Multi-touch technologies, the reacTable* and building a multi-touch device for use in composition and performance

Timothy Roberts

s2599923

Subject: Music Technology 6

Course Code: 3721QCM

Lecturer: Dave Carter

Word Count: 5165

Abstract

As time passes the devices that are used to interact and manipulate sound changes. This paper investigates a technology that has seen rapid advancement in the past decade, multi-touch. This paper looks at current technologies used in optical multi-touch devices. Also covered are the concepts and techniques behind the Music Technology Group's reacTable*. The paper also includes a discussion on aesthetic considerations made during the construction of a musical multi-touch device and concludes with the process of creating the hardware and software of the device. This is done through study of literature as well as action research for the programming of the device. The paper concludes with options for further research in the field of multi-touch devices with the software and final patches for the multi-touch device included in the appendices for both Windows and OS X.

Table of Contents

Table of Figures

Introduction

As time passes, the way that composers, performers and audience interact with sound and music changes. Styles of music, popular culture and the instruments used to create sound also change. An area that has seen development within the past couple of years is the use of multi-touch panels and displays as a way of creating and manipulating sound. This paper will investigate the technology currently available for multi-touch devices and the reacTable* created by the Music Technology Group based at the Universitat Pompeu Fabra. The paper will also document the development of a multi-touch device for composition and performance.

Open source software, such as reacTIVision (Music Technology Group, 2009) and Community Core Vision (NUI group, 2009), have brought the ability to build multi-touch instruments to the masses and has inspired many different iterations. These include the Tacchi[1], Audio Touch[2], ReacTable Role Gaming[3], Brick Table 2.0[4] and many more. It is a combination of these new interfaces and the original ideas that has inspired me to conduct research, into building a multi-touch device for creating music. Puckette's (2007), *The Theory and Technique of Electronic Music*, my own previous research into Pure Data[5] and reacTIVision have also inspired me to conduct this research. I will be creating a multi-touch device for composition, performance and installation works.

It is my belief that this research should be undertaken to increase awareness and knowledge around using afore mentioned software for creative purposes, such as composition and performance. It is also my hope that the paper will be informative for other music technologists who are looking for new ways to control sound, as well

1 http://rjmarsan.com/thereactable/
2 http://sethsandler.com/audiotouch/
3 http://nuigroup.com/forums/viewthread/4177/
4 http://flipmu.com/work/bricktable/
5 http://puredata.info/

as those wishing to build a multi-touch device.

Literature Review

When building a multi-touch device, as with most things, research into what has already been written is an important step. Much of the research within the field of multi-touch devices is concerned with improving existing techniques and creating new techniques for sensing of multiple touches. Research has also been conducted into the usability of these devices. However there is a lack of holistic sources containing information on the technology as well as the process of creating a device.

Sources that I have used include books, papers, software manuals, and blogs about multi-touch devices as well as websites. I am focussing on one main book, for information on multi-touch technologies, but there are also other papers that have valuable information that I will also be cross referencing.

Multi-touch Technologies (Nui Group, 2009) contains details on the hardware technology and software for multi-touch devices. "...the Natural User Interface Group ... is an open source interactive media community researching and creating machine sensing techniques to benefit artistic, commercial and educational applications" (NUI Group, 2009). The group's main focus is in "accelerating [the] development of existing sensing and visualization solutions" (NUI Group, 2009). I will be using this book for general information on multi-touch technologies, in particular the type of sensing to be used in the creation of the device. However because this book is an online community project, the information contained should be cross referenced against other scholarly papers.

The book can be split into two main sections, hardware technologies and software and applications. Within the hardware section it discusses optical sensing techniques as well as other hardware components of a multi-touch device. The software and applications section is concerned with tracking, gesture recognition and using different programming environments to create software for the devices. The book also includes examples of building three different types of multi-touch devices.

Puckette's excellent *The Theory and Technique of Electronic Music* (2006) also informs this research in relation to creating the audio interface and sonic palette, the sounds available for use. As Matthews notes, Puckette's work is a "uniquely complete source of information for the computer synthesis of rich and interesting musical timbres" (2007). The book presents the theory behind many types of synthesis and effects, but also contains useful exemplars of each theory within the Pure Data environment. These exemplars allow rapid creation of sounds to start the composition process.

Methodology

Research for this paper was conducted in two ways. A literature survey to inform choices in regards to the construction, and programming of the device. An action research method was then employed to observe and reflect on the creation of my device. By employing a qualitative research method of action research backed by the literature survey I believe that the result is a much more holistic approach to documenting the development of a multi-touch device for composition and performance. Figure 1 shows a visual representation on the framework that I used to conduct the research.

By using the framework of plan, action, observe and reflect it allowed me to be both 'inside' and 'outside' the building of the device allowing me to view the process subjectively.

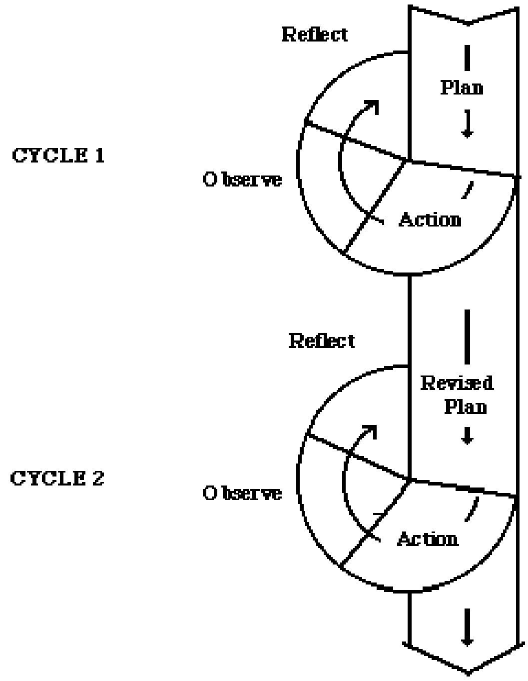


Figure 1: Simple Action Research Model
Retrieved from O'Brien, R. (1998). An overview of the methodological approach of action research. Retrieved October 24, 2009, from http://www.web.net/~robrien/papers/arfinal/html

Multi-touch Technologies

Multi-touch technologies encompass any "set of interaction techniques that allow computer users to control graphical applications with several fingers" (NUI Group, 2009, p. 2). These technologies are currently employed in a myriad of devices including Apple's iPhone[6], the Microsoft Surface[7] and a large variety of DIY devices such as the Audio Touch. Many of these devices, particularly the DIY devices, are 'proof of concept' devices which use applications designed for demonstrating the device. However a move in recent times towards using multi-touch devices for both creative and serious purposes, has resulted in devices such as the, JazzMutant Lemur[8] and Dexter[9] and the Music Technology Group's reacTable*[10]

6 http://www.apple.com/iphone/
7 http://www.microsoft.com/surface
8 http://www.jazzmutant.com/lemur_overview.php
9 http://www.jazzmutant.com/dexter_overview.php
10 http://www.reactable.com/ (2009a)

appearing in the market. A majority of the devices are optical based, however there are other techniques for sensing touch, which include “proximity, acoustic, capacitive, resistive, motion, orientation, and pressure” (NUI Group, 2009, p. 2).

All optical based multi-touch systems consist of a camera, surface, a system of visual feedback and a form of lighting the surface, which is usually in the infrared spectrum. By restricting the camera to only a section of the infrared spectrum interference from the visual feedback is avoided. Optical multi-touch devices work by creating hot spots of light when the surface is contacted. These hot spots, called blobs, are recognised by tracking software, which then outputs the position of the blobs.

As mentioned earlier, and can be seen in figures 2-4, light is generally supplied by IR or Infrared Light Emitting Diodes (LED’s). Infrared light is a section of the light spectrum which is just above the visible range of the human eye. There are various reasons for why IR is used to illuminate the surface. The first of these is that most digital camera sensors are sensitive to this bandwidth of light, however many cameras also have a filter to remove this part of the spectrum to limit them to the visible spectrum. “By removing the infrared filter and replacing it with one that removes the visible light instead, a camera that only sees infrared light is created” (NUI Group, 2009, p. 3). By doing this it is possible to avoid misinterpretation of the visual feedback used in most devices.

Most multi-touch devices have a system of giving visual feedback to the user. This is usually achieved by either a projector or a Liquid Crystal Display (LCD) monitor. If using a projector there are a number of things which should be taken into consideration. One of these things is the throw distance of the projector. “This is the distance that is needed between the lens of the projector and the screen the get the right image size." (NUI Group, 2009, p. 23) A mirror can be used to increase the throw distance of the projector. The second display method is to use an LCD screen. "All LCD displays are inherently transparent – the LCD matrix itself has no opacity" (NUI Group, 2009, p. 24). This means that if the casing and IR blocking diffusers are removed, the matrix can then be used in a multi-touch device as it allows IR light to pass through it. This assumes that the other required components, power supply and controller boards, can be moved far enough away to avoid obstructing the IR light from the surface.

Currently there are five major sensing techniques that are used in optical multi-touch devices. These are Frustrated Total Internal Reflection (FTIR), Diffused Illumination (DI), Laser Light Plane (LLP), LED-Light Plane (LED-LP) and Diffused Surface Illumination (DSI) (NUI Group, 2009, p. 2). Each of these techniques has advantages, but also disadvantages. The advantages and disadvantages for the techniques not discussed can be found in the *Multi-touch Technologies* book from NUI Group.

A standard FTIR device will consist of plexiglass, a silicone layer, a projection surface and a frame containing LEDs to shine through the side of the plexiglass. A simple example of an FTIR setup can be seen in figure 2. The benefits of FTIR include stronger blob contrast and an enclosed box is not required. This technique allows for varying blob pressure and when using a compliant surface, it can be used with an object as small as a pen. However, an FTIR setup cannot recognise objects or fiducial markers (symbols) and requires a compliant surface as well as an LED frame (NUI Group, 2009; Han J., 2005).

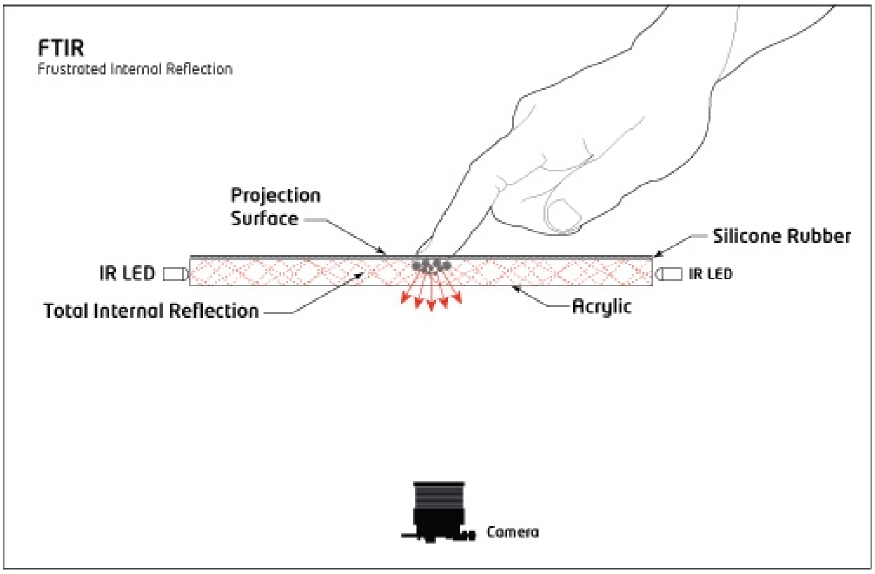


Figure 2: FTIR Schematic
Retrieved from Roth, T. (2008). DSI – Diffused Surface Illumination. Retrieved October 27, 2009, from http://iad.projects.zhdk.ch/multitouch/?p=90

Another technique that is used is DSI. The advantages of a DSI setup include even finger and object illumination throughout the surface as well as being pressure sensitive. It also allows for detection of objects, fingers and fiducials. It also has the advantages of not needed a compliant surface and has no hotspots. However a special type of acrylic is needed, which costs more than regular acrylic, blobs have lower contrast than FTIR and LLP, and there are possible size restrictions due to the softness of plexiglass (NUI Group, 2009; Roth, T., 2008). An example of DSI can be seen in figure 3.

Rear DI is the last technique that I will discuss. This technique gives the ability to track objects, fingers and fiducials. It doesn't require an LED frame or soldering of single LED's, as the illuminators can be bought pre-assembled. Rear DI doesn't need a compliant surface and any transparent surface can be used. However it can be difficult to achieve even illumination of the surface and the blobs have a lower contrast. There is also a greater chance of false blobs and an enclosed box is required (NUI Group, 2009). An example of a Rear DI setup can be seen in figure 4. This sensing technique is used in many devices including the Microsoft Surface and the Music Technology Group's reacTable*.

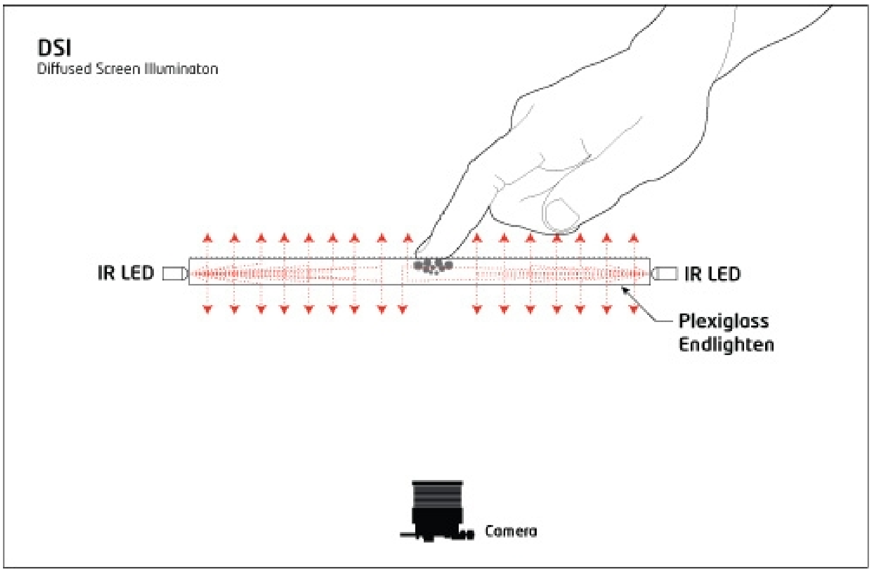


Figure 3: DSI Schematic
Retrieved from Roth, T. (2008). DSI – Diffused Surface Illumination. Retrieved October 27, 2009, from http://iad.projects.zhdk.ch/multitouch/?p=90

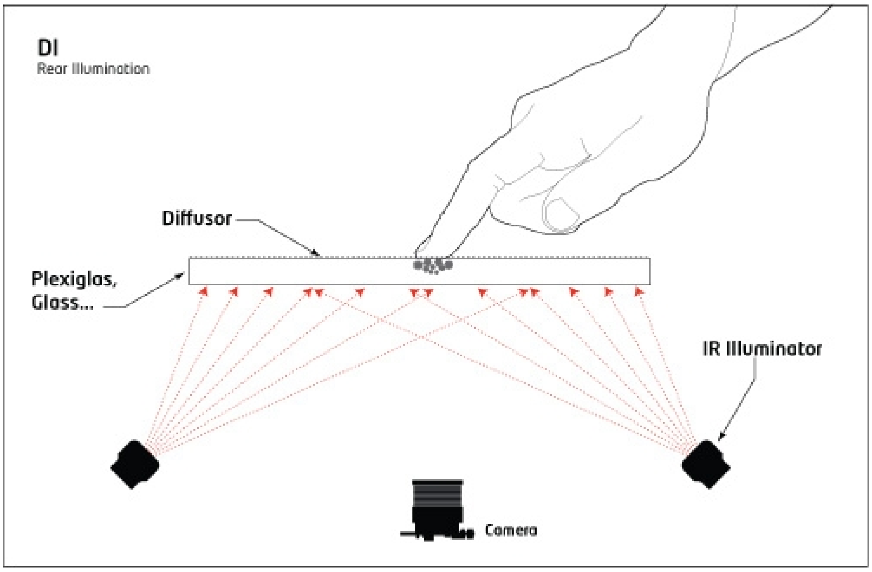


Figure 4: DI Schematic
Retrieved from Roth, T. (2008). DSI – Diffused Surface Illumination. Retrieved October 27, 2009, from http://iad.projects.zhdk.ch/multitouch/?p=90

The reacTable*

The reacTable* is a "collaborative electronic music instrument with a tabletop tangible multi-touch interface" (Music Technology Group, 2009b). It was created by a team from Music Technology Group based at the Universitat Pompeu Fabra in Barcelona. It was originally described as a "novel multi-user electro-acoustic music instrument with a tabletop tangible user interface" (Jordà, et al., 2005). It uses a rear DI sensing technique to detect fiducials, special symbols, which are placed on the surface. By moving tangibles, objects with the fiducials attached to them, it is possible to control sound and vision. An overview of the system can be seen in figure 5.

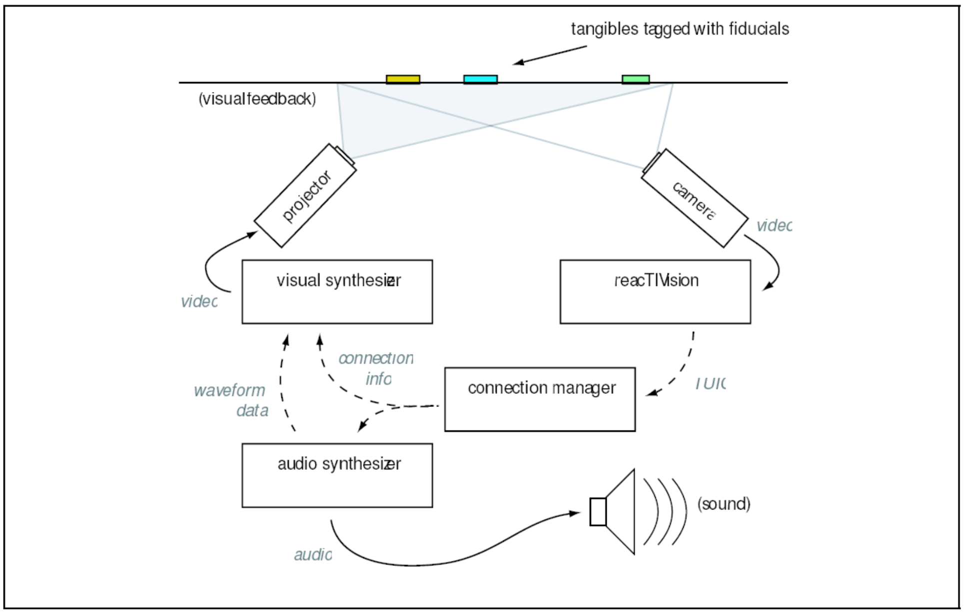


Figure 5: The reacTable* architecture

Retrieved from Jordà, et al. (2005). The reacTable*. Retrieved October 9, 2009, from http://modin.yuri.at/publications/reactable_icmc2005.pdf

As a simple explanation, vision of the fiducials from the camera is processed by the reacTIVision software. The software detects which fiducials are on the surface as well as the position and rotation of each. The position and rotation is then sent to the visual synthesizer and audio synthesizer to give visual and aural feedback. In this way, the audio and visuals are controlled the tangibles.

Early in the development process, the jobs the tangibles controlled could be split into “seven different functional groups: Generators, Audio Filters, Control Filters, Mixers, Clock synchronizers and Containers” (Kaltenbrunner, et al., 2004, p. 2). This was later refined to six categories in 2005, see figure 6, and has remained the same since (Jordà, et al., 2007).

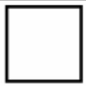
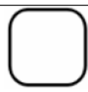
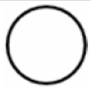
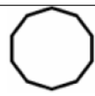
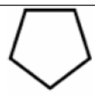
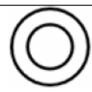
	Connections	Shape	Examples
Generators	▪ 1 audio out ▪ N control in		▪ square wave ▪ sampler player
Audio filters	▪ 1 audio in ▪ 1 audio out ▪ N cntrl in		▪ resonant filter ▪ flanger
Controllers	▪ 1 cntrl out		▪ sine wave low frequency oscillator ▪ 12-step amplitude sequencer
Control filters	▪ 1 cntrl in ▪ 1 cntrl out		▪ decimator ▪ sample & hold
Audio mixers	▪ 2 audio in ▪ 1 audio out ▪ N cntrl in		▪ mixer bus ▪ ring modulator
Global	▪ N cntrl in		▪ metronome ▪ tonalizer

Figure 6: A summary of the reacTable objects types.

Retrieved from Jordà, et al. (2007). The reacTable: Exploring the synergy between live music performance and tabletop tangible interfaces. Retrieved October 9, 2009, from http://modin.yuri.at/publications/reactable_tei2007.pdf

When building the reacTable* the Music Technology Group envisioned that the audio element would be similar to Max/MSP, a graphical high level programming language, but were very aware at the start that they were building "an instrument, not a programming language" (Jordà, 2003, p. 6). This concept of building an instrument was later elaborated on. "Building the instrument is equivalent to playing it and vice-versa, and remembering and repeating the construction of a building process can be compared to the reproduction of a musical score" (Kaltenbrunner et. al, 2004, p. 2). Due to this, the reacTable* had to work and produce an audible result when interacted with. "There is not anything like an editing mode and running mode (at least for installation users); the reacTable* is always running and being editing" (Jordà, 2003, p. 6). By doing this the device would avoid causing the user frustration.

The concept behind the audio creation is expanded upon in Kaltenbrunner, Geiger and Jordà's *Dynamic Patches for Live Musical Performance*. The reacTable* was designed to be similar to a modular synthesizer and initially relied on Puckette's Pure Data (PD) to generate audio (2004). PD contains a large amount of basic synthesis and control objects which were used in the generation of audio. Examples of these objects are oscillators, wave-table oscillators and a variety of effects such as high pass, low pass and band pass filters. In creating the patch for audio creation, each tangible was assigned a particular patch within the larger patch, also known as an abstraction. Each of these abstractions could then be added and removed at anytime (Kaltenbrunner, et al., 2004).

ReacTIVision, the software used for processing the video stream and identifying symbols, has been released for free under a GPL license and is currently at version 1.4. The software works by analysing the video stream for fiducial markers. Once a marker has been identified, its rotation and position is calculated and sent out of the program according to the TUIO protocol. This data can then be received by any program capable of receiving these messages. Multiple TUIO clients have been released by the Music Technology group for programming environments such as Java, Pure Data, Max/MSP, Flash and more (Music Technology Group, 2009c). The Music Technology Group has not however released the other software that is used for either the visual feedback or audio creation. It is for this reason that I built a device similar to the reacTable*, as well as to add to the number of multi-touch devices using multi-touch technologies in creative ways.

Building my Reactable

The process of building a Reactable can be split into 2 distinct sections, the hardware and the software. However before starting either of these, it was important for me to think about how my final production would be used and the aesthetics that would accompany it. My original idea was completely remove the visual feedback so that the device is almost completely about the sound. When performing with the device the room could be completely dark, allowing the audience to be completely consumed by the sound rather than what the performer was doing. Removing the visual feedback would also give extra time for programming of the audio system. This was particularly important given the three month duration of the project.

Another idea was to make the technology as invisible as possible. Donaldson makes the following comments in regards to laptop performance. “The screen of the laptop forms a barrier between the audience and the performer, preventing some audience members from seeing the performer’s face, and preventing them from seeing what the artist is doing” (2006, p. 713). By hiding the computer completely, I felt it would be possible to remove this barrier. To make the device as simple to setup as possible I also decided that it should only require a single power lead to function, have the option between speakers within the device and an audio out. Continuing this, the device should only need to be turned on with the required programs launching automatically.

It was also important that the device be playable by multiple people, in a similar style to the reacTable*. Because of this the device must be intuitive to use and like the reacTable* any gesture should give audible results. Labelling the tangibles with what they control was also very important allowing users to distinguish between each tangible.

Hardware

After deliberation I settled on using rear DI as the sensing technique. This allowed me to wire the IR LEDs to be plugged into the power supply for the computer that would be inside the device. However, I could not achieve an even illumination with the two IR lights that I had. After I completely changed direction with the lighting and placed a single light-bulb inside the box. This created a problem with reflections off the glass removing the ability to use symbols in certain parts of the device. This was solved by placing the light at one end of the box and reducing the area that symbols were active in to avoid the reflection. By reducing the performance space it gave an area that the tangibles could be placed while not in use. This can be seen in figure 7 with the performance space on the right of the surface.

Figure 7: The reacTable during a performance at Sound Composition 09
Retrieved from Roberts, T. (Producer). (2009a). *Sound composition 09 reacTable Digital portfolio.* [Youtube]. Brisbane, Qld. Retrieved October 28, 2009 from http://www.youtube.com/watch?v=r4zHIoTRgzk

The original size of my Reactable was based on both the maximum size possible with a camera at a resolution of 640 x 480 as well as the size of the tray that the motherboard for the computer sits on. The final design can be found in appendix A. However this design was slightly modified to include a door on the back rather than a hole. A false floor for the camera and lights to sit on as well as hide the computer was also added.

Software

When programming the audio and sound generation for my Reactable, I took an approach similar to the Music Technology Group and assigned each tangible to a particular sound or generator within the Pure Data environment. It became apparent very early that the creative process begins when assigning sounds to each tangible, rather than just during the performance. The programming creates the building blocks, which will be combined during the performance or installation and as such is the genesis of the creative process. As the main performance techniques are improvisation and sound installations, I believe that this device belongs to the tradition of blurring the line between, performer and audience particularly when in a sound installation context. However I believe that the device also blurs the line between composer and performer. When programming I used an action research framework and methodology, to give structure to the process. By splitting the entire process into smaller segments or cycles, also allowed by to quickly find what was working and what wasn't. The final Pure Data patch, after further work beyond the scope of this paper, can be found in appendix B along with required software for both Windows and OS X.

Cycle Number 1

My plan at the start of the first cycle was to simply get sound from the device. I also planned to start learning to program within the Pd environment. I did this by finding similar projects to this one online. The two examples that I chose to use were the TUIO Theremin patch contained within the TUIO Pd client from the reacTIVision site, and a patch called cutre_reactable_v0.0pd from musa's blog (musa, 2007). After loading these patches into the reacTable I set about experimenting and creating different sounds using the downloaded patches. I also started modifying some of the values to see how the sound produced was affected. Throughout this process the main thing that I observed is that sine tones and square waves are not pleasant to the ear when the fundamental frequency is above 2-3 kHz. This is around the highest notes playable on a violin and among the top notes on a piano. Reflecting on this I decided to limit my Reactable to producing most sounds below this. I also found that using parts from other patches is an effective method for learning to use the software. Often the sounds generated in the patches are close to what is needed, but must be slightly modified in order to fit properly. It was during this cycle that I realised that process of composing for the device would be very different to composing for a traditional instrument.

Cycle Number 2

Continuing on from cycle number 1, my revised plan consisted of creating a playable Pd patch for a couple of symbols. I also needed to figure out a way to create a score that can be played. In this way it would be possible to give a performer a framework to create and improvise within. This would also allow for a body of work to be written for the Reactable.

The first symbol that I added was an On/Off symbol. I did this because I felt that if the audio system was to lock it would be important to have a master switch. This symbol is linked to enabling and disabling the compute audio function. The second symbol that I added was a simple sine tone. This symbol had its X position linked to its volume and the angle to the pitch created. The third symbol added was a low frequency oscillator (LFO) and uses all 3 parameters. The X position is linked to the volume, the Y position is linked the pitch and the angle to the frequency of oscillations. The fourth symbol that was added was a square wave which was adapted from the example patches that accompany Puckette's T*he Theory and*

Technique of Electronic Music, and is controlled in the same way as the sine tone. Figure 8 shows the implementation of the first four symbols.

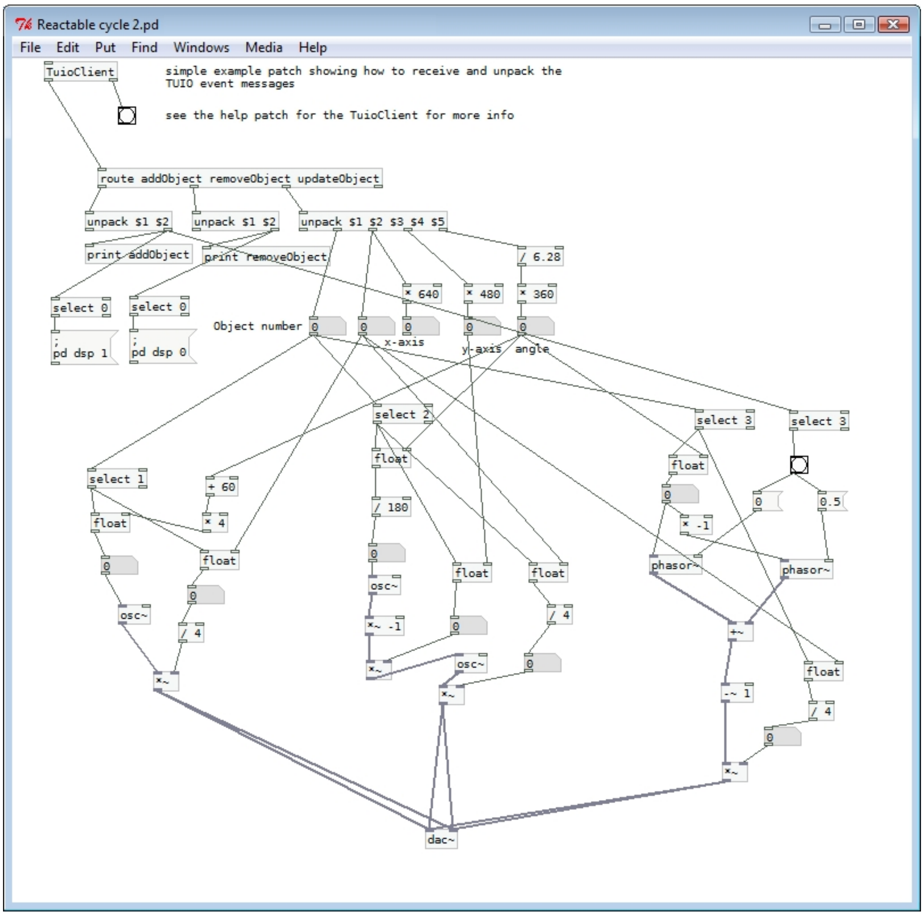


Figure 8: Screen capture of Pure Data patch during cycle number 2

The final sound that was assigned to a tangible was sample playback. The X position again controls the volume and the playback speed is determined by the rotation. This was adapted from the cutre_reactable_v0.0.pd patch (musa, 2007). I also settled on a way of composing which is as simple as possible leaving the performer to interpret as s/he chooses. A graphical score of a composition can be seen in figure 9. The line maps the intended emotional contour of the piece in relation to a timeline. The player is then free to improvise and play, but while trying to match the contour of the score.

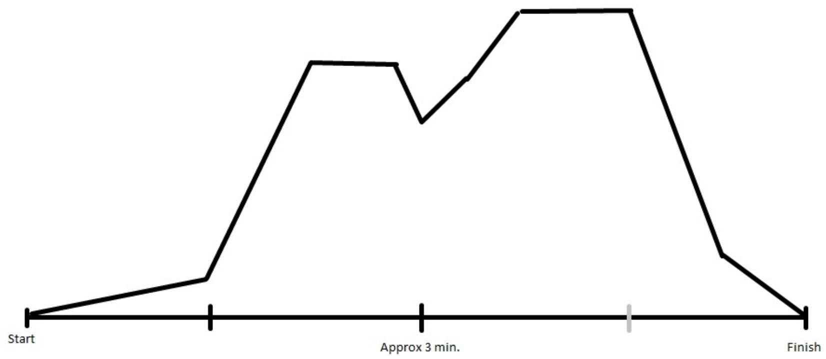


Figure 9: A graphical score for a composition for the Reactable

Retrieved from Roberts, T. (2009b). Sound Composition 09. [Score]. Unpublished graphical score. Griffith University, Brisbane, Australia.

The time needed to complete the action part of this cycle was relatively quick, as most of it was aggregating separate patches into a single one. This then allowed me to control multiple sounds at once. However testing many different patches to find ones that made sounds that I liked was time consuming. This was mainly due to trying to work out how each of the patches worked and how each patch could be implemented into the Reactable.

Reflecting on this cycle I found that the biggest thing that needed to change was that symbols must stop making sound when removed from the table. Another thing that is very important is making sure that the output of each symbol is always multiplied by a value between 0 and 1 to occurrences of the audio system becoming overloaded. The final thing that realised is that I needed to use the Y position to control the sound in some way. However the minimal use of the Y axis allowed for symbols to be moved around the space without altering the sound being generated.

Cycle Number 3

The plan for this cycle was to add more sounds, allow for more than 1 sample to be played at the same time and at different speeds. I also planned to start using effects like high pass, low pass and band pass filters and to implement sound generators with a random element.

The action part of this cycle began by adding an off function to all the previous sounds. I continued by adding two sounds which use a form of granular synthesis. These sounds were much more complex that the previous sounds and consisted of at least four parameters to be controlled. To do this I decided on using two symbols to accomplish one job. The first of the symbols controls the distance between the highest and lowest note with the X position, the lowest frequency by the Y position and the speed of the synthesis with the rotation. The second symbol controlled the volume with the rotation and the frequency of a hi-pass filter with the X position. Contained within this patch is an array which influences the sound created. This is one parameter that I decided wouldn't be controllable by the user. This patch was duplicated, but the array was changed to give two different sounds. Figure 10 shows the first of the two granular synthesizers. A modification that I made during this cycle was to limit the playback speed of the samples to 1 and -1, while having the symbol rotate from 2 to -2. This gives 90 degrees in which the playback speed is constant and at the original speed.

Throughout the process I observed the patches being implemented becoming increasingly complex. This made finding what was happening in each patch more difficult. A prime example of this is the granular synthesiser seen in figure 10. I understood where to connect the X position, Y position and rotation, but I didn't know how and why each object was used and the part that it played. The patch for the granular synthesizer was adapted from 3-7-3-1-granularsynthesizer.pd (Puckette, 2007). A noticeable highlight during this cycle was the increased depth of sounds and compositional variety when random objects are used. For example, the random generators could be used as a non-static rhythmic bed with other generators as 'soloing' instruments over the top.

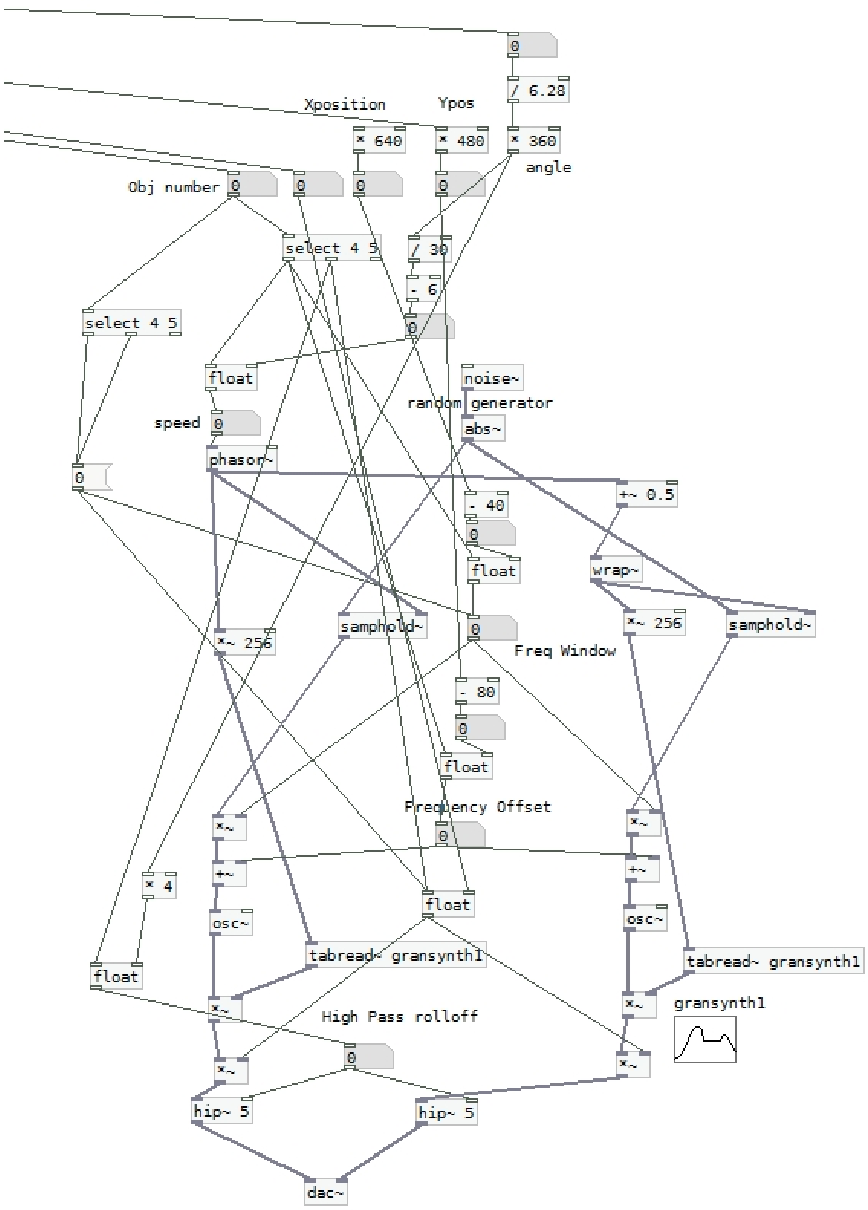


Figure 10: Granular Synthesizer patch

After playing with this iteration of the software I found that I needed to implement a symbol which isn't controlled as a continuous variable to add some sort of tonality for the ear to latch onto. I also found that controlling a single sound with two different tangibles was not a simple thing to do, and required a fair amount of explaining when others are using these objects. I discovered this through allowing peers to ‘have a go’ of the device. The user would have no trouble recognising the sine tone or square wave symbol, but would almost always ask what each of the tangibles labelled 1A, 1B, 2A and 2B did. A simple reply explaining the need to use both 1A and 1 B to make sound would generally be sufficient.

When reflecting on this cycle, there were many things that came to my attention that should be changed or added. These included standardising the X position to control the volume for all generators and adding the option for sounds to playback just once (one-shot). These sounds, as well as the looping sounds, must load automatically. A final point that came to the fore while reflecting is that the patch is became quite crowded and messy. In response to this, I decided to look into ways of simplifying the patch.

Cycle Number 4

Following on from my reflection in cycle three, I planned to create a symbol that is limited to 6 notes. I also planned to add symbols which play once, make all samples load before the tangibles for these are added to the performance area, as well as find out how to use abstractions and sub patches. Standardising the X position to control the volume was also included in the plan.

The 6-note symbol was the first sound that I added which didn’t find its genesis in another patch. The patch was constructed by splitting the data stream of the rotation, into sixths. Each of these sections was then linked to a separate oscillator and the appropriate on and off signals. This patch was then implemented into the reacTable patch three times to give a range of three octaves for the user to use across three tangibles. One-shot sample playback was also implemented allowing for short samples to be used. One use of samples that only play once is the creation of a timer like device. When a tangible is added to the performance area, a timer is started. After a pre-defined amount of time a sample is played a single time letting the performer know how far through the performance they are. As a simple option instead of sub patches I used send and receive objects, which transfer data without the need to be connected. This allowed me to send data anywhere in the patch and create visually independent patches within the larger patch. Standardisation of the X position controlling the volume was also implemented.

During this cycle I realised that creating patches from scratch is quite a difficult thing to do, but it enforced my belief that the programming of the device is part of the creative process. The process of creating an idea, which is then fulfilled through the addition of a new sound as well as adding another layer of creativity to the device brings great satisfaction, which far outweighs the difficulty. As always more generators and effects could be added, but this is something to consider for future work.

Conclusions and Future work

This paper has discussed the current technologies used in optical multi-touch devices, with a focus on the way that the camera is able to detect touch. This is through a variety of different surfaces and lighting techniques such as FTIR, and DSI. The concepts and techniques behind the creation of the Music Technology Group's reacTable have also been discussed. Finally I discussed the aesthetic considerations of constructing a tangible multi-touch device similar to the reacTable. I also described the process of creating the hardware and the programming of the software.

By conducting the research for this paper I have been able to create a device which can be used as a compositional and performance device. This has introduced me to a new creative process as well as new sonic palettes.

Future work will include research into a main feature of the original reacTable, dynamic patching (Kaltenbrunner et al., 2004). If this feature were to be implemented a plethora of extra sounds would be added as well as another level for the player to master. A simplification of the controls for the granular synthesizer may also be considered in the future, although this is a component which adds a level of complexity. Another which should be considered in the future is the use of the reacTable to control the sound of an acoustic musician by using the sound to trigger certain events within the environment. My research into this area will also be continuing with plans to construct a multi-touch device, with visual feedback for tactile mixing and audio creation.

Reference List

Apple. (2007). iPhone. Retrieved August 13, 2009 from http://www.apple.com/iphone/

Donaldson, J., (2006). Limestick: designing for performer-audience connection in laptop based computer music. *CHI '06 extended abstracts on human factors in computing systems,* 712-717.

FlipMu. (2008). Brick Table 2.0. Retrieved October 11, 2009, from http://flipmu.com/work/bricktable/

Han, J. (2005). Low-cost multi-touch sensing through frustrated total internal reflection. *UIST '05: Proceedings of the 18th annual ACM symposium on user interface software and technology, 115-118.*

JazzMutant. (2005). Lemur. Retrieved August 13, 2009, from http://www.jazzmutant.com/lemur_overview.php

JazzMutant. (2007). Dexter. Retrieved August 13, 2009, from http://www.jazzmutant.com/dexter_overview.php

Jordà, S. (2003). Sonographical Instruments: From FMOL to the reacTable*. Universitat Pompeu Fabra, Barcelona, Spain. Retrieved August 29, 2009, from http://mtg.upf.edu/files/publications/nime03-jorda.pdf

Jordà, S., Geiger, G., Alonso, M., Kaltenbrunner, M. (2007). The reacTable: Exploring the synergy between live music performance and tabletop tangible interfaces. Universitat Pompeu Fabra, Barcelona, Spain. Retrieved August 29, 2009, from http://modin.yuri.at/publications/reactable_tei2007.pdf

Jordà, S., Kaltenbrunner, M., Geiger, G., Bencina, R. (2005). The reacTable*. Universitat Pompeu Fabra, Barcelona, Spain. Retrieved August 29, 2009, from http://modin.yuri.at/publications/reactable_icmc2005.pdf

Kaltenbrunner, M., Geiger, G., Jordà, S. (2004). Dynamic patches for live musical performance. Universitat Pompeu Fabra, Barcelona, Spain. Retrieved August 29, 2009, from http://modin.yuri.at/publications/reactable_nime2004.pdf

Marsan, RJ. (2009). Tacchi. Retrieved October 9, 2009, from http://rjmarsan.com/thereactable/

Matthews, M. (2007). [Foreword] In M. Puckette, *The theory and technique of electronic music* (Draft). World Scientific Publishing Co. Pte. Ltd. Retrieved August 13, 2009, from http://crca.ucsd.edu/~msp/techniques/latest/book.pdf

Microsoft Corporation. (2007). Microsoft Surface. Retrieved August 13, 2009 from http://www.microsoft.com/surface

Musa. (2007, October 21). Cutre reactable: Home-made reactable with reactivision and puredata. Message posted to http://musa.poperbu.net/index.php?option=com_content&task=view&id=40&Itemid=1

Music Technology Group. (2005). reacTable*. Retrieved August 13, 2009, from http://www.reactable.com/

Music Technology Group. (2009a). reacTIVision (Version 1.4). [Computer Software]. Barcelona: Spain. Available from http://reactivision.sourceforge.net/

Music Technology Group. (2009b) The reacTable. Retrieved August 13, 2009, from http://mtg.upf.es/reactable/

Music Technology Group. (2009c). reaTIVision. Retrieved August 13, 2009, from http://reactivision.sourceforge.net/

NUI Group. (2009). Community Core Vision (Version 1.2). [Computer Software]. Available from http://ccv.nuigroup.com/

NUI Group Authors. (2009). *Multi-touch technologies.* Retrieved August 13, 2009, from http://www.nuigroup.com/

O'Brien, R. (1998). *An overview of the methodological approach of action research.* Retrieved October 24, 2009, from http://www.web.net/~robrien/papers/arfinal/html

Puckette, M. (2007). *The theory and technique of electronic music* (Draft). World Scientific Publishing Co. Pte. Ltd. Retrieved August 13, 2009, from http://crca.ucsd.edu/~msp/techniques/latest/book.pdf

Puckette, M. (2009). Pure Data (Version 0.41.4). [Computer Software]. Available from http://puredata.info

Roberts, T. (Producer). (2009a). *Sound composition 09 reacTable Digital portfolio.* [Youtube]. Brisbane, Qld. Retrieved October 28, 2009 from http://www.youtube.com/watch?v=r4zHIoTRgzk

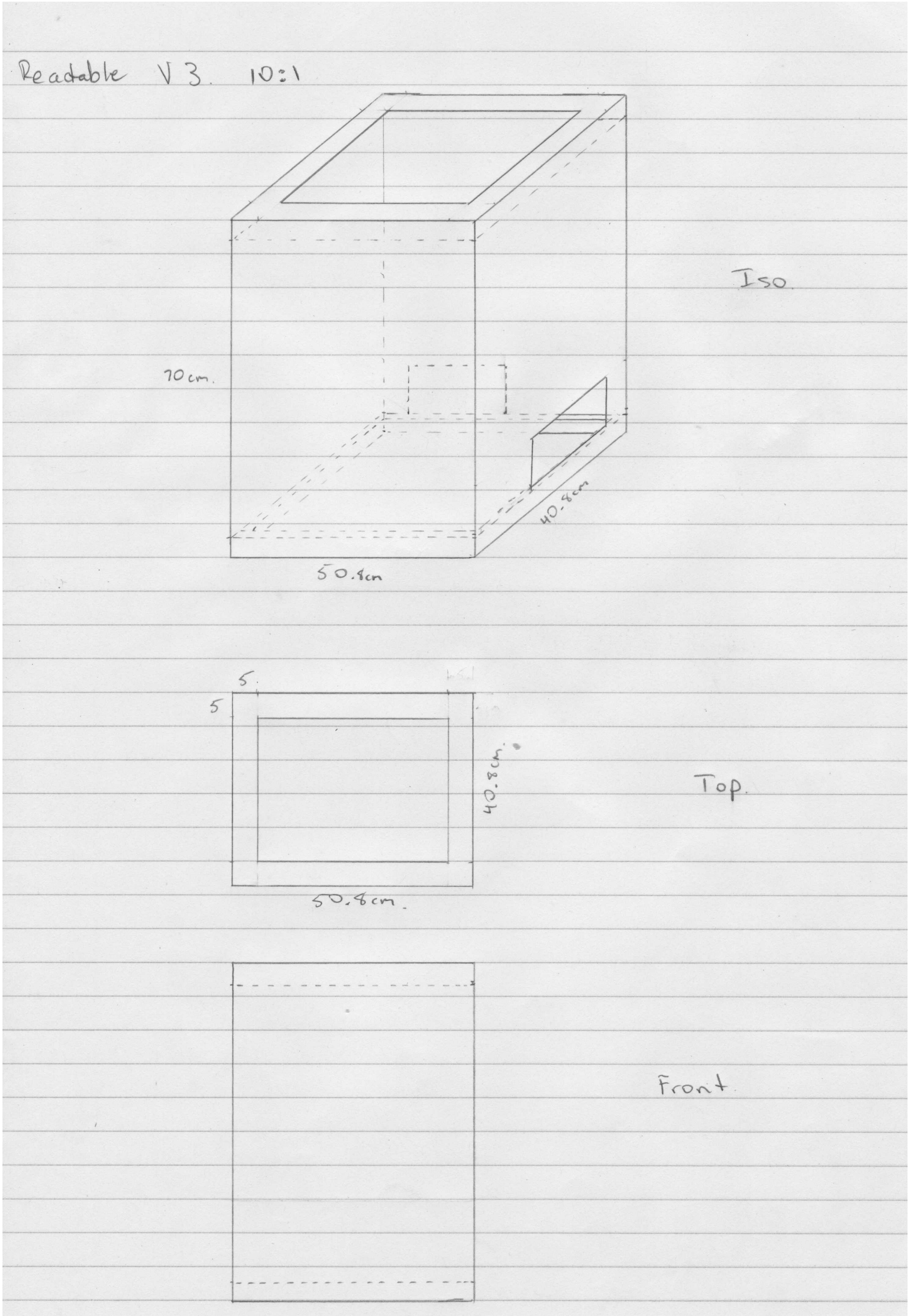
Roberts, T. (2009b). Sound Composition 09. [Score]. Unpublished graphical score, Griffith University, Brisbane, Australia.

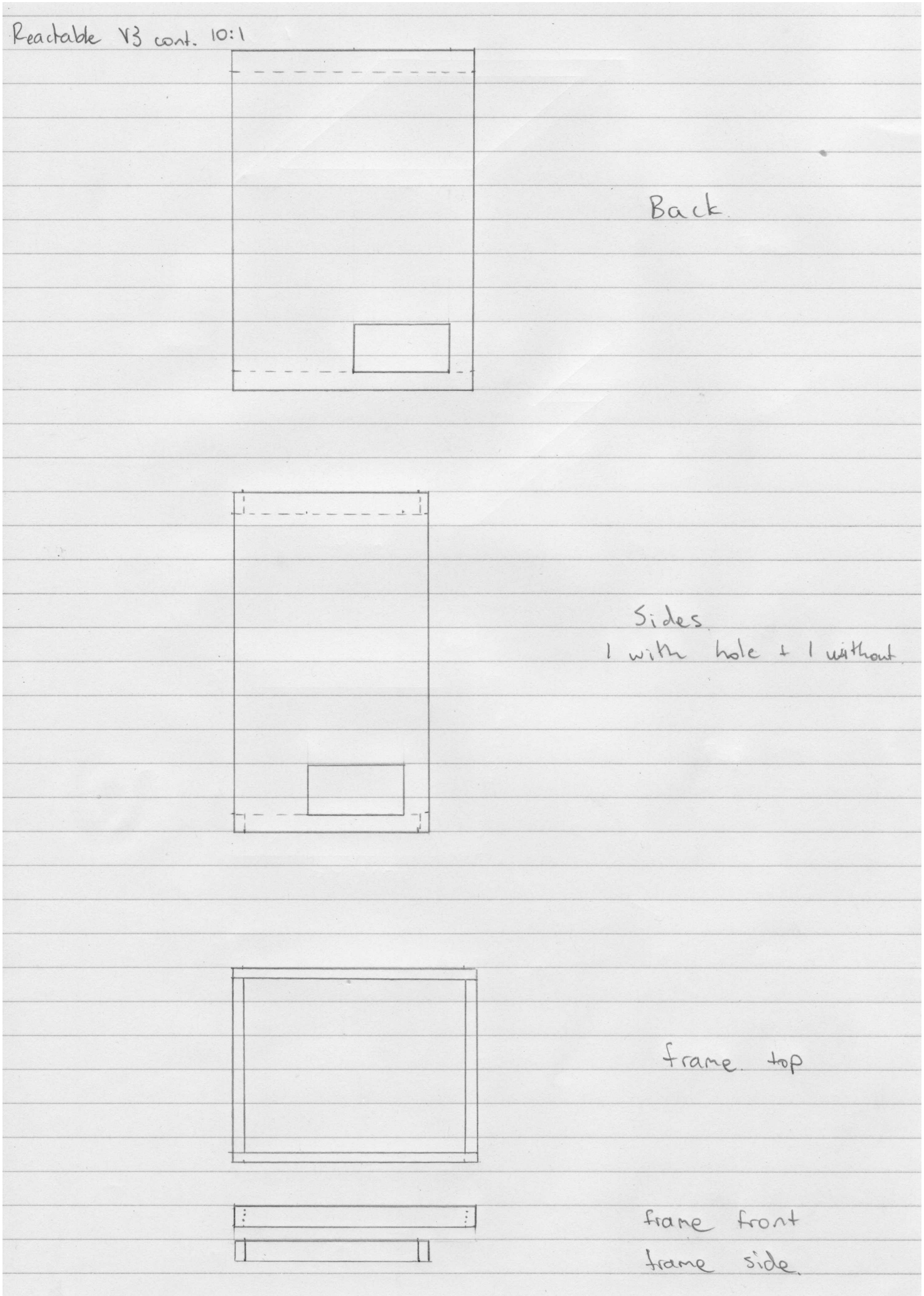
Roth, T. (2008, June 9) DSI – Diffused Surface Illumination. Message posted to http://iad.projects.zhdk.ch/multitouch/?p=90

Sandler, Seth. (2008). Audio Touch. Retrieved October 9, 2009, from http://sethsandler.com/audiotouch/

Zamrod. (2009). ReacTable Role Gaming. Retrieved October 8, 2009, from http://nuigroup.com/forums/viewthread/4177/

Appendix A


Readable V3. 10:1
Iso.
70cm.
40.8cm
50.8cm
5
5
40.8cm.
Top.
50.8cm.
Front.


Reactable V3 cont. 10:1
Back
Sides
1 with hole + 1 without
frame top
frame front
frame side

Appendix B

How to use the software and patches

1. Install Pure Data for the operating system you are running.
2. Open Reactable Data/Platform Independent/Reactable Simulator/Reactable Simulator.jar.
3. Open Final Reactable Patch/Reactable.pd
4. Use the Reactable simulator to control Pure data and make sound by moving the symbols onto the performance space.
5. Also included is the reacTIVision software version 1.4 for both PC and Mac. This is only necessary if using a physical symbols to manipulate the sound. The symbols can be found in reacTIVision 1.4/symbols/default.pdf in either Windows or OSX folders.

List of fiducial id's and respective implementation

0 = On Off
1 = Sine wave
2 = LFO
3 = Square wave
4 = Granular Synth symbol 1A
5 = Granular Synth symbol 1B
6 = Granular Synth symbol 2A
7 = Granular Synth symbol 2B
8 = Drum Loop
9 = 6-Note 1
10 = 6-Note 2
11 = 6-Note 3
12 = is not used.
13 = Sawtooth Wave
14 = Square Wave (Without large jump in frequency at 0/360 degrees)
15 = Sample Player 1
16 = Sample Player 2
17 = Sample Player 3
18 = Reverb
19 = Open
20 = Close

Notes regarding the usage of the symbols.

- To use the granular synths, both symbols must be on the table.
- To change the drum loop take any loop, name it Drums.wav and place it in the folder with the pd patch.
- Sample player 1 is currently a timer that plays 15_1.wav 1.5 minutes after being placed on the table, 15_2.wav 3 minutes after being placed on the table and 15_3.wav 4.5 minutes after being placed on the table.

- Sample Player 2 plays 16.wav. This file can be change as long as the file replacing it is named 16.wav.
- Sample Player 3 plays 17.wav, and can be changed in the same way as Sample player 2.
- The reverb effect is setup so that it can multiply itself allowing some interesting effects. However this does mean that if it is left multiplying itself for too long the audio system will stop generating sound. To get audio back, just rotate the tangible clockwise until you get sound back.