EIT 2008

ART AND SOFTWARE
WILLAGE NR 13

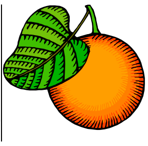
lux vitae

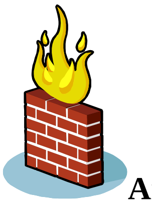
670501 678343 661061 647241 982702 689227

Table of contents

Acknowledgments

Special thanks to Professor Maria Letizia Jaccheri, Espen Gangvik and the learning assistants Anne Marie Haugen and Martin Almaas for guiding us, the Orange group, through our Experts in Team project. We are grateful for all the constructive criticism and feedback that has been given to us.

The  range Group



Anders: Background in cybernetics and currently pursuing his master degree while working full time as a software engineer.

Are: Background in drama, theatre and peace studies. He will finish his master project in May 2008. He is researching the possibility to apply drama and theatre in peace education and conflict-resolution.

Finn Kristian: Background in pedagogy and he is taking a master in counseling. It is a subject that focuses on helping people to work with themselves and together with other people.

Jacob: Background in physics and currently working on his master's thesis in physics at NTNU.

Marianne: Background from Art Academy. She have been working in different boards in organisations such as youth club in the cancer union where she have been working with very different types of people in different situations in the age from fifteen till thirty five. She had also work a lot in school related issues like exhibitions and discussion groups.

Phuong: Background in medicine. Before that she has been taken several classes in natural science subjects at Gløshaugen.

Introduction

Problem

How could a wall look like when the goal is interactivity between art/software and people in the room?

Delimitation

The wall: The wall project was initiated by Åsmund Gamlesæter with Espen Gangvik functioning as an aesthetics consultant. This project is in many ways representative of our expectations for the village of Art and Software. Given our very different backgrounds, this was the only project in which we felt that every one of us would be able to contribute something substantial to the end result. We felt that the wall was the only project that emphasized the artistic side of “Art and Software” strongly enough. We see it as a valuable tool with a lot of potential to achieve our goal of interactivity. Art in this context means the possibilities of the wall and how we can say something about interactivity by using this particular wall.

Interactivity: What do we mean by this term? In the fields of information science, communication, and industrial design there is debate over the meaning of interactivity. In the "contingency view" of interactivity, there are three levels: Non-interactive, when a message is not related to previous messages; Reactive, when a message is related only to one immediately previous message; and interactive, when a message is related to a number of previous messages and to the relationship between them (1).

In computer science interactive refers to software which accepts and responds to input from humans, for example commands or data. If the response is complex enough it is said that the system is conducting social interaction and some systems try to achieve this through the implementation of social interfaces (2). Interactivity also relates to new media art technologies where humans and animals are able to interact with and change the course of an artwork. New media art is a genre of art which is making possible new media technologies, including computer graphics, computer animation, the Internet, interactive technologies, robotics, and biotechnologies (3).

Denis McQuail mentions interactivity as one of the main characteristic of the new media. He quotes: (4)

“Interactivity: as indicated by the ratio of response or initiative on the part of the
user to the "offer" of the source/sender”.

Meeting room: Based on our focus on the term interactivity as mentioned in the previous section we have mainly concentrated on how we can introduce interactivity between the wall and the meeting room at IDI. It raised a question of what can possibly happen in a meeting room during a meeting. It will be our task to address this question and use the results to make the wall interact in a manner that is suitable for a meeting.

Background

In the mid-1960s many artists became interested in the creative activities at the intersection of art and technology. Previously, most people involved in the creation of computer art were engineers and scientists. The reasons were many: The only kind of computing resource available at the time was mainframe computers, available only to scientists at industrial and university scientific research labs. There existed no real interactive software, requiring the scientific programming expertise of the engineers and scientists. The early process of making art on the computer was therefore algorithmic in nature, and was not necessarily the way traditional artists were used to thinking. And last but not least, the art community was hesitant to regard the new art form as an acceptable art form. In spite of this, collaborations between artists and scientists began.

Early computer graphics was often in black and white, with linear formations and had little aesthetics interest. A. Michael Noll`*Gaussians Quadratic* from 1965 is a typical example in the early work from this field.

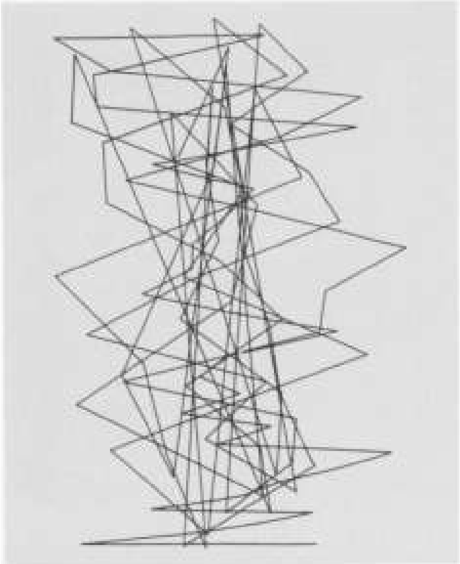


Figure 1 A. Michael Noll, *Gaussian Quadratic*, 1964, photograph, 11 × 8½ inches. © AMN 1965

In the 1970s, a significant shift takes place in computer art. Many artists, once hostile to computers, now enthusiastically start exploring the computer as an art making device. Over time, the artists fill the space originally taken by scientists and engineers. Much of this is due to technological innovations in the period, especially with the advent of real interactive software and graphical user interface. The availability of the microcomputer in the 1980s and the dawn of the internet in the 1990s gives a whole new generation intimate familiarity with the computer as a new medium of expressing oneself.

The field of computer art today is vast, and it is difficult to give a proper overview of the field here. Computers can support artists in the traditional fields of art, but the use of computers has also opened up completely new fields, such as interactive installations and fractal art. Digital technology can be involved in all stages in the process of artmaking from design to execution (5). Many artist and other creative professionals today use computers rather than traditional media. Using graphic art software can be more efficient than using traditional media by requiring less eye-hand coordination, requiring less visualization skills, and utilizing the computer's quicker (sometimes more accurate) automated rendering functions to create images.

Artists discovered possibilities of using the software in their artwork as quickly as the programs were launched into the marked. (6) Some of the more wonderful features of these computer techniques are the opportunities of the artists to use the interactive potential of the computer to respond in a “real-time”-manner. It means the processing happens as soon as the commands are given, and the effects are visualized right away. All this opens for a lot of experimentation. As one pioneer computer artist has said: “The computer is challenging the definition of an artist’s creativity –it may encompass the exploration of many more styles than ever before” (7).

There are some downsides however, to the use of computers in creating art. Advanced level computer styles, effects and editing methods may require a steeper a learning curve of computer technical skills than what was required to learn traditional hand rendering and visualization skills. The steeper technical learning curve may hinder creativity, despite the lure of enhanced creativity with reduced development time and reduced utilization of mental energy. (8) And for the aspiring digital artist, recognition of ones work in the wider public might prove difficult. Despite being around for half a century, computer art has yet to gain the acceptance and regard reserved for the classic forms of art. This is perhaps due to the erroneous impression that the computer does the work for the artist, and the fact that art in digital form is so easy to reproduce.

The idea process

When the group agreed on the project of constructing an artistic software process we were quite uncertain about what we wanted to get out of the project and what our possibilities were. Some of us had little or no experience in the area resembling our project. We started to talk about what our different strengths were and that we would like to have all of the group members contributing to the final result. The fields of art, software and counselling were a big part of the project. Art and software are obviously directly related to the wall installation, and counselling contributes to the analysis of the group’s dynamics. This meant that we would have to strive to include the medical, physics as well as the theatrical parts.

The first step of the project was to have a small brainstorming session. This is where everybody speaks out whatever springs to mind. No idea is to be considered to far-fetched in the first brainstorming session, and all of the group’s members have to contribute something. At most of the time the process will automatically lead to something more solid that the group can continue to work on. This will happen when people in the group are reacting to the ideas from other members. They will then start reshaping the ideas they like and slice away the parts they dislike. If this is not happening the group will have to engage in a more thorough discussion about the common goals of the group, and go through all the ideas again and carefully weigh the positive parts against the negative.

During the brainstorming session many ideas surfaced on how to position our working material in the room. Should we make a sculpture out of it, place them on the floor, on the wall, or perhaps on both? We contemplated placing the installation on the marked square in the room (the radioactive zone), or to have sensors monitoring the movements inside the square and producing some kind of related output to a wall-mounted screen. Some other ideas were to use mirrors to make the lights move more into the space.

Two other groups had chosen the same project as we had. This meant that all the material including the 108 circuit boards must be shared. Some of the groups had already based many of their ideas on different ways to position the plates, which would make it difficult to use the same material. The three groups summoned a meeting to discuss this issue. Either we had to evenly distribute the plates between the groups or we would have to find a common way of presenting them. Since there were relatively few plates and only 25 pixels on each of them, we decided that we would have to use all of them in the same installation. We finally agreed on using the wall. This was something all of us had many ideas on and which was also a part of what our contractors originally had in mind during their presentation. In the short amount of time available to us on this project, it is also a practical solution.

Ideas on how to present the wall

During our brainstorming session we came up with a vast number of ideas. The main mission of this section is to shed light on how our different competences have led to different ideas of thinking about the possibilities of the wall.

Abstract pattern: Building the wall and having a randomly abstract pattern displayed. The pattern will react either to sound or movement.

Living art: Does art die when it is forgotten? We had an idea on how to make an artwork live. If nobody cares for the work, it would die. This would involve the use of sensors in the room.

The bunny: This idea is to use the sound in the room. The room that the wall is to be built in is used for meetings, lectures and for lunch. The screen will react on the different frequencies and volumes in the conversations around the table. On the screen we will have a rabbit jumping, growing, sleeping or applauding according to what happens in the room.

Dropping dead: In the lecture about working in groups there was an example on people's reactions. If a man got a heart attack while walking in the street with many people present there would be a significantly lower chance that someone would stop and help him than if there were only one. We thought about how to use the screen to convey this idea.

Telling story: The idea was to having a story unfolding on the display. The story could be made interactive by having sensors that would enable people to influence sequence of events.

Sensors: The group discussed how we can make the lights respond to sensors in the room. A lot of our ideas already included the use of sensors both with sound and movement. Different kinds of sensors were discussed. This could be a way of bringing physics or cybernetics into the project.

Finally we came to the agreement of using the bunny to make the wall come to life. We named the project Lux Vitae, meaning light and life, to illustrate the connection between the bunny and its surroundings. This meant that the bunny will appear on the wall in different kinds of animations according to the changes registered by a sensor in the meeting room.

Sound in the room

Quite early in the decision process it became clear to us that we wanted our wall to interact with its surroundings. Given the time constraint on the project a relatively easy (and non-expensive) way to implement this is to use a microphone to record sound in the room. To process the sound we have chosen the FMOD audio package by Firelight Technologies, which is available free of charge for non-commercial applications.

The way our wall communicates with the room must obviously be dictated by the room’s primary function. We were informed that the room is mostly used for meetings. This sets some limits to our use of the wall. If it becomes too distracting it might be a very short-lived installation indeed. We wish to find some way of making the wall communicate in a manner that is beneficial to the meeting.

FMOD offers a wide range of functionality to help us realize our vision of the interacting wall. Using its built-in spectrum analyzer we can distinguish between different volumes, differences in pitch and between certain patterns in the sound. The time it has taken us to get everything up-and-running however, limits what we are able to achieve at the moment. But it is of some consolation to us the fact that it is not necessarily true that a higher level of sophistication produces better results.

Sound plays a very important part in our lives. We can recognize different types of sound such as speech, music, noise, thunder, noise etc. Sound is also one of the most important ways in which humans communicate. When we decided to use sound in the meeting room we were focusing on how the sound level have to say for the atmosphere in the room. Since we have two ears we can perceive other aspects of the sound than just its content. We have the ability to localize where the sound comes from. Our brain has a highly advanced centre for processing sound. One of its most important functions is the so called cocktail party effect. It enables us to follow a conversation in a room while there are several other conversations are going on at the same time.

Our current version of the wall need not be the end for future users; we aim at making our wall easily expandable for different purposes. Future users should be able to easily modify the walls set-up to accommodate their needs.

Building the wall

The three groups collaborating on the same project chose one to two representatives to work with the construction of the screen. The plan from our contractor was to have a metal frame on which we would mount the boards. This frame had not yet been made and to complete it within three weeks would be a difficult. The construction group decided to build one in wood, and agreed together with artist Espen Gangvik that a format of 1x3 (6x18 plates) would be visually best on the wall that had been chosen for this project. Later we had limited the size to 6x16 because of broken boards. To make the construction facilitate easy moving and replacing of parts, we based our work on an example that had been made one year ago. This was a small frame that consisted of six plates (2x3).

With a deadline only one week we faced a major problem of getting the frames built in time. After many rounds of negotiations we finally got the workshop at the faculty for architecture to do the job for us in just three days. We also got a carpenter to set up three mouldings to attach the frames on. Since the wall was made of gypsum and was not strong enough to carry the weight of 16 wooden constructions, the carpenter also hung up the wooden frames. All this was finished the morning on the last day of our time schedule and the construction group could then start to attach the circuit boards. They had to collaborate with the group in charge of the software so that everything was set up correctly with power cables and other wires. Since the boards were cut by hand and the frames were made in a hurry we got a problem with aligning them accurately. We got a few gaps between the plates but tried to make them as small as possible and distribute them evenly over the wall. On the final product this misalignment is hardly noticeable.

Technology

Hardware

The wall consists of 6 x 16 boards. Each board has 25 LED lights mounted onto the surface, emitting red light. The lights have 99 possible light intensities in theory, but less than that will actually be possible to perceive by the viewer. The upper range of light intensities is hard to distinguish.

The boards have their own microprocessor, power connection, and ethernet connection. Connection to the main controller device has to be established through a set of switches or hubs. In short, this is a massive parallel network of boards. The microprocessor on each board has to be pre-programmed with software for handling requests over the network, setting the wanted light intensities according to the received commands.

Software

Networking

UDP (User Datagram Protocol) is used for communication with the boards. UDP is a connectionless communication transport method that offers a limited amount of service when messages are exchanged over the Internet Protocol (IP). Unlike TCP, UDP does not acknowledge or guarantee delivery, nor does it provide sequencing of packages. This makes the protocol favoured for time-sensitive data such as real-time audio and video, where packets can be dumped as there is no time for retransmitting the data.
The boards have an IP stack embedded, and are programmed to respond to four possible commands:

PACKET_DATA_IMMEDIATE will update board memory buffer with new light intensities, and at the same time turn on the lights.
General format of the package is “$1L_1L_2L_3..L_{25}$” where L_n is the light intensities supplied as byte values.

PACKET_SETUP is used to set a new IP and MAC address for the board. Each board has been pre-programmed with a default MAC and IP address, so this should be used with caution.
General format of package is “$2ABCDM_1M_2M_3M_4M_5M_6$” where ABCD are byte values representing each part of an IP (A.B.C.D), and M_1 are byte values representing each part of a MAC address (M_1:M_2:M_3:M_4:M_5:M_6)

PACKET_DATA updates the board memory buffer with new light intensity values. This will not update the actual lights, only the internal buffer.
General format of package is “$3L_1L_2L_3..L_{25}$” where L_n is the light intensities supplied as byte values.

PACKET_UPDATE sets the light intensities according to the board memory buffer. General format of package is “4”.

By experimentation, it was discovered that the cards stops to respond after a while if no data is transmitted to them. One therefore has to “ping” the cards at given intervals to avoid this. It's reasonable to believe that this is related to a bug in the firmware source code. An application doing this has been made, and has to run in the background while using the wall.

Available software from Open Digital Canvas

Some time was spent looking at available source code and utilities found on the odc website (9) and from similar projects ran at Samfundet (10). The general impression was that a lot of the available source code seems to have been made in a hurry, with little or no commenting, lacking features.
There were some code using PERL or Python and a daemon using sockets for receiving data, but they were incomplete as well. We wanted to enable interaction with the help of sound, so the control system has to be able to hook up with a sound library in some way.

Available software from similar projects

Max/MSP (11) is graphical development environment for music and multimedia developed and maintained by a company called Cycling '74. It has been used for over fifteen years by composers, performers, software designers, researchers and artists interested in interactive software.

Max is developed in such a way that it is very modular, with most of the functionality located in so called shared libraries. This gives developers the possibility to create third-party routines, called external objects. The availability of external objects is driven by the needs from the user base using this software. Unfortunately, Max costs a lot of money.

There exists API documentation on how to create external objects, but:

- Max costs a lot of money
- Given the short period of time, it was not realistic to be able to implement external objects and to get familiar with the application itself.

Pure Data (12) is a graphical programming language, quite similar to Max. Pd is an open source project and has a large developer base working on new extensions to the program.

As with Max, there exists API documentation on how to create external objects, but:

- Given the short period of time, it was not realistic to implement, or get familiar with.

Blinkenlights (13) was a project performed in Berlin, 2002 – and a follow up later in 2002. The first project decorated a whole façade of a building with controlled lights. The file format used for their animations is available, and there exist a lot of tools for creating them in a simple way. The catch is that the number of lights is restricted to 18x8, which is too little for the amount of lights we would like to use. You have the option to scale the movie when displaying it, but that would not use all of the walls potential. Secondly, the Blinkenlights format uses only on/off for lights, while our wall can display several light intensities.

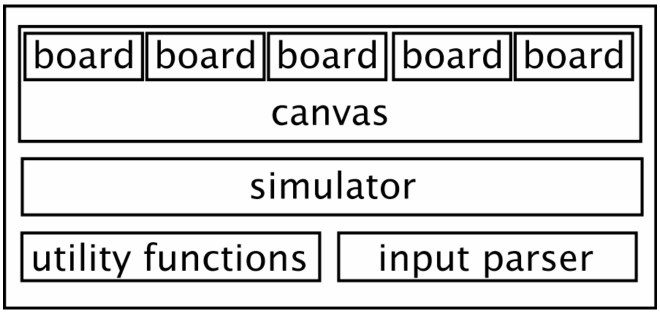
The second version of blinkenlights introduced more lights, and intensity values for the lights. This project used XML for content, and there are utils available for creating such files.

Software Development

Driving the display

We decided to create our own framework, programmed in the c language. There were several reasons for this.

- Speed – c is compiled into native machine code, giving the best possible speed without having to use a low-level language like assembler.
- Compatibility – c is a supported language among several platforms.
- External libraries – a lot of open source libraries exists, which can be used in c.



The Framework

The framework was created from scratch. See figure above for a general overview of the structure of this framework.

We wanted the following features from the framework:

- display images
- display simple animations
- display blinkenlights movies
- display raw input (that is, a long list of data)
- draw primitives like lines and filled triangles
- looping of the animations
- a simulator, so that testing can be performed without using the wall
- independent of the number of boards, orientation and direction.

Referring to the Figure, a “board” is the actual physical representation of the hardware boards and a “canvas” is a collection of such boards – the wall.

At lot of effort was put into the work, and all of the above mentioned features was implemented and tested to work. The framework handles gracefully different configurations of boards with respect to orientation and number of boards on the canvas. This simplifies the process of reusing the source code for other projects using the same kind of hardware.

Flash

Adobe Flash (14) is used for creating web applications, games, movies, and content for mobile phones and other embedded devices. Flash was introduced in 1996, and the Flash technology has become a popular method for adding animation and interactivity to web pages. Numerous devices are able to display Flash.

The Flash files, also called “Flash movies” or “Flash games” have a .swf file extension and may be a part of a web page, or a standalone application running inside a Flash Player.

In more recent versions of the Flash format, ActionScript has been introduced, giving direct control of the internal objects through programming or scripting. By using ActionScript you can, among other things, control the movement, orientation, style and appearance of the graphical content. By utilizing networking, ActionScript enables exchange of data across devices and data sources.

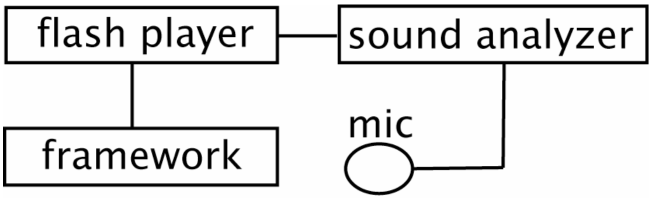
Gnash (15) is an open source Flash movie player, capable of playing a lot of Flash movies, as long as their not using the latest technology. The Flash file format is not public, so a lot of work has been done reverse engineering it. This is the reason why Gnash has a delay for supporting the newest versions of the Flash file format.

Gnash includes several back-ends for supporting several operating systems and windowing systems. We were able to create a custom backend for it, displaying Flash movies real-time on the wall by passing data to the previously mentioned framework. This process works by scaling the movie down to a size matching the wall, then converting the colour output to greyscale. The data is then passed to the framework as raw data input.

Interacting with sound

We wanted animations on the wall to respond to sounds in the room. The question was how to do this. Flash does not have features for analysing sounds, so we needed some way of exchanging analysed sound data with the Flash animation running on the wall.

The ActionScript embedded in Flash is able to exchange data, in any form, over a network. We decided to use FMOD (16) for analysing the sound. By enabling networking in the service analysing the sound, we were able to exchange data to the Flash animation.



Complete system with flash and sound analyser.

The complete system is visualized in the figure above.

The flash player passes data locally to the framework, while at the same time gathering data over a network from the sound analyser. The sound analyser program can be run on the same machine as the flash player, or from any location that have a networking connection. You can even have several sound analyser programs running at different locations. This is highly scalable.

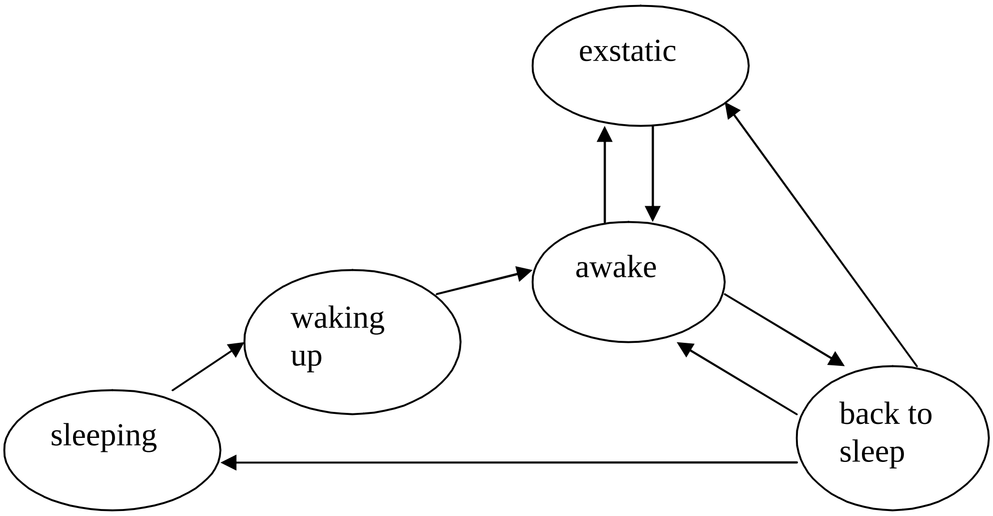
The actual sound analysation is done by frequency and volume analysis. By using FFT (Fast Fourier Transform) (17) on the sound data, the frequencies can be extracted. Pitch can also be detected by looking at the dominant frequencies for a limited period of time.

Licensing

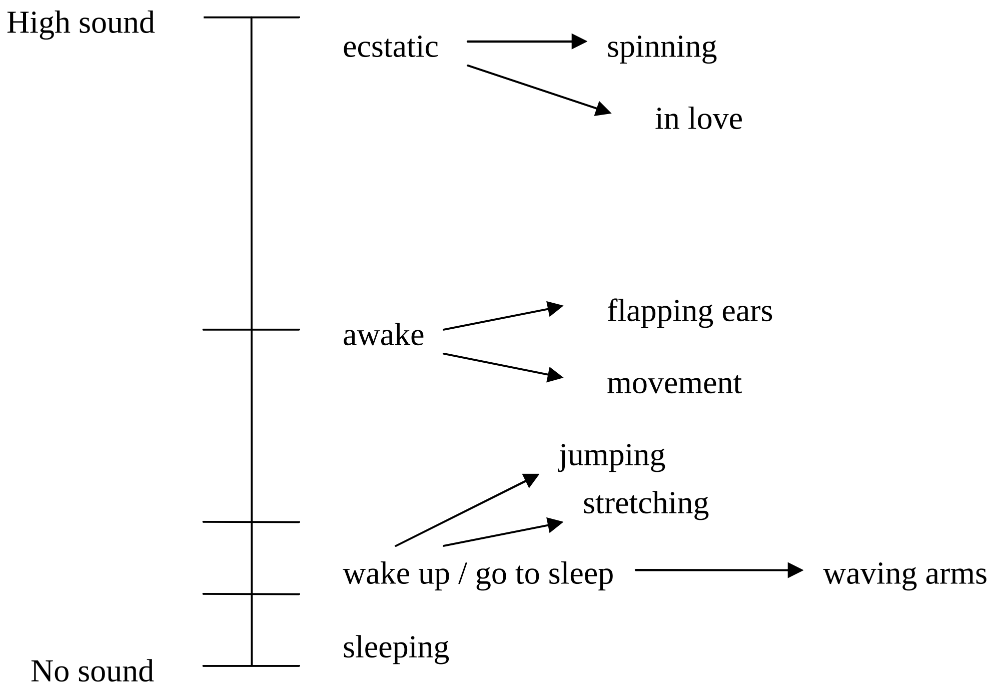
We want the work from this project to be available at no cost for anyone to use. This is a continuous piece of work, which we hope will be continued by others. There are a lot of different licensing styles for source code (18). We have chosen one that is very open, the “BSD License” (19). This license has few restrictions compared to other licenses used by open source software. The BSD License allows you to use the source code in any project, commercial or non-commercial, with or without modifications. You can even change the license and redistribute the source code under it. The Gnash (flash player) source is public domain, so it can also be modified and used without restrictions. FMOD has some restrictions when used for commercial products – it's free for non-commercial usage.

The result: LUX VITAE

We called our project LUX VITAE, meaning light and life. We have chosen to realize our vision of the interacting wall through a small bunny that will respond to certain sound levels in the room. Due to the time between the deadline for this project report and our presentation of the installation, we might want to add some extra functionality in the next two days (for example letting the bunny react to changes in the pitch of the sound as well). As of today our little friend the bunny reacts in the way depicted in the illustrations below.



The figure above describes the various modes the bunny can be in, while the arrows describe the transitions between these modes. You can only go from one mode to another in the direction of the arrows. This is also sometimes referred to as a “state transition diagram”.



This figure describes the actions taken by the bunny with respect to sound levels. It starts with no sound, where the bunny is sleeping, then there's a transition to being awake, and another transition to being ecstatic (high sound). The bunny will randomly choose among the available animations for each mode it's in.

Future projects

When working with this project, we were conscious about the reuse of this work. We would not like the project to end with this. We kept that in mind when developing the software controlling the boards, so that it would be easy to extend and reuse for future projects.
There are literally endless possibilities with the introduction of a wall rendering Flash movies. By utilizing the networking capabilities of Flash, it is possible to interact with several sources of data. This is not limited to just other software programs – you could also embed sensor technology directly affecting the Flash movies, using the same approach as we did with sound analysing.

As previously mentioned, Max/MSP and PD is a well known piece of software amongst artists creating digital art. We did not have time to investigate, and create compatible software for them – but it is realistic to do. These programs work in such a way that the flow of data is transparent. You can for instance create sound or music real-time, affecting the wall by using the sound or music as input itself – or perhaps the opposite way, creating content on the wall that is given as input to the routines creating sound or music. Enabling support for Max/MSP or PD would let artists familiar with this kind of software use the wall for their projects.

Another option would be to connect the wall to the World Wide Web, letting the community decide what is to be displayed on the wall. How about attaching a web camera in the room, letting everyone view what is on the wall?

The system for analysing sound is so scalable that you can place several microphone sources at different locations.

Conclusion

We came to the end, eventually. During these three weeks we have developed ideas and proposals, we have evaluated, planned, and programmed, animated, build, reflected, written and discussed (a lot!). The process of decision- making was very intense. We had to take choices all the time. It’s a big challenge balancing between decision- making and the production of ideas in intensive projects. Some topics could be discussed more, and some topics could be discussed less. We asked ourselves the question: How could a wall look like when the goal is interactivity between art/software and people in the room? Our solution is described in this report, but it is only hypothesis. To draw a real conclusion we need more feedback and information. The level of interactivity between art/software and the people inside the room must be evaluated by the people in the room, and we rely on their feedbacks. All we can do is to talk about it, explain, advertise and demonstrate our tool, and then see if people like it.

References

1. ^ Sheizaf Rafaeli defined Interactivity as "an expression of the extent that in a given series of communication exchanges, any third (or later) transmission (or message) is related to the degree to which previous exchanges referred to even earlier transmissions. Rafaeli, 1988
2. http://www.answers.com/interactivity#wp-_note-0
3. http://www.answers.com/topic/new-media-art
4. McQuuail, Denis (2005) *McQuuail's mass communication theory*,London:Sage publication:143
5. Digital Visions: Computers and
Art (New York: Harry N. Abrams, 1987)
6. The Digital Revolution: Art in the Computer Age, by Cynthia Goodman
Art Journal © 1990 College Art Association, Inc.
Published by College Art Association
7. Ibid. Interview with Ruth Leavitt, January 27, 1987
8. Graphic art software, http://en.wikipedia.org/wiki/Graphic_art_software
9. http://odc.opentheweb.org/wiki/Main_Page
10. http://itk.samfundet.no/dok/Lysfontene
11. http://www.cycling74.com/products/maxmsp
12. http://puredata.info/
13. http://www.blinkenlights.de/
14. http://www.adobe.com/products/flash/
15. http://www.gnu.org/software/gnash/
16. http://www.fmod.org/
17. http://en.wikipedia.org/wiki/Fast_Fourier_transform
18. http://en.wikipedia.org/wiki/Comparison_of_free_software_licences
19. http://en.wikipedia.org/wiki/BSD_license