that follows at the end means to keep matching throughout the whole text instead of stopping after the first match. "\\s" is a symbol for "whitespace" which is any space not containing a character (i.e. Spaces, tabs or linebreaks). So what this is matching is any space within the text and whenever a match is found, it is added as a new element to our array. In other words, yes... our array is filled with empty spaces.

The "foreach" comand is a handy way of looping through all of the elements of an array when you don't know how many there are. "foreach(@social_space)" will execute the code contained inside the following curly brackets for each element contained in the array. Within the curly brackets you will see the variable "\$_". This is unique to Perl and is one of a collection of so-called "special variables". Within a "foreach" loop, the \$_ variable will contain the contents of the current array

element. So, for the first loop `$_` will contain whatever `$social_space[0]` contains, the second loop will contain whatever `$social_space[1]` contains, etc. Until it reaches the end of the array.

So to sum up this rather complex explanation, what this program does is retrieve a collection of the most recent entries to the popular social website `twitter.com`, picks out all of the empty space within the text and prints it to the terminal window. Obviously, this questions our notions of "space" in the digital realm. It attempts to look at "online" space as if it were physical space occupied by letters where spaces between words represent empty space. In day to day conversation, we tend to use the word "space" to refer specifically to empty space (i.e. "I had to park across the street because there wasn't any space over here."), therefore it's logical(?) to consider "social space" in the digital sense as those empty spaces that occur within texts that appear on a social website.

I hope that this primer can help "non-coders" to see and understand some of the possible artistic concepts that can be embedded in this sort of artwork. In the case of "Microcodes" the concepts have been purposely veiled in the output of the codes, forcing the viewer to consider both the code as well as its output. So as to not overwhelm the viewer, the works are produced in short, succinct code. These are not examples of proper coding procedures. Quite the contrary. The emphasis is on their conceptual meaning at the expense of their functionality.

Pall Thayer
<http://this.is/pallit>
Reykjavik, Iceland
21. May, 2009

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Re-ware

Hans-Christoph Steiner (US), Marius Schebella (AU),
Chris 'the Widget' DiMauro (US)

We are surrounded by gadgets, from iPods to phones to PDAs. These little black boxes are in fact full-fledged computers and are eminently hackable. Yet all of these devices are built around one assumption: someone else creates, you consume. They are locked down and when using them, you are prevented from doing anything but consuming: consuming media and consuming minutes. On top of that, we discard these devices at an alarming rate. By creating new possibilities for these devices, this project will stem the tide of electronic waste, and in the process, make open, democratic, and sustainable gadgets. Thanks to many hardcore hackers, you can now run Linux-based open source software on them. Once you have Linux on them, you can transform them into true read/write digital tools that free you to create your own gadgets. This is where the Re-ware project comes in. For the past year, we have been releasing easy-to-use Linux distros for these devices, teaching workshops, making our own projects using mobile hardware from the trash. This activity lays the foundations for building a framework for people to turn their electronic trash into devices which encourage interactive engagement. Like C++ behind OpenFrameworks, Python and Pd run on these devices, and upon this a framework can be built for creating rich interactive applications.

re-ware.org

Hans-Christoph Steiner
<http://at.or.at/hans/>
Interactive Telecommunications Program, New York University
United States

Hans-Christoph Steiner spends his time designing interactive software with a focus on human perceptual capabilities, building networks with free software, and composing music with computers. With an emphasis on collaboration, he has worked in many forms, including responsive sound environments, free wireless networks that help build community, musical robots that listen, software environments that allow people to play with math, and a jet-powered fish that you can ride. To further his research, he teaches and works at various media art centers and organizes open, collaborative hacklabs and barcamp conferences. He is currently teaching courses in physical interaction design NYU's Interactive Telecommunications Program.

Marius Schebella
<http://www.schebella.com/>
Salzburg Research Forschungsgesellschaft mbH
Austria

Chris 'the Widget' DiMauro
United States



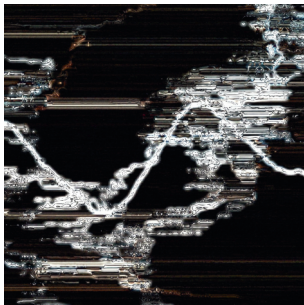
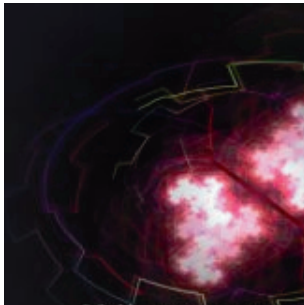
Milkymist, an open hardware VJ platform

Sébastien Bourdeauducq (FR)

The Milkymist' project develops a stand-alone device in a small form factor that is capable of rendering MilkDrop-esque visuals effects in real time, with a high level of interaction with many sensors and using live audio or video streams as a base. Open source components and design tools have been developed or used as much as possible. The design is also highly modular and documented, making the code easy to re-use in other open source system-on-chips.

The presentation focuses on the technical aspects of the project. Live demonstration included !

URL: <http://www.milkymist.org>



Sébastien Bourdeauducq
<http://lekernel.net>

Even though lekernel loves most areas of hacking, a large number of his projects have been around the latest electronic devices out there. He has joined the Prism54 developers' team in 2004, authoring drivers and firmware for different, newer chipsets for which no specification was available. He has also been involved in different industrial projects such as the development in 2006 of the 802.11 embedded driver stack for the Wi-Fi smart rabbit called Nabaztag. In 2008, he has co-founded the OpenPattern company, which produces FPGA-based open hardware devices, and he is mostly involved in PCB and FPGA developments. He is also an active member of the /tmp/lab geek collective, and one of the inventors of the "Consumer-B-Gone" ringtone. He is now spending a lot of time on the development of the Milkymist videosynth platform.

Noise & Capitalism

Mattin (ES)

'Noise' not only designates the no-man's-land between electro-acoustic investigation, free improvisation, avant-garde experiment, and sound art; more interestingly, it refers to anomalous zones of interference between genres: between post-punk and free jazz; between musique concrète and folk; between stochastic composition and art brut. – Ray Brassier

This book, Noise & Capitalism, is a tool for understanding the situation we are living through, the way our practices and our subjectivities are determined by capitalism. It explores contemporary alienation in order to discover whether the practices of improvisation and noise contain or can produce emancipatory moments and how these practices point towards social relations which can extend these moments.

If the conditions in which we produce our music affects our playing then let's try to feel through them, understand them as much as possible and, then, change these conditions.

If our senses are appropriated by capitalism and put to work in an 'attention economy', let's, then, reappropriate our senses, our capacity to feel, our receptive powers; let's start the war at the membrane!

Alienated language is noise, but noise contains possibilities that may, who knows, be more affective than discursive, more enigmatic than dogmatic.

Noise and improvisation are practices of risk, a 'going fragile'. Yet these risks imply a social responsibility that could take us beyond 'phoney freedom' and into unities of differing.

We find ourselves poised between vicariously florid academic criticism, overspecialised niche markets and basements full of anti-intellectual escapists. There is, afterall, 'a Franco, Churchill, Roosevelt, inside all of us...' yet this book is written neither by chiefs nor generals.

Here non-appointed practitioners, who are not yet disinterested, autotheorise ways of thinking through the contemporary conditions for making difficult music and opening up to the willfully perverse satisfactions of the auricular drives.

If you wish to receive a printed copy of this book please write or send an email with your postal address submitting your own feelings about or responses to the words 'Capitalism' and 'Noise'. Any letters, comments, criticisms, records, CDs or contributions in other media related to this book, will be gratefully received in exchange. Post: Arteleku, Kristobaldegi 14 (o nuevo P. Ainzieteta), Loiola Auzoa, 20014 Donostia - San Sebastián (Spain). Email: arteleku@gipuzkoa.net

This book can be downloaded as a PDF file:
http://www.arteleku.net/audiolab/noise_capitalism.pdf

Spanish and Basque editions of this book will be published in early 2010.

<http://mattin.org>



Anti-Copyright: Why Improvisation and Noise Run Against the Idea of Intellectual Property - Mattin

"Property is theft."

– Proudhon

"Intellectual property is shit."

– Billy Bao

INTRODUCTION

No other type of music-making contradicts itself through its recording like improvisation does. In this essay I intend to explain certain aspects inherent within the practice of improvisation and noise that counter the idea of intellectual property practically and conceptually. While many musicians would probably argue in favour of getting rid of any notion of authorship, and sharing their recordings, there is often a lack of discussion about this aspect of musical practice. Almost all the people that I know are downloading music, but people rarely talk of the consequences. Some people tell me it is very utopian or naïve to think that one can get rid of copyright and intellectual property, but to a certain extent it is already happening in practice. Most of the music that is heard in the world is likely to be from downloads using different peer to peer (P2P) networks such as Soulseek, Amule or Bittorrent, or one-click hosting pay websites such as Rapidshare. Because of its rigid and bureaucratic structure, the law is always left behind by the questions posed by new technologies. But, apparently, it is only a matter of time before the law catches up. Right now repressive measures aided by technologies of surveillance and control are already being developed without our consent by the most powerful governments under the pressure of corporations (ACTA being a good example).[1] Should we allow them to do this or should we start to develop our own platforms outside of the ideological framework that lets them behave this way? I will argue that the practice of improvisation in itself questions the foundations upon which intellectual property is based, such as: authorship, rights, restrictions, property, and the division between production and consumption. Improvisation and noise distribution, with their hardcore do it yourself (DIY) aesthetics, indicate alternatives to the mainstream means of production and distribution of music. Both practices are intertwined and share many things in common, but I am taking their obvious characteristics as a way of showing that within these types of music-making, there is already an existing critical attitude towards copyright that should be deepened and developed consciously.

Read full article

http://www.arteleku.net/audiolab/noise_capitalism.pdf
p. 167 - 191

Open Source Software Tools for creativity

Letizia Jaccheri (NO)

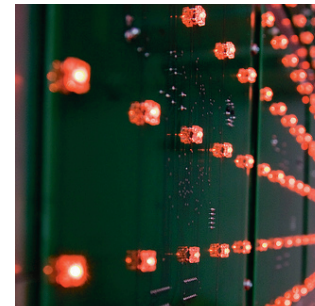
The ArTe contest encourages students of age 13 to 15 to work in meaningful cooperative projects with the goal of producing novel forms of new media art. The ArTe project is focused on the relationships between new media art and openness, organized around three topics: the relation between the author and the audience, the media of the artwork, and the tools used to realize the artwork. This presentation will focus on eight open source tools evaluated according to the ArTe goals.

What is new media art?

By new media art, we mean all forms of art that exploit information technology (IT). Examples of new media art are, but are not limited to, digital images, videos, music, games, digital stories and poems, as well as interactive installations.

Openness

The projects might address openness in different ways. This means either exploitation of one or more open source software tools, or re-use of artworks with open licenses such as Creative Commons when developing artwork of your own. By open source software tools we mean tools that are available for everybody and that can be not only downloaded and used but also inspected and modified.



Letizia Jaccheri is a Professor at the Department of Computer and Information Science at the Norwegian University of Science and Technology (NTNU) since 2002. In the last five years, she has been involved in the supervision of at least 10 PhD students. She has more than 15 years experience with research projects, both at National level (Italian and Norwegian) and international. She has been working with software research issues since 1988. She wrote her PhD in 1994. She is interested in software intensive processes with special focus on artistic ware intensive processes with special focus on artistic software and open source software. Principal contact for editorial correspondence.

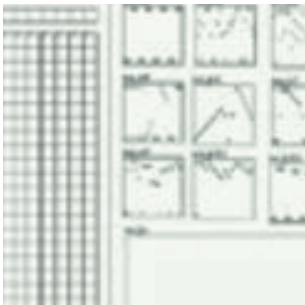
Pure Data Rhythm & Bass Machine

Carlos Tricas (ES)

The Pure Data Patch is an actualization of old rhythm machines, where the interface is inspired in step-by-step programming interface of first drum machines. It can be used for live playing or study of various PD capabilities. The most important development and implementation is the use of matrix's to load and save registers and states of patrons for live performance.

VIDEO: <http://www.youtube.com/watch?v=ks2y1UUedIQ>

Pure Data Rhythm & Bass Machine

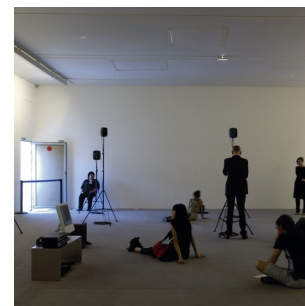
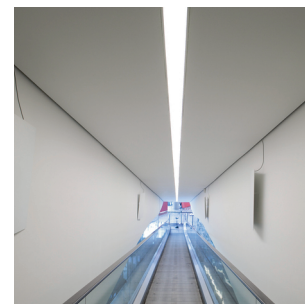


From here to eternity with FOSS

Tommi Keränen (FI)

This presentation explains the concept and implementation of the software solution used in a sound installation work installed permanently at Hijmans van den Bergh building, University of Utrecht. The installation is a generative sound work based on neural network controlled sound synthesis and processing where the composer (Florian Hecker) has educated the neural networks by example. The concept was developed by Tommi Keränen and Florian Hecker, and the software implementation was done by Tommi Keränen. The installation is created using free open source software (Ubuntu Linux, SuperCollider, and FANN library), and it has been running without interruptions since July 2006.

<http://florianhecker.blogspot.com/2006/07/show-release-hijmans-van-den-bergh.html>



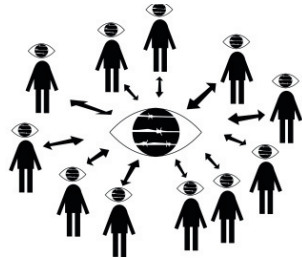
Tommi Keränen (b. 1974) has been active in the field of experimental music since the late 1990's. Over the last few years his primary interests have been harsh noise, improvisation and computer music. He performs solo as well in groups Testicle Hazard (with Lasse Marhaug) and Gentle Evil (with Janne Tuomi), and has collaborated with artists such as Incapacitants, Pain Jerk and Dror Feiler. In computer music his primary interests are sound synthesis methods and generative music. Without formal education neither on music or computer science, he often employs his background in theoretical physics and (psycho-)acoustics in the computer music works.

GATE PEEPIN'

Linda Hilfling (DK)

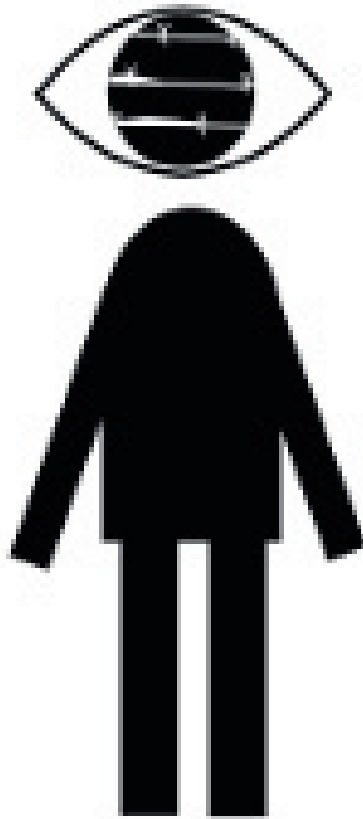
The Internet services of Web 2.0 are acting as community platforms for users to engage and participate in the creation and development of content and in this way new kinds of private-public spaces are appearing. Gate peepin' is an open source tool that alters the browsing experience of different Web 2.0-service platforms according to the Terms of Service regulating the services. The project is a Firefox extension that compares the Terms of Service document with the text content of a site and re-edits it by inserting phrases from the regulations into the content of the site.

URL: Download and install Gate peepin' from <http://www.gatepeepin.org>



Danish artist Linda Hilfling works with the premises of participation and public spaces within media structures, with a focus on means of control (codes, organisation and law) and their cultural impact. Her artistic practice takes the form of interventions reflecting upon or revealing hidden gaps in these structures. Linda has an MA in Networked Media from Piet Zwart Institute and a BA in architecture as well as a background in filmmaking. Her interest in those areas is founded in an attention to the structures they are part of and how practice is inscribed in but also re-forming these fields. This has lead her to interventions within existing media structures - recombining old and new, analogue and digital. Linda's works range from concepts for using ATM-machines or surveillance cameras as local-media platforms, to software interventions altering the user's browsing experience as for instance the "Misspelling Generator" - a Firefox extension that circumvents Google's self-censorship and rigid information structure or "Gatepeepin'" - a tool that allows its users a peep into the otherwise hidden layers of regulations governing the use of Web 2.0 service platforms. Linda is also director (together with Kristoffer Gansing) of the Art of the Overhead - a media archeological festival which pays tribute to the almost forgotten apparatus of the overhead projector.

**STOP SURFING THE WEB
WITH YOUR EYES CLOSED**



START GATE PEEPIN' NOW!

OHANDA - Open Hardware and Design Alliance

Jürgen Neumann (DE), Tuomo Tammenpää (FI),
Gisle Frøysland (NO), Bengt Sjöln (SE)



OHANDA (Open Hardware and Design Alliance) is an initiative to foster sustainable sharing of open hardware and design. It was started at the GOSH!-Grounding Open Source Hardware summit at the Banff Centre in July 2009 and one of the first goals of the project is to build a repository for open hardware design which includes a certification model and a registration service.

Direct adaptation of the freedoms from Free Software Definitions:

Freedom 0. The freedom to use the device for any purpose.

Freedom 1. The freedom to study how the device works and change it to make it to do what you wish. Access to the complete design is a precondition to this.

Freedom 2. The freedom to redistribute the device and/or design.

Freedom 3. The freedom to improve the device and/or design, and release your improvements (and modified versions in general) to the public, so that the whole community benefits. Access to the complete design is precondition to this.

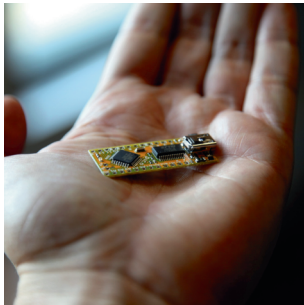
<http://www.gosh2009.ca/wiki>

<http://www.ohanda.org>

Gisle Frøysland

For more than a decade, Gisle Frøysland has been one of the key figures of the Norwegian electronic arts scene. He was a founding member of the Bergen Centre for Electronic Arts (BEK), initiator/maintainer of the FLOSS video-software MøB and initiator/main organizer of the Píksel festival in Bergen, Norway. Since the early 1980s he has been working as a musician, VJ and visual artist.

Web Sites: <http://www.gislefroysland.com> | <http://www.220hex.org> | <http://piksel.no>



Juergen Neumann started working with information technology in 1984, and since then has been looking for ways to deploy ICT in useful ways for organizations and society. As a consultant for ICT strategy and implementation, he has worked for major German and international companies, as well as on many non-profit projects.

Besides his professional engagement, in 2002 he co-founded <http://www.freifunk.net>, a non-profit campaign to spread knowledge and social networking about free and "open" networks – a campaign globally regarded as one of the most successful grassroots community projects in this field.

Then, frustrated from reflashing hundreds of wireless access points over the past years, in 2007 Neumann initiated the Open Hardware Initiative – an alliance of activists lobbying for open source hardware at the sidelines of both eastern and western hardware industries. In 2008, he co-organized the first Open Technology Summit in Taiwan.

Besides his job as CEO of a private consultancy company, his recent activities include digging deeper into the possibilities of manufacturing open source(d) chip designs and lobbying for new and more open licensing models for the radio spectrum.

Tuomo Tammenpää has a background in visual arts and design. He works as a media artist and designer in Finland, sharing his time between practice-based research and development, art productions and commercial design work at concept agency YATTA.

After fifteen years of work with interactive installations and screen-based design, he has focused his work on exploring the potential of physical and ubiquitous computing and tangible media in artistic and design practice. He has exhibited award-winning interactive media installations in Scandinavia, Europe, North America and Asia. Tammenpää has been involved in several European cultural networks, most lately Pixelache [9], an electronic arts and subcultures festival. He is a member of Grafia ry and SIGCHI/ACM. Besides his design work in YATTA, Tammenpää collaborates with English game designer Daniel Blackburn on electronically enhanced games and plays. They are interested in combining the social and physical aspects from plays and board games with the computational possibilities of microcontrolled electronics. In his artistic practice, Tammenpää is studying the use and misuse of everyday electronic devices in the context of the demystification and democratization of technology. Hacking, modifying, circuit-bending and open hardware ideology provide the working context, wherein he questions the relationship between the electronic objects, consumers as users, and the industry. Website:

<http://www.google.com/profiles/tuomo.tammenpaa>



Bengt Sjöln is an independent software and hardware designer/hacker/artist based in Stockholm, with roots in the home computer demo scene.

He is not part of one single group, but rather collaborates with several networks, including Teenage Engineering, a multi-disciplinary design studio based in Stockholm, and aether architecture in Budapest. Collaboration is often in the form of peer production, where the network, rather than the individual, is the author, and where not only is the network crossing discipline boundaries, but so is each individual participant, too, creating a sum larger than its parts through sharing and cross-breeding of ideas.

Sjöln is doing projects in the contexts of media art, science, sound, visuals, architecture and technology. He experiments with, among other things, programmatic generation of design, function, hardware and software. One example of this is customizing computer-controlled machines with parts manufactured by the same machines to enable rapid prototyping of electronics and mechanics from generated designs, bypassing the traditional processes and economics of prototype design and manufacturing.

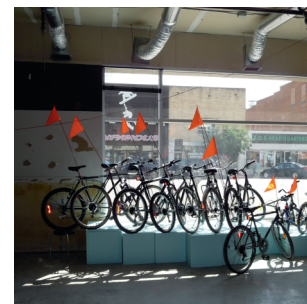
Sjöln's work has been exhibited internationally in venues like Synthetic Times Exhibition (Beijing), NTT ICC (Tokyo) and the Biennale of Architecture (Venice).

Flock

Kelly Jaclynn Andres (CA)

Flock is an experimental system modeled on the synchronization of human movement and flight patterns demonstrated in some species of birds, such as Canadian Geese or European Starlings. The work was created to be used by a group of participants on bicycles equipped with wearable cycling vests that display alternating light patterns for participants to follow in the context of a multi-player, synchronization game. New gestures are introduced to the participants through the system, challenging current intuitive patterns of interaction and introducing new ones.

URL: <http://www.kellyandres.com/flock.html>



Kelly Jaclynn Andres (CAN) is an interdisciplinary artist who applies the apparatus of technology, portability, and location, from bicycles to communication devices, to create situations that combine the absurd with sensory-based experiences. By incorporating banal objects along with electronic media often as interactions between species such as yeast, bacteria, poultry, plant, and human, Andres deploys simple systems, objects and performances that allow participants to explore and interact with the immediate environment.

Piksel Plenum #2 – Open Artistic Production

Angela Plohman(NL), Wendy Ann Mansille(DE),
Jordi Puig(DE), Julien Ottavi(FR),

Open Artistic Production: Blueprints, archiving and open source technology

In the Piksel Plenum panel, we invite you to participate in a lounge talk where we explore how processes of artistic production can be shared beyond pure archiving and documentation.

This talk is directly connected to Piksel's collaboration with Baltan Laboratories in Eindhoven (NL) which have been running from September til November 2009. Artists Wendy Ann Mansilla and Jordi Puig have partaken in workshops and carried out an artist residency at Baltan Labs. Here they have worked with developers from the Piksel network towards a commissioned work, Flick Flock, which is to be exhibited at Baltan Laboratories and this year's Piksel festival. An important criteria for this project is a reflection of how work processes can be opened and shared in art production. We ask Baltan Lab director, Angela Plohman and the two artists to share their experiences from the residency. We have also invited Julien Ottavi from art and technology network APO33, to share insights from his research and experience on open and free technologies and how such tools can stimulate to collective creative processes.



Angela Plohman <http://www.baltanlaboratories.org>
Angela Plohman has worked for over ten years in the field of art and technology. From 1998 - 2001, she was Program Officer at the Daniel Langlois Foundation for Art, Science and Technology, Montreal. In 2002, she moved to Amsterdam where she has worked as a freelance writer, project manager and researcher for a number of organizations including Hull Time Based Arts (UK), the Blender Foundation (NL), the Inter-Society for the Electronic Arts (ISEA), V2 (NL) and others. From 2005-2008, she was the Content Developer at LaborCulture.org, a partner initiative of the European Cultural Foundation. Recently, she worked as project manager and content developer of the new Van Abbemuseum website. She is currently the director of BALTAN Laboratories, a new art and technology laboratory in Eindhoven (NL).

Wendy Ann Mansilla (<http://www.q2s.ntnu.no/~wendyann>) received her M.Sc. in Digital Media at ISNM (International School of New Media) at the University of Luebeck, Germany. In 2005, her master's thesis on acousmetre in virtual environments was presented at the Europrix Toptalent Award, Vienna. She also participated in various conferences and workshops, gave presentations and published several scientific and research papers on digital media, virtual reality, video games, acousmetric cinema, among others. In 2007, she worked at Universitat Pompeu Fabra in the domain of mixed-reality environments. Along with her academic work, she conducts lectures on new media technologies and augmented reality, and produces and exhibits various interactive media and artworks. She is currently a Ph.D candidate with the Center of Quantifiable Quality of Service at Norwegian University of Science and Technology (NTNU).

Jordi Puig studied Electronic Art at Escola Superior de Disseny (ESDi) where he participated in the production of a number of artistic projects: an interactive installation for public spaces called IDADES; a network art and information visualization project called Nodemail; and interactive audiovisual robot in Connected Memories. He also produced and collaborated on various projects developed at Media Lab Madrid such as "MASK", "Palimpsesto", "Delicate Boundaries", and "Casas Tristes". Aside from developing art productions, he lectured at ESDi on audio visual installations, computer vision, and fusing interactivity in art and technology. He co-founded a creative advertising company called Cuatic, specialized in the development of physical interaction where he acted as a director of new media technologies for two years. Most of the projects that he developed at Cuatic won considerable awards at digital media and advertising festivals. He also worked at Ars Electronica Futurelab as a Creative Engineer. Aside from visual art, he is also interested in sound projects, for example he recently participated in a theatrical production called Verion. He is currently working as a researcher at the Norwegian University of Science and Technology (NTNU).

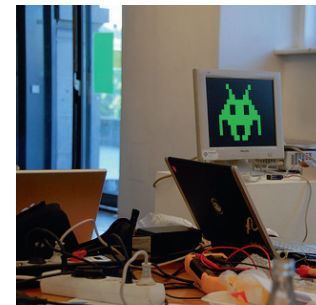
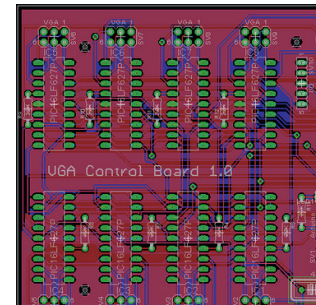
A mediactivist, artist-researcher, composer / musician, poet and tongues destroyer, experimental film maker and anarchist, founder and member of apo33, Julien Ottavi is involved in research and creative work, combining sound art, real-time video, new technologies and construction of electronic devices. An activist in the free software movement, he develops Apodio, a Gnu/Linux multimedia distribution. He is actually co-director of the Areal0Medialab in London and participates in, and produces, numerous events such as conventions of researchers, artists and activists, sound art or multimedia festivals, workshops on free software and DIY electronics, etc.

_oneliner: presentation

Arjan Scherpenisse (NL)

In this presentation Scherpenisse gives an introduction to the _oneliner project – a self-reflexive installation consisting of a long line of interconnected VGA monitors. In addition to share important insights from his research process and present the end result, he will discuss what role open source plays in this particular work and in work as an artist in general.

URL: http://www.scherpenisse.net/_oneliner/



Arjan Scherpenisse
<http://www.scherpenisse.net/>
 Netherlands

Amsterdam, NL-based artist, recently (July 2009) graduated from the Gerrit Rietveld Academy. Background in software development, artificial intelligence, new media research. Works with electronics, building interactive installations. Keywords: atmel, pic, diy, vga, i2c, bluetooth, python, c++. Also gives workshops on various electronics-related subjects. Mediamatic, Amsterdam, Kunstuniversitat Linz, SGMK Zurich, ISEA'08

Respirator

Bjørnar Habbestad (NO), Jeff Carey (US), Roar Sletteland (NO)

The RESPIRATOR is an electroacoustic performance environment based on custom hardware and software. The project includes the development of a custom built audio mixer, extensive use of HID sensors and controllers, novel synthesis and processing techniques as well as the development of a large scale SC environment for handling multi modal mapping of synthesis, contro data and signal processing.

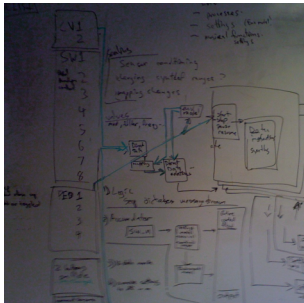


The project is a collaboration between Bjørnar Habbestad and Jeff Carey, Roar Sletteland (BEK) and Hans Wilmers (Notam). The mixer development is based on public domain designs, digital work is realized in SuperCollider and the sensor technology is based on STEIMS junctionboard and Notams SenseBow.

project blog
<http://www.respira2or.wordpress.com>
 project presentation
<http://www.bek.no/~bjornar/respirator.pdf>

PRESENTATION

The presentation will focus on the collaborative process and the fundamental ideas on instrument development and the need for modality of control data. Roar Sletteland will also present the development of the mixer.



CONCERT

A duo concert with USA/USB - aka Bjørnar Habbestad and Jeff Carey



Bjørnar Habbestad (1976) performs, improvises and creates music and sound art. Recent activity includes the norwegian premiere of Luigi Nono's "Das Atmende Klarsein" with the Danish Radio Chamber Choir, trio performances with free-jazz veterans Frode Gjerstad and Fred Lonberg-Holm, a Habbestad&Larsson sound installation for the art biennale ARTICLE and the performance of USA/USBS latest electroacoustic work at the dutch GAUDEAMUS festival.

Habbestad performs as a soloist, chamber and ensemble musician in Scandinavia, Europe, Asia and the US, covering musical grounds from noise to classical contemporary, electro-acoustic and free improvised music. Has collaborated with a range of composers in both an acoustic and electroacoustic domain. Since 2000, Habbestad has created telart - a cycle of sound-art works based on mobile telephone technology.

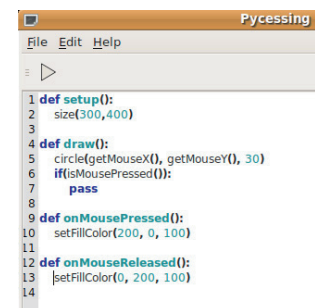
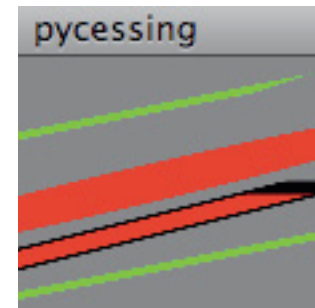
Jeff Carey
 American composer Jeff Carey's music is a blend of fixed-media composition and electro-instrumentalism -- the music of the visceral, ecstatic, and electric moment. His inventions of fluid sound are in an ever changing state: a mobile moves in the wind, changes shape, and is in constant renewal. Listeners are invited to explore an abstract sonic universe and engage in a world inspired by fictional cosmology, code breaking, and mathematical conundrums.

Py-Cessing

Brendan Howell (DE)

Python is a programming language that is powerful, fun and easy to learn. Py-cessing is a free software system that lets you create interactive audio-visual applications by programming them in Python. As the name implies, it is very much inspired by Processing. The current system provides easy routines for vector drawing/animation, image processing, audio playback, typography and live input from mouse, keyboard and joystick. Py-cessing runs on Linux, OSX, and MS-Windows. Everyone is invited, from beginners to Python masters!

PDF: <https://piksel.no/ocs/index.php/piksel/piksel09/paper/viewFile/179/33>



Brendan Howell (b. 1976, USA) is a media artist and an engineer. He has created various software works and interactive electronic inventions. He lives and works in Berlin, Germany and teaches at the Berliner Technische Kunsthochschule. He works as a developer for the Fritzting project at the FH Potsdam, University of Applied Sciences.

<http://wintermute.org/pycessing>

APODIO : A GNU/LINUX MULTIMEDIA DISTRIBUTION FOR EVERYONE!

Julien Ottavi (FR)

APODIO is a 6 years old Gnu/Linux distribution, dedicated to multimedia tools for audio/visual creation, to run your own radio, TV, multimedia tools for audio/visual creation, to run your own radio, TV, make your own film, animation, art installation, Live Coding and other A/V performance? .



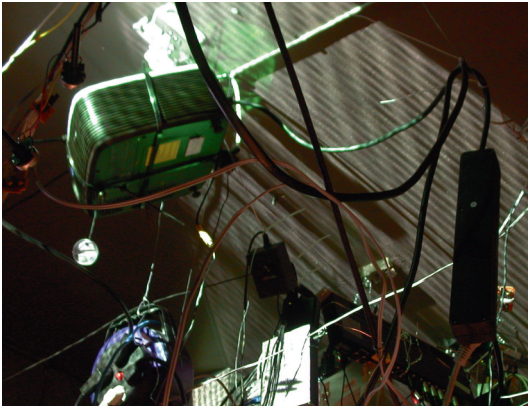
APODIO is a GNU/Linux platform containing audio, text-friendly, 3D, Streaming, graphic, Live Coding and video tools. It can be used as a liveDVD or be installed on a partition of your hard disk (on any PC 32bits to Mac Intel). APODIO is a GNU/GPL project, a part of the s to Mac Intel). APODIO is a GNU/GPL project, a part of the GNU/Linux Ubuntu family: APODIO uses a well known system operating the GNOME desktop. APODIO can easily and quickly be installed, as a whole, coherent, pre-set and instantly ready system on your computer, rather than a ramshackle combination of packages, And what you see as you use the LiveDVD is also what you get after the installation. APODIO respects the free/Libre software tradition because it fully supports Ubuntu without upsetting Ubuntu's convenient software packages and administrative scripts and because the liveDVD generative scripts are inistrative scripts and because the liveDVD generative scripts are always accessible : once you have installed APODIO, you can generate a new liveDVD out of your own installation, onto another DVD. APODIO is a free project under GNU/GPL licence, that fully complies with the GNU spirit and standards, as it encourages knowledge diffusion and knowledge sharing in the practice of multi-media tools, rather than just replacing an economic dependency with a technical one. APODIO is part of a long term undertaking ? not just a one off achievement.

APODIO 6 (new version 2009) have been developped with others groups/organisation (ping, labomedia, fabrique du Libre) in a very open collaborative process (hacking the same ISO master) . The presentation will summarize this collective process and explain how to develop apodio yourself in another situation (art project, radio, dedicated to a software, installation, studio...etc).
<http://www.apodio.org>

A mediactivist, artist-researcher, composer / musician, poet and tongues destroyer, experimental film maker and anarchist, founder and member of apo33, Julien Ottavi is involved in research and creative work, combining sound art, real-time video, new technologies and construction of electronic devices. An activist in the free software movement, he develops Apodio, a Gnu/Linux multimedia distribution. He is actually co-director of the Area10Medialab in London and participates in, and produces, numerous events such as conventions of researchers, artists and activists, sound art or multimedia festivals, workshops on free software and DIY electronics, etc.

The 'Free' and New Creative Practices : Open Source Modular Art-e-facts

version 0.1 (source : julien Ottavi – GNUart licence 2008)



The Intuitively Free and the Free as Practice

The free software movement came into existence with the GNU philosophy, originally developed by Richard Stallmann, through the GPL licence and the GNU operating system, and later, through GNU/LINUX. It was called 'Free' from the beginning, already an attitude, a choice for a certain kind of society and human behaviour, even before involving software and encoding; which should be considered in terms of continuous development, an in-progress operation, a sharing process. Gathered under this name the enduring questions of the sensible, intuitive and practical approaches to reality frequently arise. The 'free' movement cannot only be thought of as a practice limited to the realm of computing and art, since its origin was the result of a relation to production and to social trade. The concept of 'free' was not born out of the 70/80's, but had previously travelled through various periods, various processes, feeding on the all sorts of cultures. Tales, myths, the learning of techniques – all new knowledge enriched through the exchange and circulation of their content and of their practice in times and communities which shared the same interests and the same wish to understand. However, with the onset of capitalism the notion of property – the ownership of knowledge and of technology, dominates the processes by which knowledge is handled and obtained, especially with relation to the invention of copyrights. Yet, contrary to this the 'free' movement is an intuitive and deeply engrained attitude that has traversed the centuries. Rather than being detached from an extinct tradition, the 'free' movement continues to enlarge those ideas of its ancestors – which, recently replaced by a new language and new ideas, is now confronting a society where everything is bound by the concepts of property and the licensing of individual rights and commodity trading – objectifying every beginning of an idea or a product. GPL aims to overturn such a perspectives by upsetting that infamous right not to copy and instigating a new process of creation, of sharing and of the diffusion of source codes. But how do we move from the intuition of the 'free' to implementing its practice? By opening the faults that undermine a system, such as those cracks created beneath the capitalist structures, by practices contrary to those structures pretending to reach a new kind of stability in a society whose systems are crumbling, the 'free' movement is beginning to get a foot hold. Practising the 'free' signifies exiting the spheres of trade and fetishist capitalism; questioning the system (even its reality), organising its mutation, allowing others to develop their own ideas, projects and practices beyond the schemes imposed by the corporate machine. The practice of the 'free' can be found at all levels of human activity, be it immaterial (ideas, digital data, knowledge) or industrial, agricultural, manual (craft) or artistic. It becomes ever more obvious that praxis and production can exist under different guises, outside the notion of property. The 'free' unlocks the doors of fear or ignorance of dead end economics.

Free software and creation: beyond models, synergy and mutation

How do free software and open source influence each other and encourage creative processes and artistic practices? For a few years now, free software has participated, through creation-oriented tools, in artistic practices which could then be set free from the domination of large proprietary data



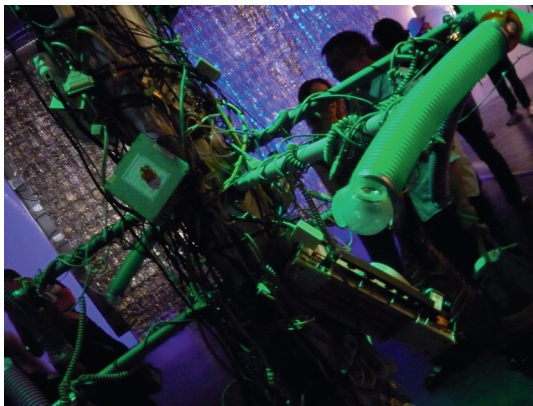
processing corporations. But we now discover a whole field of concepts through free software: the logic of source code, of sharing and open distribution, the freedom to modify sources, new modes of sharing, new ways to learn or to transform other people's work, working in a net or a collective, questioning copyrights and other property rights... The first effects of the 'free' on creation were felt against the notion of copyright in addition to the invention of new modes of distribution and sharing content or knowledge - especially digital content (audio, text, video, image). Questions concerning authorship had already been raised by previous avant-garde artistic groups, such as Dada, Lettrists, Situationists, as well as through unclaimed works and imaginary names, and so on. Yet the concept of authorship eventually came back with a vengeance; the individual's wish for recognition prevailed, programmed into us by the system and its formatting devices (family, school, prison, army, media...). The 'free' movement does not negate the existence of an author; on the contrary: the author introduces him/herself and allows us to develop his/her work beyond any denial to share and transform. What we call 'author-owner', as opposed to plain 'author', is controlled by jurisdiction, his/her ideas or products (books, cds, films, sculptures, clothes, designs...) would remain restricted or regulated. In many countries, all works of art or of the imagination, are « copyrighted » by default, i.e. their use is regulated and they are legally protected from free copying or transforming. The red tape of the bureaucrats has already decided what is best for us, we are entwined from the start; new concepts must necessarily belong to somebody; hence, they must be protected and licensed. At this stage, the notion of 'free' comes onto the scene, and licences such as 'GPL' become essential, allowing again the creative process to be developed outside of the rigid proprietary rules that binds it. The product is now set free from the world of traded commodities, ready to be shared, distributed or transformed. The author can now let his/her work proliferate and generate new processes. In a world of (almost) costless reproduction, the outmoded notion of the 'unique' has no reason to subsist; objects, concepts are no longer irreplaceable. We now live in a time of proliferation, of (digital or industrial) diffusion: when production can be multiplied, in different modes, from different points of view – not just the perspective of the originator, the « owner's ». The doors of fear and ignorance of one-way economics may be set ajar. Things can now be copied at will and immediately, as soon as they have been created or expressed. Reality now confronts the great propagating machine which can swallow anything and regurgitate so many of its replicas. More and more ants stream out of a digital planetary ant-hill. The author has now been 'liberated' through the opening of source codes and by the possibility of their modification. This trend can be observed in the artistic arena, thanks to many kinds of licences that have been chosen specifically for creation (creative commons, art libre, gnu-art etc.), and including works open to transformation, modification or re-appropriation... Where this opening had previously been intuitive, now it has become the conscious choice of the original author; there is no more talk of 'influence', 'quotes' or other ways to modify works (previously considered as 'assets', liable to be 'controlled'). When governed by these new licences, works become part of an on-going process, they are 'in progress', part of a multi-layered product. In this context, any member of the art community may consciously develop, disrupt, esthetize, redefine, divert a piece of work, a process, an idea, originated by others; questions can now be openly raised, the author/genius can be upstaged and the author/owner becomes rather run-of-the-mill. Everybody can now believe that that part of the gap that separated them from creativity has been filled and that art can penetrate our most trivial actions. More and more projects are using the concepts of open codes as a basis to push and encourage later transformations by individuals and groups. This means we may now imagine sharing what we are and what we mean without fear, fear of the other, of whoever might contradict us, question us or our project. We can now proliferate, become multi-cellular, imagine a manifold entity in each work, each process... History is no longer made of Great People only, but of multi-generational nuclei. We contemplate the renaissance of the collective author.

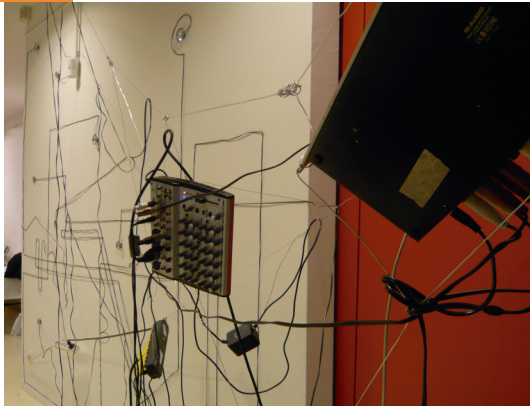


New practices influenced by the free movement :

For a few years, free software has been engaged in network practice. Through these first networks, source code has been able to circulate, it has been shared, modified, copied. GNU/LINUX first met recognition when it engaged in network technology. Free software could only be developed out of collaborative work, multi-authored projects, programming, corrections, beta-tests... From the outset the 'free' project participated in the Internet project and the practice of net-working. Without free software and the options of out-of-copyright licences, the notion of digital networks would have found itself limited to paying sites, or sites controlled by specialised companies. Whilst these companies are visibly present in the current system, they have to compete with more open techniques in addition to a mass of products stemming from the free movement, such as: software, texts, ideas, documentation, distribution, community, mutual aid, forums, modes of sharing etc. Many forms of creation implicating contemporary notions of networks, sharing and collaboration have been developed in and outside of the Internet. They intuitively borrowed concepts from the net and the digital, and from the resulting multiplication of forms and layers, the deregulation of time (stretched), the new ubiquity of our way of life. When the author is multiplied tenfold, thousandfold, when the machine (prosthesis of the human) becomes creator, autonomous and almost unbound: we could see a new society being born where new visions mingle and get bruised, pile up and explode, new hopes arise leading to mutations. Machinic mutations lead to chaos, the unknown, and unexpected behaviour. We are now running into darkness with fear as our only light, maybe toward our termination, like Icarus aiming for the sun, trying to disappear into the sun. However, what has this got to do with artistic practices? Maybe these practices only mirror our scopes, desires, phantasms? Maybe desire is necessary for our transformation, and we need to create, with machines, through networks, to participate collectively in an incommensurable and endless work of art, with networks acting as multipliers in a myriad of permutations. The continual production of a work, forever ongoing in its actualisation, can now be considered – with so many layers or versions, temporary endings and pauses, as a new process that explodes the myth of the completed work of art, of achievement – work that is perpetually in progress is no longer the dream that was previously imagined, rhetorically invoked : it can actually be realised. Relayed by many craftsmen, artists, technicians, who modify the very concept of work of art are no longer caught up in a 'movement', nor prisoners of the framework of Art History (as in Conceptual art), but free to shift within an ongoing practice, across diverse milieus and activities. New creative forms can connect, through networks, feeding and exchanging data from afar, ubiquitous. New forms of creation everywhere, reproducing like organic cells. Creators can now exchange in real time, exchange content and/or processes, as well as receive some form of instant feedback; they can imagine catching or transforming things differently, and hope that their own contribution will participate in a gigantic constellation. Any experiment has the potential to continue beyond any pre-defined framework, following directions never conceived previously. The notion of a categorised, quantified product could sometimes disappear, being replaced by undefined forms destined to evolve unexpectedly. The machine becomes poetical, almost human. Or we are becoming our own prosthesis, the cosmic phantasm that we exhale, integrating the here and there, the now and after. Free software and the practice of the 'free' in general liberates the desires that we have not yet expressed. Here they appear, suddenly realised, developed, diverted, recuperated and they communicate, but they dissolve as soon as they are born, and multiply to an infinity of forms, shared and open; their mere existence is questioned as much as their future and their definition. The pirate does not depend on the fake, the faux (remaining a prisoner of corporations or institutions), he explores the unknown, beyond definitions, beyond routine.

By developing from a free content, by sharing with multiple authors, we can reconsider the various co-actors or co-producers of a collective work, who are no more prisoners of a specific role: the





conceptor, the author, the technician, the producer, the distributor, the artist, the spectator, the participant... : we can now become all or anyone of these. And we know our (and their) work can now be distributed without any legal or technical constraints. With gratitude, we are lost. We have derailed the beaten path of easy categorisations and obvious histories, new modes are open : real-time, permanent diffusion, network distribution, countless copies, ubiquity on the net and the division of product-source. Even this text has not been written where you/they think: it's here and it's there, everywhere, you read it in one form but, maybe, it has already been copied and modified in ways you can't imagine. It is becoming something else, it has become already, it does not wait for us before it dissipates or proliferates into the universe.

Useful Links :

<http://www.apo33.org>
<http://www.noiser.org>
<http://poulpe.apo33.org>
<http://gnu.org>
<http://www.gnuart.org>
<http://artlibre.org/>
<http://creativecommons.org/>
<http://piksel.no/> <http://www.constantvzw.com>
<http://www.labomedia.net/>
<http://www.platoniq.net/>

...

Multi-Touch 360

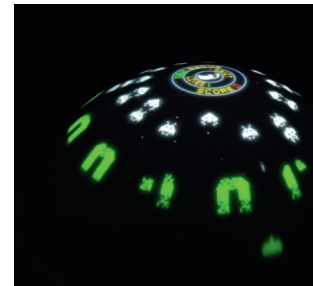
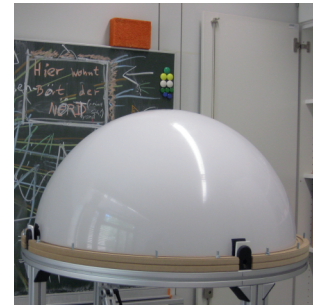
Thorsten Blum (DE)

Multi-Touch 360 is a DIY project by Thorsten Blum and Johann Korndörfer about building a hemispherical multi-touch display for interactive art installations.

The hardware for Multi-Touch 360 is up and running, and we can detect touches. We are currently working on the Python implementation for the visualization. We would be delighted to present it at Píksel and to give a talk about the project and the making-of.

<http://www.multitouch360.com/>

<http://www.youtube.com/watch?v=eO-tzlaQdEo>



Thorsten Blum
<http://www.toblux.com/>