UNIVERSITY OF YORK

Development of a Game Controller Interface for Soundgraphs

by

Thomas Smith

A report submitted in partial fulfillment for the degree of
MSc Information Technology

in the
Department of Computer Science

Number of words = 18,136 as counted by Microsoft Word count command. This includes all of the body of the report, excluding the Appendices and Bibliography

September 2014

Abstract

Soundgraphs have been researched as a way of communicating mathematical information to blind people for nearly 30 years. During this time a number of different user interfaces of varying levels of complexity have been suggested and tested. However, none have really taken hold. The aim of this project is to implement a user interface on top of a previously created soundgraph representation.

Game controllers have been suggested as the primary input to the interface. This is due to the continuing development in the games industry leading to ever better, more easily used, and more adaptable controllers. In particular, motion sensing controllers such as the Microsoft Kinect and Nintendo Wii remote have showed promise in other assistive technologies. There will also be an evaluation of the interface undertaken to identify areas for future development.

Ethics Statement

This project follows the ethical principles of the Computer Science department.

Do No Harm: No participants in the project were required to take part in any activity that could cause them harm.

Informed Consent: All participants were required to read a form which gave information regarding the evaluation they were about to undertake. The form explicitly asked them to sign it to confirm their consent to take part. They were made aware of their right to stop or withdraw from the study at any point. Participants were advised that their data will be treated with confidentiality and used only for the purposes of this study.

Confidentiality of Data: Minimal personal information was taken from participants and no identifying information was gathered or used at any point. All data gathered was kept securely and treated confidentially. No information from this study will be used for any other purposes.

Acknowledgements

Firstly I would like to extend my gratitude to my partner Clare and daughter Emie. They have been both a support and an inspiration for me throughout the duration of this course and I thank them for their patience. They are hopefully looking forward to having their partner/daddy back.

My thanks go to Alistair Edwards, my supervisor, without whom this project would not have occurred. I am very grateful for his guidance and expertise.

Thanks also to Killian Murphy who provided the foundation upon which I was able to construct this project.

Finally, if anyone should wish to take this project and develop it further, thank you for continuing the line; I wish you the best.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This is an extension and revision of a BEng Computer Science project, "Design, Implementation, and Evaluation of Discrete-Time Soundgraphs," which was completed in June 2014.

1.1 What is a Soundgraph?

A soundgraph is a way displaying mathematical data which allows blind and partially sighted people to access maths more easily. Due to the visual nature of maths representations, the visually impaired are at a general disadvantage when compared to sighted people. Using soundgraphs, the curve of a line graph is rendered as an audible pitch which rises and falls with the values of the y axis. Changes in values of the x axis are normally represented by passing time.

1.2 Continuous vs. Discrete Representation

As discussed in the previous project, most soundgraphs display data as a constant tone which closely resembles the image of a line upon a graph. It was argued that a continuous-time representation such as this doesn't allow for an intuitive grasp of a function's shape. Since no attempts had been made previously to implement a non-continuous representation, the author suggested that taking this different approach may have a better outcome. In a discrete-time soundgraph, each data point is displayed as a distinct sound rather than displaying the data range and trend as a continuous tone.

1.3 Project Aims

Since the audio display of the graph was the main drive of the previous project, it's user interface was necessarily rudimentary. As such this project can be considered as a second iteration of a spiral development process developing this application. It is my aim to implement a user interface for this program, ideally one that is more easily used by people with visual impairments. To better achieve this aim, I intend to take the primary user input from a games console controller.

1.4 Why Games Controllers?

The main impetus behind the design of games controllers is to minimise the separation between player and game, and therefore make more enjoyable games which then attract more and more players. Moving towards this goal has involved large amounts of research into a user's interaction with the controller: how comfortably it sits within the hand, how easily buttons and joysticks fall beneath the fingers, how easy are they to push or move, and how quickly a new user can use the input at the necessary level. Ergonomic factors such as the range of movement involved in any specific control motion are well studied and the outcomes included in the final designs. The designs are then tested in focus groups who provide feedback and the design cycle starts anew. The aspects of usability and learnability are also very important when considering interfaces for blind people. Controls which fall easily beneath the fingers and are quick to learn will be vital if the user cannot see the controller. Also, due to the minimal number of computer games for blind people, the intended users may never have used a games controller before.

1.5 Implemented interface

The intended primary input is a Nintendo Wii Remote, also known as a Wiimote. The Wiimote was chosen since it is inexpensive, easily connectable through Bluetooth, and several open-source software interfaces already exist. It also has a history of use in assistive technologies and rehabilitation. Although the Wii was initially very popular, it has now fallen mainly into disuse. This hopefully means that many test subjects will be quite unfamiliar with the interface and its ease of use from initial pick up can be inferred. The commands to be implemented are deliberately limited since this is a proof of concept and ease of use/shallow learning curve will be crucial for evaluation. All controls are to be implemented using data from accelerometers since this would ideally be the major input. Users are able to freely browse through the x axis of a graph while the y value is displayed for every data point that they traverse. The system has a great potential for "zooming" the graph by increasing or decreasing the range of the x axis and/or altering the resolution by increasing or decreasing the number of values calculated within the overall range. However this is considered beyond this simple proof and should be explored in later iterations. The ability to select and display graphs from an array will be implemented using the nunchuk, as will the ability to mark an arbitrary data point in the graph as being of interest.

1.6 Evaluation

A small evaluation study will be undertaken which to test and obtain feedback upon the interface's usability in three areas: comprehension of a graph's shape, navigating arrays of graphs, and marking points within a graph. Initially the subject will hear three soundgraphs, selecting one visual graph out of four that matches each one. This section should acclimatise the subject with the interface and hopefully confirm previous results. They will then be presented with arrays of four soundgraphs and select, using the nunchuk controller, which one they think matches a visual graph. The outcome of this question further confirms the general comprehension of the audio while also testing the usability of the interface. Finally, subjects will mark the maximum, central, and minimum values of three soundgraphs, evaluating the ease of the control and perceiving the difference between more similar pitches at these points in the graph.

Chapter 2

Literature Review

2.1 Maths for people with visual impairment

People who suffer from blindness or visual impairment are at a disadvantage when accessing maths. The inability to manipulate formulae on paper or to perceive a function's graph as a whole means they lack two of the most vital tools for its study. Taking the pop culture view of the mathematician, with a blackboard full of scribblings behind him, we can see the vast amount of information that doing complex calculations requires. This amount of information is very difficult, if not impossible to retain without the external memory of the blackboard or a piece of paper with graphs or formulae on [1]. Equally, while braille is able to represent text adequately, it is not an efficient way of representing maths[2]. Normal text is intended to be read linearly while an equation is two dimensional. See fig 2.1 below for an example of a relatively simple equation.

$$a = \sqrt{\frac{x^2 - y}{x}}$$

FIGURE 2.1: Equation expressed as mathematical formula

That equation would be represented in braille as shown in figure 2.2. While this is just about adequate, as equations become more complex standard braille becomes a less and less viable representation method.

$$a = sqrt(((Xsuper2) - y)/x)$$

FIGURE 2.2: Equation expressed in Braille

2.2 Soundgraphs

In 1985, Mansur et. al.[3] published a paper entitled "Sound Graphs: A Numerical Data Analysis Method for the Blind" in which they compared the effectiveness of audio representation of visual graphs against a tactile one. This work is generally considered to be the first description of the concept of a sound graph. They concluded that information regarding the "symmetry, monotonicity, and the slopes of lines could be determined quickly using sound." It was also noted that using sound was much quicker, more convenient and more cost effective when compared to tactile graphs. A continuous sound was played with pitch representing values on the y-axis and time representing the x-axis. It seems that any following work on the subject has continued with this convention.

2.3 Interaction styles

One elusive aspect of these sound graphs is the user interface. In many cases, such as the program "Listen"[4] and it's successors MUSE[5]and MUSEART[6], the interface is not mentioned at all as the programmes themselves are the focus of the paper. Where a control method is mentioned there are a wide range of styles that are adopted.

The program SonicFunction implements an interface with the computer keyboard due to it being a familiar interface for the visually impaired[7]. Controls were mapped for the following elements: output amplitude, x-axis navigation, selecting the 'step size' or resolution of the navigation for either 1/30, 1/10 or 1/6, selectors for each of the default test functions, and setting markers for maxima, minima, $f(x) = 0$, and $x = 0$. These were mapped to the arrow keys and several distinct areas of adjacent keys. Each data point on the graph was represented by a discrete sound which was positioned within the stereo field relative to its distance along the x-axis. Mathgrasp[8], an extension of Mathtalk[9], took gesture based input in three possible ways: first, by using an instrumented suit and glove, second, using a three-space tracker using only the hand, and third a video camera with gesture recognition capabilities. While these each have their unique benefits and drawbacks to offer, they all allow the user to interact with virtual objects in 3D space. Each of these objects represents an element of the function being evaluated, e.g. the objects in the formula $3x = 7 - 4$ are $3x$, 7 and 4. The user makes gestures such as pointing fingers, grabbing or opening the hand, and throwing away to select, read, move, alter, and delete elements of the formula at hand. Stevens et al[10] used a command language to help users navigate through elements of an expression with the aims of: making the reading of formulae an active exercise, keeping reading as the primary experience, minimising awareness of the interaction, and providing an interface that is relatively easy to learn. To achieve these aims they developed a two part command structure with one part indicating the action to undertake and the second identifying the target for that action. Examples of command pairs are "speak expression", "next term", or "current item".

There are a number of ways that have been developed or discussed which can present information regarding functions in a non-visual manner which exploit a persons other senses.

2.4 Haptic Interface

Haptics can be defined as "...relating to the perception and manipulation of objects using the senses of touch and proprioception"[11]. A haptic display relays force and touch feedback to a user through a robotic device which can simulate an object with physical characteristics such as shape, mass, hardness and texture. Such an interface can be valuable in areas such as computer aided design and prototyping and, simulating hostile or delicate environments such as hazardous waste management or complex surgical procedures. Devices such as this can also be used for more abstract purposes. In one such case an oral interface was developed which relayed navigational cues to the roof of the mouth via an electrotactile display and was controllable with a tongue touch keyboard[12]. A number of interfaces to represent mathematical graphs have been developed which influence the movement of the user's hand. Several of these were implemented with the Sensable PHANToM(now Geomagic Touch) which is either a stylus or fingertip cup connected to an motor-controlled, jointed arm with three directions of movement and force. The force feedback can aid visualisation of an object or curve by guiding the users finger or hand as if interacting with a solid representation. Due to the high cost of the PHANToM device, it isn't practical outside of a professional or academic context so other devices have been explored. The Logitech Wingman Force Feedback Mouse has been suggested as a more affordable interface for navigating virtual graphs with some success. However the mouse does suffer from some issues such as a small workspace, no third dimension and, limited amount of force feedback compared to other devices[13]. Since this kind of interface simulates only a single point of contact with the virtual object, its feedback is very limited when compared to the number of haptic receptors in a human body[14]. However it should be adequate for understanding the overall curve of a graph.

2.5 Gestural Interface

Gestures consist of non-verbal movements of the face and body and can both complement and replace verbal communication. Since gesture is so vital a part of everyday communication it makes sense to attempt to base human-computer interfaces around it.

Gestures can be split broadly into three categories: mimetic, deictic, or abitrary[15]. A mimetic gesture is representative of the characteristics of an object. For example, describing the shape of a sphere or the height of a person with your hands. Deictic provide contextual information which supports other communication. Indicating the object of a conversation or pointing to describe the position of something out of sight are both examples of this. Finally, arbitrary gestures are learned motions generally used in the place of other communication such as semaphore code or the hand signals used on baseball pitches[16]. Ideally the input to a gestural interface should use a majority of mimetic or deictic gestures. Because, although arbitrary gestures are very useful for passing complex information simply, they delay a user's effective use of an interface.

The gestures that are read for input can come from a very wide variety of sources. One example is the physicist Stephen Hawking's interface with his speech generator. It is controlled by the twitching of a single muscle in his cheek which is picked up by an infra-red sensor mounted on his glasses[17]. Other interfaces such as the Nintendo Wii remote or Microsoft Kinect have a much larger area of detection but lack the delicate detection of that sensor. The

A gestural interface can be used to make a virtual reality more immersive or to aid people with disabilities to overcome communication difficulties that they may have. As we use gestures in our everyday communication they are very easy to adapt to other uses. It does however take a long time to develop a gestural interface as the movements to be used need to be decided upon and refined from all the possible movements of several parts of the body. The level of exactness required of the user needs to be carefully measured to ensure it is within the reach of all users. Also, it is vital that ergonomics are taken into account. Holding hands in rigid positions for a long time or overextending joints can cause discomfort and damage to a user.

2.6 Olfactory/Gustatorial Interfaces

Olfactory and Gustatorial interfaces target the senses of smell and taste respectively and are some of the least developed areas of study in human-computer interfaces.

2.6.1 Olfactory Interface

Using smell to affect users has met with some success. For example, it has been demonstrated that a subjects exposed to a peppermint odour performed better at spatial visualisation tasks than those exposed to a lavender scent or no scent at all[18]. There are still may unanswered questions regarding olfactory display: which smells to use, how to synthesise and store them, and how to control atmosphere surrounding the subject. Suggested delivery systems have ranged from a sealed room with air filtration, to a mask which includes all filtration and synthesis equipment within it. Each different interface also suggests a different approach to the display: it is likely that a large room won't be able to change its output with great frequency so it could only display some general factor which applies to a large set of the displayed data. If you were to use a mask it's possible that the period between different scents could be brief, in which case more frequently changing information could be displayed. Since this is a novel way of communicating data, it is likely that it would require a period of practice before it could be used to its full potential.

2.6.2 Gustatory Interface

Using taste as an interface is difficult in part because of the nature of taste itself. There are only a few distinct taste sensations, and the texture of the material is also an important element of its taste[11]. The main aim for people studying gustatory interfaces seems to be virtual reality with little or no interest in using it to display data.

2.7 Drawbacks

2.7.1 Lack of test subjects

As noted by Edwards et al[19] the issue that these programs try to address, i.e. the difficulties that face people with visual impairments with regards to learning maths, frequently mean that there are, at most, a limited number of mathematically literate, visually impaired subjects available. In fact in most cases sighted people were used to evaluate the projects.

2.7.2 Soundgraph Limitations

Sound graphs have some drawbacks as a display tool. For example they are not able to satisfactorily convey the data from graphs that are not "time dependent" such as pie charts[20] and purely "musical" graphs have no way to express non-data information such as the labels of the axes.

2.8 Game Controllers

Arguably the first games controller consisted of a knob and button attached to an oscillator screen on which the game "Tennis for Two" could be played to keep visitors to the Brookhaven Lab in New York entertained[21]. Following from that point game controllers have continued to be developed and improved. From the first appearance of joypads in the 80s there has been almost an arms race between the leading console makers. The developments in the ergonomic design of the shape of the controllers, the position and action of the buttons and triggers and the haptic feedback from motors within have been constantly improving since then[22]. Motion sensing was introduced into the game controller world by Mattel with the Power Glove, released in 1989[23]. Since that point motion sensing has continued to be put into games controllers, which have generally been poorly received. This changed with the release of the WIimote, soon followed by The Playstation Move from Sony and XBox Kinect from Microsoft.

The controller most used for non-game interfaces outside the platform it was made for is Nintendo's Wii remote. Since its release in 2006, Nintendo's Wii remote (or Wiimote), the controller for the Wii console has been the subject of a great deal of "hacking" from a worldwide community[24]. Due to the simple interface and Bluetooth connectivity of the Wiimote it has been used for a number of unintended but very interesting applications. The accelerometer data from the Wiimote was used as part of introductory physics experiments, measuring harmonic motion from an oscillating spring[25]. It has also been used to control a HOAP-2 humanoid robot in robotics researech at The Jožef Stefan Institute[26].

Of particular interest to this project is the use of the Wii remote in assistive technologies. It has applications in physical rehabilitation following stroke or injury. The measurement of motion it provides can aid in both the detection of impairments in joint flexion and extension, or general gait and in their improvement[27][28]. Another area in which the Wiimote is providing a vital interface is in technologies for the blind and visually impaired. It was found during research into accessible interfaces and games that the game Wii Tennis was both accesible and enjoyable to two blind players[29]. In a more innovative development, a Wii remote is being investigated as a possible cane/guide dog style assistant for a blind person.

Chapter 3

Design

3.1 Initial specification

These goals are based on the project specification and the controls implemented by the SonicFunction team[7].

- The user should be able to browse back and forth through the graph at their own will and pace.
- There should be a button that when pressed, will cause the currently selected values of x and y to be spoken.
- Another button will be available to mute the audio output.
- The resolution, i.e. the number of data points calculated over the range of the graph, should be alterable to allow a deeper look at the data being displayed.
- The range of the graph, the distance between the highest and lowest values of x that are displayed should also be alterable so a user can “zoom out” and experience the wider overview of the function.
- The ability to navigate between test functions needs to be implemented so graphs can be browsed consecutively for comparison.
- You should also be able to select two graphs and browse them concurrently to identify points of convergence.
- When browsing concurrently, the graphs should be displaced across the stereo field, introducing spatialisation to help differentiate between them.
- A change in timbre will also enable identification of a specific graph.

- A user should be able to mark data points such as maximum and minimum values, crossings of axes, convergent values, or any arbitrary point of interest.

3.2 Interface design

3.2.1 Controller selection

The Wiimote was the final choice for the control input. There were several reasons for this:

- Ease of use: when it was released Nintendo's Wii was a departure from the normal gaming console market. Instead of keeping up with Sony and Microsoft with graphics and processor improvements it targeted "casual gamers", a largely untapped market consisting mainly of older adults and younger children. As such the controller interface was designed to have the shortest learning curve possible.
- Wireless connectivity/portability: The bluetooth connectivity of the Wiimote means that the user is unencumbered by wires during use so that there are far fewer wires to be trip hazards or confusing influences. Being small, the Wiimote can be carried with a user and simply connected to another device at the end of the journey.
- Affordability: Since many people do not own gaming consoles it makes sense to use an inexpensive device. A Wii remote and nunchuk can be purchased for £8.99 from Amazon.co.uk (Another controller input option for this project, the Microsoft Kinect, costs over £88.)
- Accelerometer/infra-red sensor input: The data on the orientation of the Wiimote is gathered from a three axis accelerometer and infrared camera. The camera can trace up to four infrared light-sources from which the distance from the sensors and movement in the horizontal axis, which the accelerometers can't provide, can be detected. The nunchuk, one of the peripheral input devices for the Wiimote, also provides accelerometer data across two axes. This has the benefit that any required or commonly used functions can be controlled through the motion of the Wiimote and nunchuk, freeing the user from learning the positions of any buttons save those which fall naturally beneath their fingers.
- Ergonomically placed buttons: When holding the Wiimote there are two buttons that naturally fall under fingers. A trigger underneath corresponds to the position a user's index finger would be when holding the device. A button on top does

likewise with the thumb. The nunchuk has two buttons on the front that are placed where the index and middle fingers would naturally lie. An analogue thumbstick on top of the device completes the easily accessible controls.

- Wiimote already used for many "off-label" purposes: There is a large community presence online of "Wii-hackers", people who use the Wiimote as an input device for many extra uses. These extra uses start with "Wii-habilitation" projects where existing Wii software is used in the physical rehabilitation to implementing immersive 3D environments using the relative position detection of the Wiimote[30].

- Previous experience with a Wiimote interface: My undergraduate project used a Wiimote to control a software instrument meant for use in sound therapy. Its main aim was to assist people with movement disorders or general motor control issues to gain a sense of agency of their movements and then work towards improving their overall control.

3.2.2 Controller issues

There are a fairly small number of control inputs on the Wiimote that were identified as potential inputs for this application. This is complicated further by the fact that accelerometer data can be unclear or noisy, e.g. unless the user is capable of moving the Wiimote in only one axis while the others remain constant, they may trigger unwanted control events from the second axis or third axis during movement. Control signals from the thumbstick on the nunchuk would also suffer from this noisy data output. This is a problem that would likely diminish or disappear completely with greater experience of the interface. However, subjects would not get the benefit of that extra time and practice before taking part in this study. One possible solution was chorded input, where each button or motion has its own function, but that is then altered when it is pressed concurrently with another. An example of this kind of input is the stenotype typewriter used by court reporters. Chorded input allows for a larger number of controls but also increases the complexity and therefore learnability of the interface. This is in conflict with the aims of the project. The remaining option was to limit the implemented controls to those which are most vital to the project

3.2.3 Change of focus

The initial specification of the project involved adapting a number of areas of the soundgraph program. In order to preserve ease of use some of these elements had to be removed from the interface. Since it was the user interface that was the main target of this project, more abstract functions that manipulate data were of lesser importance. Therefore, it was the ability to alter the range and resolution of data, and the concurrent browsing of graphs that were deemed outside the scope of the project. While these functions are undoubtedly important to the program as a whole they introduce additional complications to the implementation that would overrun the available time. Altering the range and resolution of the data are theoretically infinite and run into problems relating to the user and the interface. The number of data points in a graph are increased by both actions and If the number of points becomes too large then the limitations of the ability of the human ear to distinguish between different pitches becomes an obstacle. In theory it is possible to mitigate this to a point by extending the range of motion of the arm, however this is limited by the hardware since the Wiimote can only detect the sensor bar in an arc smaller than 90 degrees.

3.2.4 Revised specification

Following the re-considerations of the interface as noted above, a new specification was produced.

- The user should be able to browse back and forth through the graph at their own will and pace.
- There should be a button that when pressed, will cause the currently selected values of x and y to be spoken.
- Another button will be available to mute the audio output.
- The ability to navigate between test functions needs to be implemented so graphs can be browsed consecutively for comparison.
- A user should be able to mark data points such as maximum and minimum values, crossings of axes, convergent values, or any arbitrary point of interest.

3.3 Graph Manipulation

The user inputs that were mapped to each control function were determined most strongly by that functions frequency of use. Since one of the aims if the project was removing barriers between the user and the maths, physical actions were favoured for interacting directly with the graphs. For example, the fundamental thing that the programme would do is to browse back and forth through a function graph. Therefore, it was movement of the Wiimote through the horizontal plane which was selected to control for that function. The sweeping motion of the arm in front of the body resembles the motion of a virtual cursor along the x-axis of a graph. Moving in between graphs and marking selected points are also "direct" manipulation of the graphs, so these were assigned to horizontal and forward movements from the nunchuk, respectively. The horizontal motion of the left hand suggests moving real-world objects back and forth while the forward action is similar to that of indicating something by pointing to it.

3.4 Audio control

Following the revision of the specifications, the only remaining controls to implement were those controlling the audio. As these were interacting with more abstract concepts they weren't seen to require the physical motion link which the interface with the graphs had. The other option then, was to link them to buttons. As they both related to sonification of the data they were implemented to be controlled by the right hand. Since the right hand which holds the Wiimote was responsible for the audio output from the graph, these additional controls were added as they fit in with the overall use of that hand. The mute/unmute function was assigned by the trigger which corresponds to the index finger. Speaking the values of x and y was controlled by the "A" button which naturally sits under that last joint of the user's right thumb.

3.5 Sonification design

Consideration of the audio output of the graph data was split down three courses. First should the audio be "musical"? That is should the audio frequency of the output be such that the pitch of all sounds output are consonant with others? The second is the discussion of timbre. The initial specification demanded a complex timbre for the output. However that was intended to be applied to a continuous representation of the graph data. Thirdly, the stereo positioning of the sounds was to be considered.

3.5.1 Harmony

The question of harmony came basically down to the number of data points which can be displayed. The intended range of sound display was from 250 Hz to 1000 Hz. An octave distance between pitches is equal to a doubling or halving of the frequency of the sound. Therefore, a frequency range of 250 Hz to 100 Hz is only two octaves. Even if you include every semitone in the two octaves, you are limiting yourself to displaying a maximum of 24 data points. On the other hand, it was shown under clinical conditions that a human ear can detect pitch changes over 3.6Hz[13.1]. That would give a maximum number of 208 discernable pitches for display. Although that number would not be as large in a real world setting it would still be a improvement of many times of the tonal representation.

3.5.2 Timbre

Since the sounds being output were no longer going to be continuous pitches, aural fatigue was no longer an issue. The fact that the program was not required to display more that one set of data concurrently removed the requirement to set several different timbres for distinct data sets. Taking these facts into account, a pure sine wave oscillator output was adequate for purpose.

3.5.3 Volume Envelope

The vital elements to take into account regarding shape of each sound were maintaining the sense of having a pitch and being separately audible when browsed through quickly. If a note is played too fast, then the whole period of the waveform may not be output, in which case it takes on a clicking noise with no definitive pitch. After experimentation, volume attack and decay periods were each set at 15 milliseconds. That duration solved both the issues of maintain pitch information and sounds remaining separate. The relative maximum amplitude of the output was set to 50%, therefore allowing some crossover of sounds without causing the output be distorted by an output amplitude above the maximum.

3.5.4 Spatialisation

The effect of spatialisation on the sound output would be of most use when concurrently displaying more than one set of data. Then it would complement the timbral change that would be implemented if concurrent display were being implemented. Since concurrent display is no longer being implemented in this project, spatialisation will not be included.

3.6 Visual design

Similarly to the control interface in the project which preceded this, the visual interface wasn't implemented due to being outside the range of the brief. It would also have been of little use to the functions included in this implementation.

Chapter 4

Implementation

4.1 Selecting implementation language

Before beginning to code, I needed to evaluate the relative benefits of two different programming languages. C++ and Max/MSP.

4.1.1 C++

C++ is a 3rd generation, object-oriented programming language. [MORE INFO] The project I was inheriting was coded in C++, a language in which I had no experience. Although I do already have experience of object-oriented programming using Java. Balancing this was the fact that all the implementation for the sonification of the graphs had already been completed. Coding in C++ would also make it easier to pass the project forward again for more development. However, the sonification element of the previous project did not work very well which seemed to be due to the audio API used. The time spent in coding the synthesis module would almost certainly take more time than implementing one in Max/MSP.

4.1.2 Max/MSP

The language Max is a visually based programming language that is mainly intended for use in multimedia generation and processing. It has also been widely used in research since the interface is quite easy to use for those with little or no coding experience. Max will interface quite easily with a Wiimote and is made for simple execution of audio output. [MORE INFO] I have more programming experience with Max/MSP than any other language, having completed my undergraduate program using it.

4.1.3 Choice of language

When considering this project as part of a spiral development process it made sense to utilise work already done and leave a program for the next developer that is more easily extensible. With that in mind C++ was the finally selected language for the implementation. Unfortunately, initiating the coding process was delayed by my own inexperience with C++. This manifested as complication both in the further development of the project and reverse engineering of the existing code. Following two weeks work with very little usable outcome, it seemed sensible to move to a different language for implementation. All the reasons for not working in Max/MSP still held so I selected Java to continue the project. Java has many similar elements and abilities to C++ and so would be able to implement all the functions that the existing application did.

4.2 Creating the program

4.2.1 Process

As mentioned, I approached the project with a test-driven, agile process[31]. As such I implemented elements of the program incrementally, completing one element to a usable level before moving on. For example, I created a method to take the output from the detection of the sensor bar and turn it into a usable stream of numbers before then implementing any audio synthesis using this data as an input. Adopting an agile approach meant that I was able to react quickly to any changes necessary in the program and was able to have working software ready to be evaluated at small increments during the project[31].

4.2.2 Interface with Wiimote

In order to get controller information from the Wiimote I used a third party API called WiiuseJ [32]. This appeared to be the most up to date API for the Wiimote even though the last update seemed to have been in Oct. 2008. Interest in the Wiimote as an alternative input having seems to have waned as the console itself became older and less popular. While this may be the case, it does not adversely affect the usefulness of the controller for this application. WiiuseJ is able to interface with the Wiimote and many of the peripherals that connect with it. This list included the Nunchuk add-on which was a vital part of my design. The following items are implemented in the API:

- Get accelerometer events on different forms (Orientation, GForce and raw Acceleration)
- Get events from infrared Camera
- Get button events
- Make the wiimote vibrate
- Set lights of each wiimotes
- Get events from expansions

Notifications are for given the start and end of an event, e.g. a button being depressed then released. If an event continued, e.g. if the button remained depressed then that would be communicated as well. All these controls have the ability to be useful in the future of the project as more controls need to be implemented. The api is also able to give feedback as it can control the vibration of the Wiimote. Although the vibration output was not used in this project it could have a wide variety of uses in future developments. The inclusion of haptic feedback to enhance the audio could be very useful addition for the future. Of course getting events from expansions is vital as the nunchuk is an intrinsic part of the design.

4.2.3 Implementing Controls

Due to time constraints following the delay caused by changing programming languages and the changes to the initial specification as noted above there were four control inputs that were implemented.

- Browsing back and forth through a specific sound graph.
- Muting and enabling audio output.
- Moving focus between a number of graphs held in an array.
- Marking an arbitrary point in a graph.

4.2.4 Browsing a graph

The allocation of the controls matched the design. In order to browse between the graphs, numerical data was generated from the lateral motion from the Wiimote as detected by the infrared camera. This was processed by filtering and rounding the numbers to provide a steady, stepped output rather than the constant stream that was generated otherwise. The continuous data from the Wiimote would have been better suited to the continuous graph representation that was discussed previously. For this discrete representation of the data, a number of concrete, constant values was more useful. The difference required between the generated values before the next sound was output was scalable. That is, the number of trigger events from the lateral movement required to trigger an output could be altered depending on the user's need. This is to lay foundations for further iterations of the program where the resolution of the graph data can be increased to include more data points for a set range of values.

4.2.5 Controlling audio output

The muting and enabling of data was implemented with the trigger underneath the Wiimote which lies beneath the index finger of the right hand. This is the button that sits the most definitely within the range of the finger. The button on the top is under the last joint on the thumb in my case but will vary from person to person while the squeezing motion for the trigger will be more or less universal. Other options for implementing this command were the two buttons on the nunchuk that are in a trigger-like position similar to that of the button on the underneath of the Wiimote. However, in the design each hand was very definitely matched to a specific target. The right hand was to interact with the sonification of the graph whereas the left was to control the more abstract elements of moving between graphs and marking data points. There was some uncertainty as to how the audio would be switched. There were two options considered to activate and deactivate the audio output. The first was a binary on-off switch similar to a light-switch. This had the benefit of requiring no additional thought or activity once it was pressed, the state of the switch would remain constant no matter what. Secondly, the control could have been implemented as a sort of reverse dead-man's switch. In which case, the audio output would be active when and only when the switch was depressed. The benefit of this switch would be to more easily allow quick switching on or off of output if there were any case where quick switching would have been required. For the use of this evaluation of interface the binary switch seemed most suitable. There was no identified need for the user to quickly switch audio on or off. The binary switch had the additional benefit that it required no thought once switched on, allowing the user to concentrate on the tasks set for the evaluation.

4.2.6 Shifting between graphs

The controls for the nunchuk were implemented purely using positional data. Initially I tried to link this control to raw acceleration left or right in order to represent an action such as turning pages in a book. However, while this action initially creates acceleration in the required direction, it is quickly followed by corresponding and opposing acceleration as the hand stops moving. A possible solution to this would be to make the input ignore any opposite movement that quickly follows an acceleration event. Doing this though, would require a lot of calibration and is subject to a number of events that could render it faulty. If a user should decelerate their hand slowly enough after a movement then their next move may be ignored. While this may be only a minor annoyance, it is certainly enough to break a user's concentration. So, due to time constraints following the language issues noted above, the command was implemented in a different way. The orientation of the nunchuk was represented by a number that was constant for any particular pitch or roll position of the nunchuk. Therefore, a constant value could be set that represented a specific deviation from an upright position. Using this the movement between graphs was controlled by a rolling movement from the wrist in the direction of movement. Since the value of any position was constant then a constant threshold could be set, over which the selected graph would be moved in the direction of the roll. Due to this, acceleration in any direction had no effect on the control and so movement in either direction did not cause interference. The rotation of the wrist has less of a relationship with a concrete action than the initial control but that shouldn't cause an obstacle to users.

4.2.7 Marking a data point

Marking a data point in a graph was controlled by a forward movement similar to the sideways action ultimately used to move between graphs. This shares all the positive elements of the pitch control used in order to move between graphs. It also has a more of a relationship with a real world action, the motion used to point to or indicate something.

4.2.8 Wii interface API delays

WiiuseJ is an unsupported program since it is created by just one person during spare time. Due to this there is little documentation other than code examples for each method. This caused further delays in the project since it was necessary for me to dig deeply through the code examples to find the parts that I wished to use. Additionally, the nunchuk interface had been written separately to the original Wiimote one and had a different ways of dealing with incoming events which I wasn't aware of initially and spent a lot of time searching for.

4.3 Audio synthesis

Java is capable of generating MIDI control signals, but not synthesize audio. MIDI is the open technical standard "Musical Instrument Digital Interface", it was created to allow digital instruments from different manufacturers to communicate in a standard way [33]. Java does include a module called synthesizer, however it has a number of inflexible preset instruments that are unchangeable. Therefore, if this application were to use the MIDI output from Java, an external synthesiser, either software or hardware, would be required. I felt that it was important to have more fine control over the sound output so it could be fine-tuned to output audio that fit the purpose of the instrument. Therefore, if this application were to use the MIDI output from Java, an external synthesiser, either software or hardware, would be required.

4.3.1 JSyn

The program JSyn is an "Audio Synthesis Software API for Java" from softsynth.com. It allows real-time audio synthesis using a library of oscillators, filters and envelope generators. It was selected because it achieves the necessary output with the minimum of extraneous functions. The basic oscillators included are the sine, triangle, square and saw waves. From the simple pure tone of the sine wave, the other outputs become increasingly timbrally complex as additional harmonic frequencies are added to initial wave. These can then be combined in various rations to one another to create more complex timbres. The filters and envelopes included are able to influence the harmonic content and "shape" of the output tones respectively. Using this tool it was possible to create the required sounds for the output of graph data.

4.3.2 The sonic display

As noted in the design, complex timbres weren't required for this discrete representation of a Soundgraph. This was due to the sound output not being constantly present enough for a listener's ears to become fatigued. Since only one graph was to be displayed at any one time, there was no need to implement different timbres for separating concurrent displays of data. For these reasons a simple sine wave generator was used to create the output sounds. This meant that it was also not necessary to use an audio filter as there was no additional harmonic content in the output to alter. By avoiding use a filter, it was possible to make the code a little simpler which may help future developers of the software. The envelope generator created a "shape" for the output amplitude of the sound. The attack and decay times were set to 15 milliseconds with no gap in the middle. One of the reasons for the brevity of the sound was to avoid sounds overlapping. There was however, still some chance that an overlap could occur. To mitigate his risk, the output amplitude was set to 0.45 to avoid distortion if two sounds did play together. Another reason for having such short pulses of sound was to maintain separation between data points, even when the user was moving through it very quickly.

Chapter 5

Evaluation

The evaluation of the program was undertaken with a small study of 11 sighted adults aged between 24 and 62 years. It was unfortunately not possible to find any blind people to take part. Demographic information regarding the participants' age, gender, hearing ability, highest level maths qualification, and frequency of Wii use was captured before test began. Put simply, the test involved listening to, and browsing several audio datasets, then using the Wiimote and nunchuk to perform a number of tasks relating to the data. The test was arranged into three sections, each testing a different part of a subject's interaction with the interface. These test were then followed by a multiple choice questionnaire to get participants opinion on the interface itself. Participants were not given any time to prepare or practice with the interface before taking part in the study.

5.1 Study design

Some previous studies into effectiveness of soundgraphs have had subjects draw the graphs that they hear. There is a benefit to this as it requires the subject to pay close attention to the sounds in order to create a mental image that they can then put into a drawing. However it can lead to ambiguous or subjective results. It is possible that a researcher, not being removed from the project, may unintentionally interpret a graph incorrectly if it is unclear. In order to avoid bias and give unambiguous, concrete results, subjects were given images of graphs to choose between at points that it was necessary. Almost all of the data gathered from the study is quantitative in nature, to allow better measurement of results and outcomes. Using quantitative data also avoid any of the ambiguities mentioned above. I did however, also include a comments box at the end of the final questionnaire to capture any qualitative information that the subjects wished to give.

5.2 Study Section One

One objective of this initial section was to allow subjects to acclimatise somewhat to the browsing action of the Wiimote and the experience of the audio display. It also went over older research regarding a subject's ability to discern the shape of a dataset when portrayed as sound. In this part of the study subjects had one soundgraph available to browse at a time, using the wiimote only. The nunchuk attachment was not introduced at this point. The subject was also presented with a piece of paper on which there were printed four line graphs. They were to choose the image they thought most matched the one they were hearing. In all the subjects select matches for three different soundgraphs and were given a different selection of four visual graphs for each audio one. Of the three incorrect visual images, two somewhat resembled the correct one in overall shape and direction of data while the fourth was very different. The difficulty of identifying the differences between the visual graphs increased as the subject moved through the tests. All the printouts of graphs that were used in the test can be found in appendix one.

5.2.1 Graph One

The audio graph for this section was a simple, linear $y = x$ function. This was to present a simple introduction to the test to make the subject a little more relaxed and ease them into the task. The to roughly similar graphs were $y = x^3$ and $y = \sqrt{x}$. Both follow a rising trend from left to right like $y = x$ and both have a generally uniform increase, although this is a little arguable in the case of $y = x^3$. The fourth visual graph was $y = sin(x)$ which as a repeating wave had no particular common ground with the audio representation.

5.2.2 Graph Two

This section was a little harder than the first with a little more ambiguity between some of the graphs. The audio graph was $y = x^2$ which is number 4 on the multiple choice page in fig 1.2. The most similar graph was graph 3 which is $y = 2x^2$. Obviously this is a very similar shape to $y = x^2$, just with a steeper increase in values either side of zero. This was added to see if a different speed of change could be adequately represented by the soundgraphs by themselves. Graph 1 is a single period of a $y = cos(x)graph$, representing the opposite shape to the correct graph. There is the potential that while a listener is able to hear that the graph has a curved shape, they can be confused as to which direction the values are increasing/decreasing. The unrelated graph in this case is $y = x^3$ which shares none of the important elements such as the curved shape or corresponding values on either side of x=0.

5.2.3 Graph Three

As mentioned above, the difficulty in detecting the differences between the graphs becomes harder as the test progresses. The audio graph is $y = \sqrt{x}$ which corresponds to graph number three of the printed graphs. Graph number one, $y = log(x)$ is the most similar, the angle of the initial curve being the main difference. Graph two, $y = x$ shares the direction of increasing values but has no curve whatsoever in the data. Graph four $y = exp(x)$ is the most dissimilar as the rate of change of values is reversed. It was expected that graphs 1 to 3 would be difficult to tell apart and would test the programs ability to display subtle differences in the rate of change of data.

The visual graphs shown top the participants for this section can be seen in figures 5.1, 5.2, and 5.3.

5.3 Study section two

This section of the test introduced the nunchuk as part of the controller input. The task to be undertaken was opposite to the previous one. The subjects were shown a pictures of a single graph while having an array of four soundgraphs to browse. Subjects were asked to select the soundgraph they thought most resembled the visual they were given. Browsing through a single soundgraph was again implemented with lateral motion of the Wiimote. Moving back and forth between graphs in the array was controlled with the left handed wrist rolling motion as described earlier. Similarly, subjects selected the matching graph by dipping the nunchuk forward. The aim of this section was to test the usability of the whole browsing interface and measure the subjects ability to recognise the shape of the graphs. The graphs in each the array were not as similar as the visual graphs from the first section. It was primarily the usability of the navigation interface that was being evaluated and so the ability to discern between similar shapes was not an important factor.

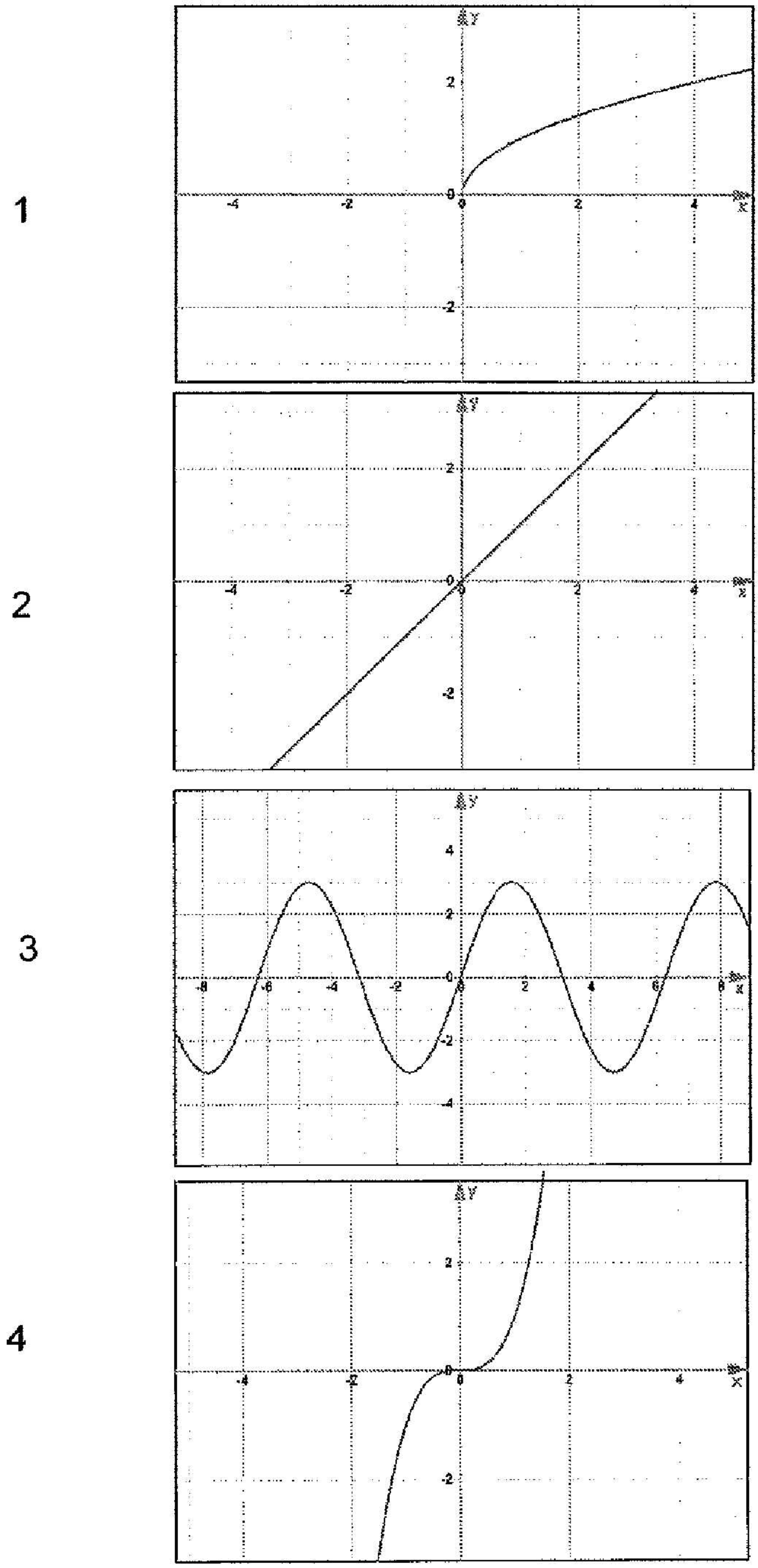


FIGURE 5.1: Graphs for 1.1

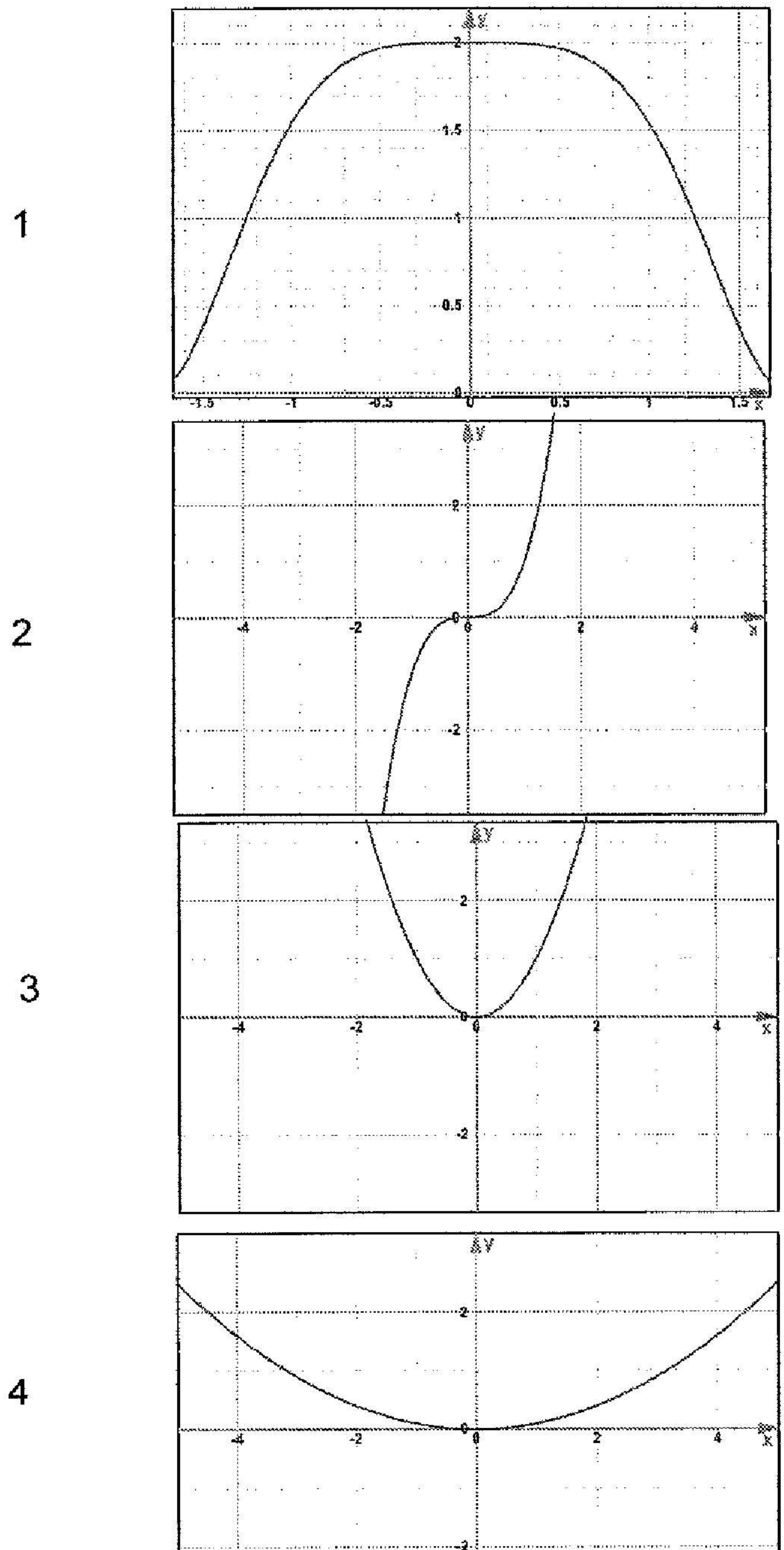
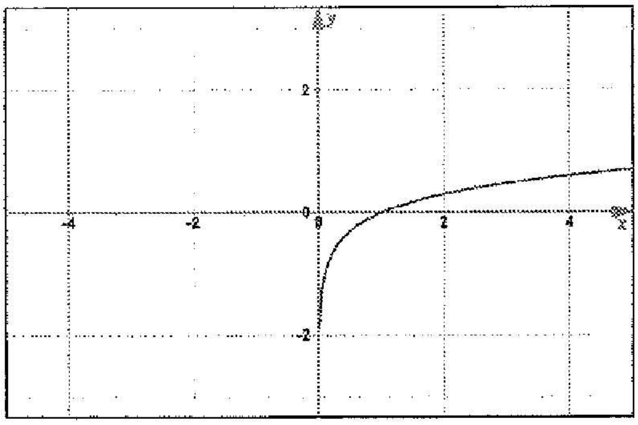
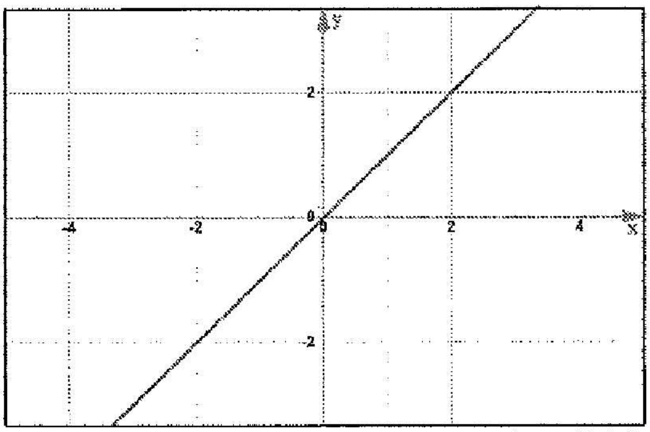


FIGURE 5.2: Graphs for 1.2

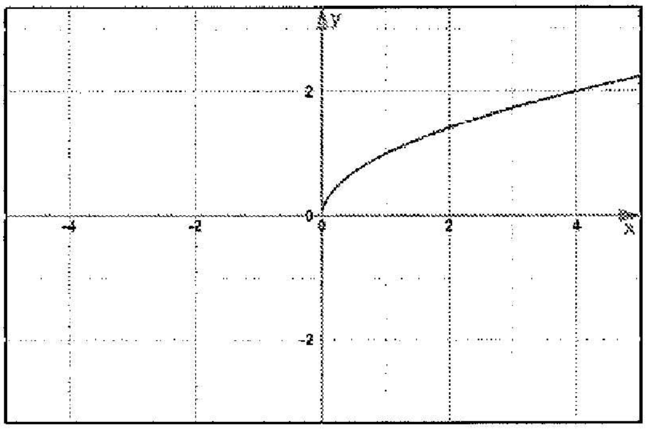
1



2



3



4

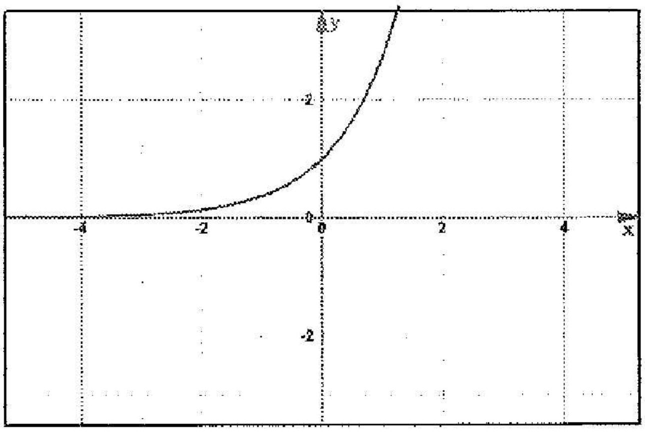


FIGURE 5.3: Graphs for 1.3

5.3.1 Graph One

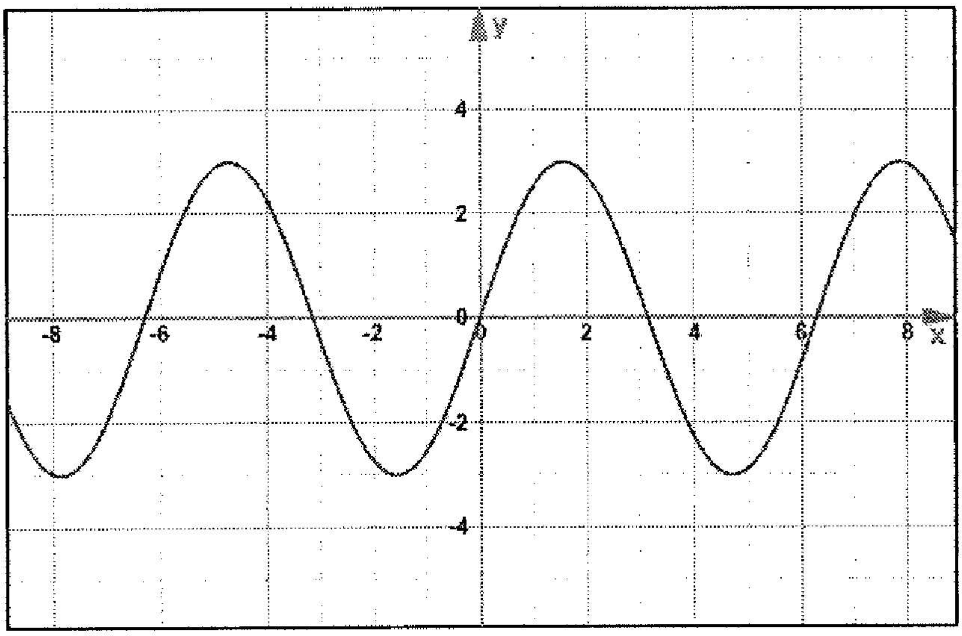
The printed graph was a simple periodic waveform, $y = sin(x)$. The incorrect soundgraphs were: a linear increase in values, $y = x$; a steep curve, $y = \frac{1}{x^2}$; and an inverse curve, $y = x^2$. While the curved graphs had some similarity to a single period of the waveform, the wave was clearly distinguishable since it went through 5 whole periods over the course of the graph.

5.3.2 Graph Two

A linear graph, $y = x$, was the visual graph to match in this part. The other graphs in the audio were $y = exp(x)$, $y = x^3$, and $y = x^3 - 20x$. In this case, all the graphs shared the overall direction of growth of the data but not the rate and consistency of the growth. For example, the final two graphs have either a flattening or reversal of data progression around the crossing with the vertical axis. The exponential graph has a very swift increase in values after the vertical axis crossing.

5.4 Study Section 3

In the final practical element of the study, users were asked to listen to a graph and select either the highest, lowest, or middle value depending on the particular graph. Overall here were three graphs that the subjects were tested upon. In order to select the necessary points this they would use the same marking motion from the nunchuk as was used before. The Wiimote was used to browse through the currently selected soundgraph as before. The graphs were heard sequentially, the next graph in order not being displayed until the previous central point had been chosen. The first graph was the bell curve $y = \frac{1}{x^2}$, the second was $y = x^3$, and the third was $y = x^2$. Participants were asked to highest, middle and lowest values respectively from these graphs. The points to select fell where the graphs were crossing the 0 point on the x axis. Around this value the differences between adjoining values were at their smallest. This section was designed to test the how easily the Wiimote could be used for the sort of fine motion that was required for this task. At the same time the ease of detecting the smallest differences in the pitch of the audio output was being evaluated. In general, it was the aim of this part of the test to test the ability of the interface to allow fine interaction with the data.

1


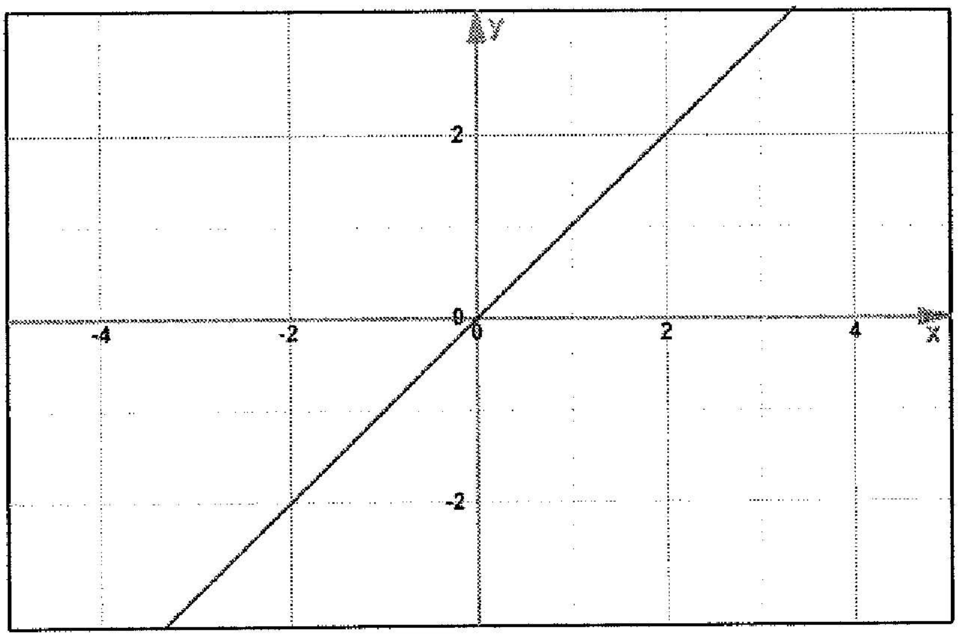
2


FIGURE 5.4: Graphs for section 2

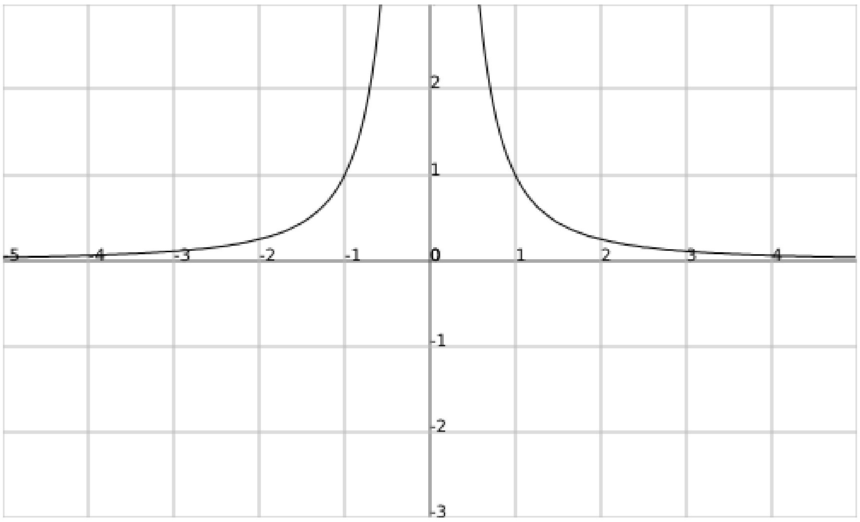


FIGURE 5.5: Graph 3.1

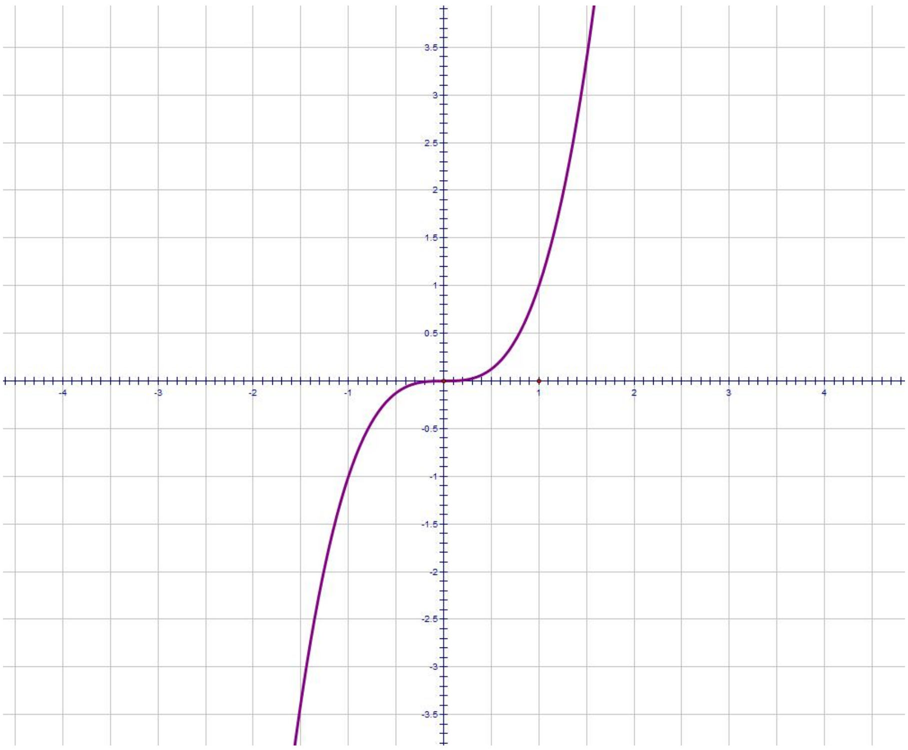


FIGURE 5.6: Graph 3.2

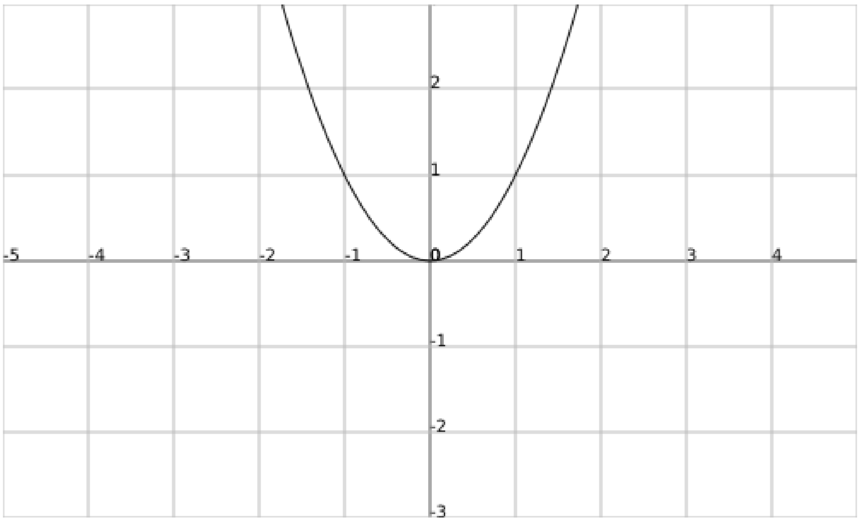
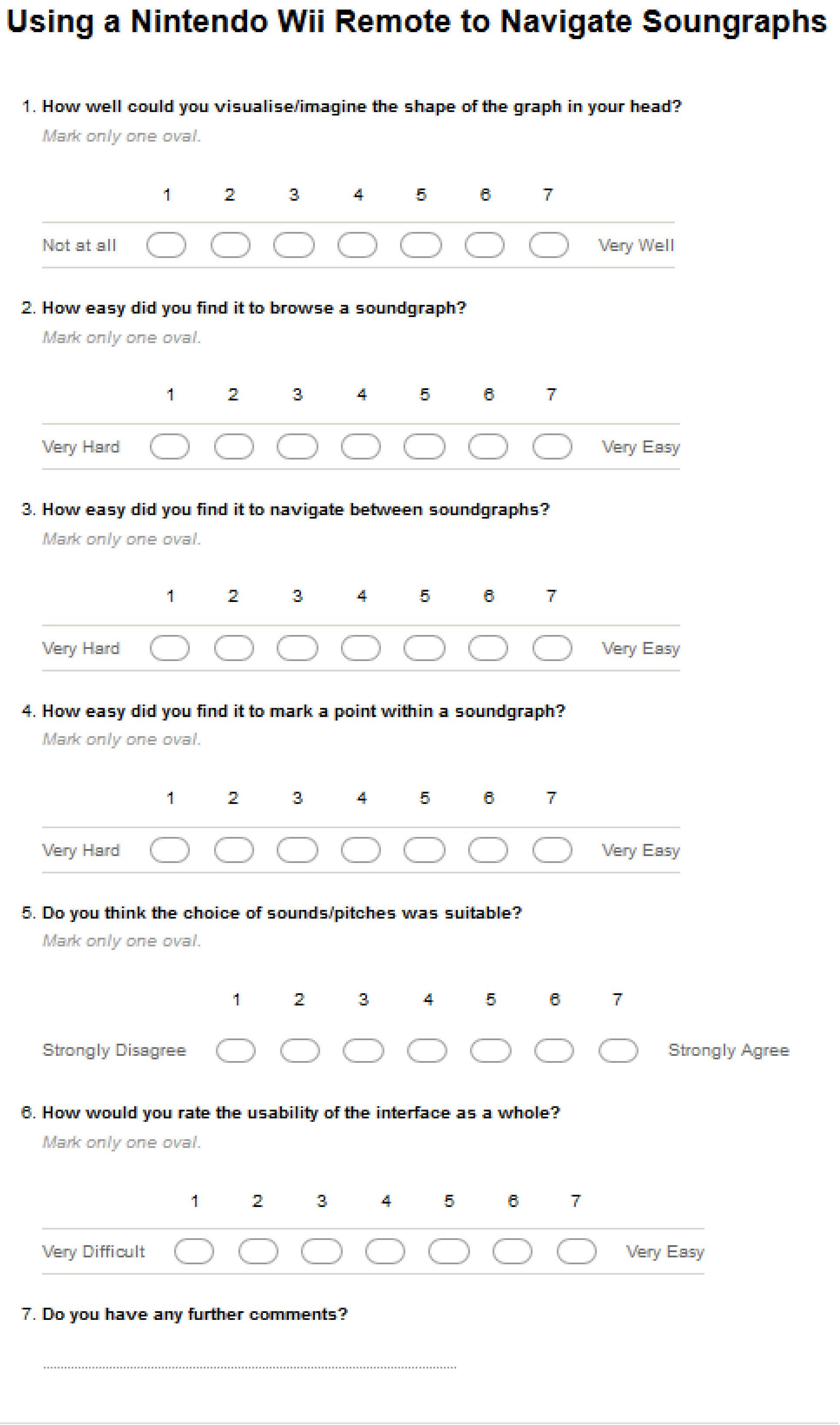


FIGURE 5.7: Graph 3.3

5.5 Section Four

The final section of the evaluation was a multiple choice questionnaire for the participants, marking number elements from 1 to 7. They were asked to evaluate the suitability of the audio display and the ease of use of different controls on the interface. A copy of the questionnaire can be found in figure 5.8.

The questionnaire continued the attempt to gather as much data as possible in a quantitative form. As such, almost all of the questions are on set on a scale from one to seven. The only exception was the final question asking for any additional comments that weren't covered by the existing questions.



Using a Nintendo Wii Remote to Navigate Soungraphs

1. How well could you visualise/imagine the shape of the graph in your head?
Mark only one oval.

	1	2	3	4	5	6	7	
Not at all	◯	◯	◯	◯	◯	◯	◯	Very Well

2. How easy did you find it to browse a soundgraph?
Mark only one oval.

	1	2	3	4	5	6	7	
Very Hard	◯	◯	◯	◯	◯	◯	◯	Very Easy

3. How easy did you find it to navigate between soundgraphs?
Mark only one oval.

	1	2	3	4	5	6	7	
Very Hard	◯	◯	◯	◯	◯	◯	◯	Very Easy

4. How easy did you find it to mark a point within a soundgraph?
Mark only one oval.

	1	2	3	4	5	6	7	
Very Hard	◯	◯	◯	◯	◯	◯	◯	Very Easy

5. Do you think the choice of sounds/pitches was suitable?
Mark only one oval.

	1	2	3	4	5	6	7	
Strongly Disagree	◯	◯	◯	◯	◯	◯	◯	Strongly Agree

6. How would you rate the usability of the interface as a whole?
Mark only one oval.

	1	2	3	4	5	6	7	
Very Difficult	◯	◯	◯	◯	◯	◯	◯	Very Easy

7. Do you have any further comments?

....

FIGURE 5.8: Questionnaire

Chapter 6

Results

In all, eleven people took part in the study. Due to the small number of people taking part in the study it not possible to make generalised statement about the general population or present any statistically significant results overall. It can however suggest something about the basic usability of the interface and how quickly users with no previous experience of the interface can interact with it to a satisfactory degree.

6.1 Participant information

The mean age of the subjects was 37, 5 men and 6 women took part and none reported any hearing difficulties. Other demographic information is stated in the tables below.

Highest Maths Qualification?	none	GCSE	A level	Degree	Masters	PhD
	0	9	2	0	0	0

Table 6.1: Highest Maths Qualification

Frequency of playing Wii?	Never	Infrequently	Monthly	Weekly	Daily
	5	5	0	0	1

Table 6.2: Frequency of playing on Wii console

Apart from greater age somewhat correlating with less frequent time spent using a Wiimote, there were no significant links between any other demographic information.

6.2 Evaluation section 1

The result of this section very much reflected the intention for each subsequent graph to be more difficult than the previous. The results are presented below in tables. To aid understanding and analysis, the graphs are arranged from left to right across the tables, starting from the correct graph and ending with the totally incorrect one. The two graphs which are somewhat similar to the correct one are placed in between.

Graph 1.1 Results	$y = \sqrt{x}$	$y = x$	$y = x^3$	$y = sin(x)$
	8	0	0	3

Table 6.3: Graph 1.1 selection frequency

Graph 1.2 Results	$y = x^2$	$y = 2x^2$	$y = cos(x)$	$y = x^3$
	5	4	2	0

Table 6.4: Graph 1.2 selection frequency

Graph 1.3 Results	$y = x$	$y = log(x)$	$y = \sqrt{x}$	$y = exp(x)$
	1	4	2	5

Table 6.5: Graph 1.3 selection frequency

6.3 Evaluation Section 2

Since the aim of this section was more to test the usability of the nunchuk interface, the graphs are in general less similar than in the first section. As such, the correct graphs' functions are displayed at the left of the following tables while the others are in no particular order to the right.

Graph 2.1 Results	$y = sin(x)$	$y = x$	$y = x^2$	$y = \frac{1}{x^2}$
	8	0	3	0

Table 6.6: Graph 2.1 selection frequency

Graph 2.2 Results	$y = x$	$y = x^3$	$y = exp(x)$	$y = x^3 - 20x$
	7	2	1	1

Table 6.7: Graph 2.2 selection frequency

6.4 Evaluation Section 3

Since this section required the selection of a datapoint rather than an entire graph the data is presented differently. The bar charts below represent the frequency of selection for all values of x between -5 and 5. The graphs did in fact extend beyond this point in both directions but since no one selected any values in that range they have been omitted for clarity. It is worth noting that there is one point in graph two which is several points above the central point. The participant said that they had mistaken the instruction and were trying to select the highest point again instead.

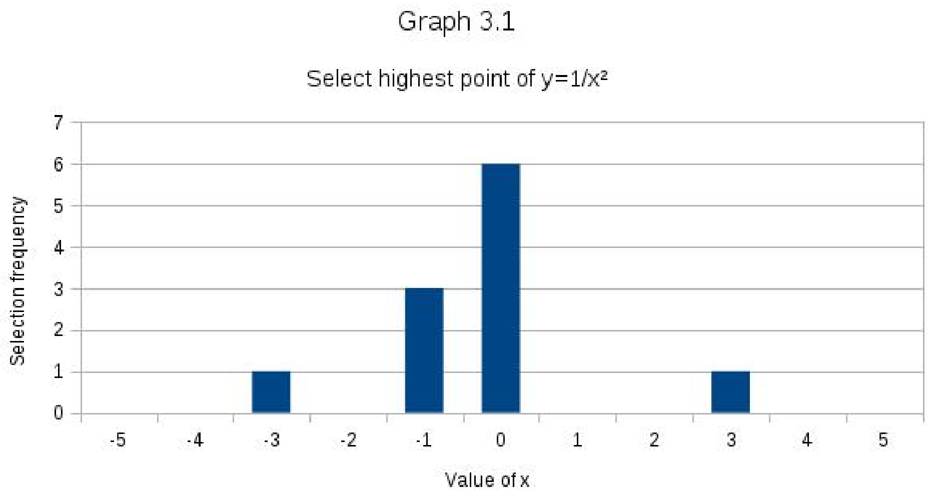


FIGURE 6.1: Results from Graph 3.1

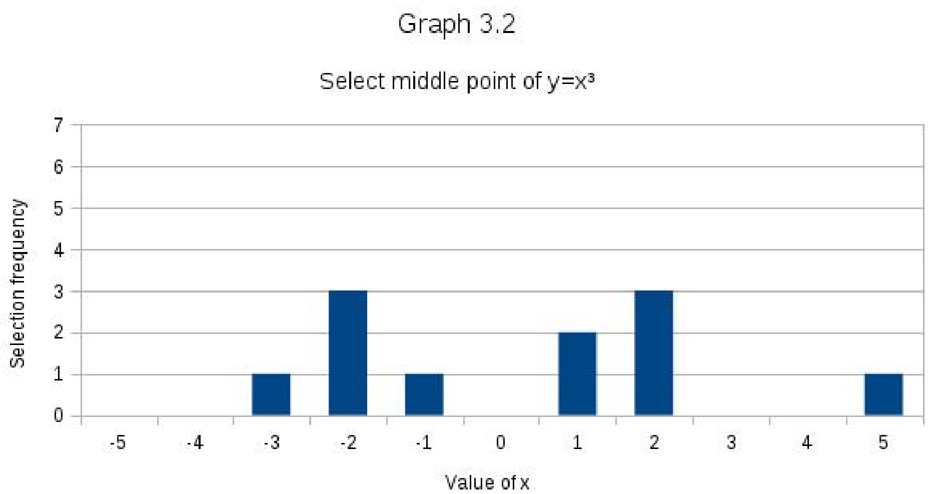


FIGURE 6.2: Results from Graph 3.2

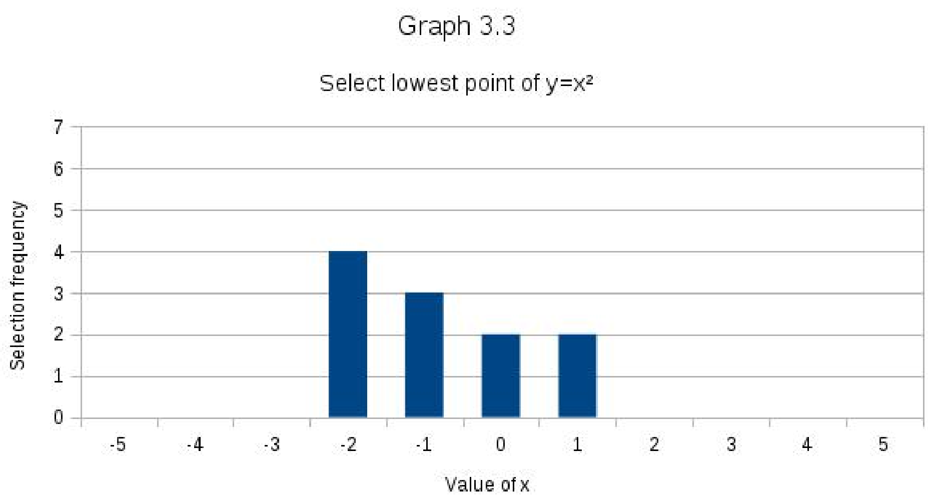


FIGURE 6.3: Results from Graph 3.3

6.5 Evaluation section 4

Due to the nature of the data from the questionnaire it has been presented as a graph of mean and range below. The questions from 1 to 6 are:

- How well could you visualise/imagine the shape of the graph in your head?
- How easy did you find it to browse a soundgraph?
- How easy did you find it to navigate between soundgraphs?
- How easy did you find it to mark a point?
- How suitable do you think the sounds/pitches used are?
- How would you rate the usability of the interface as a whole?

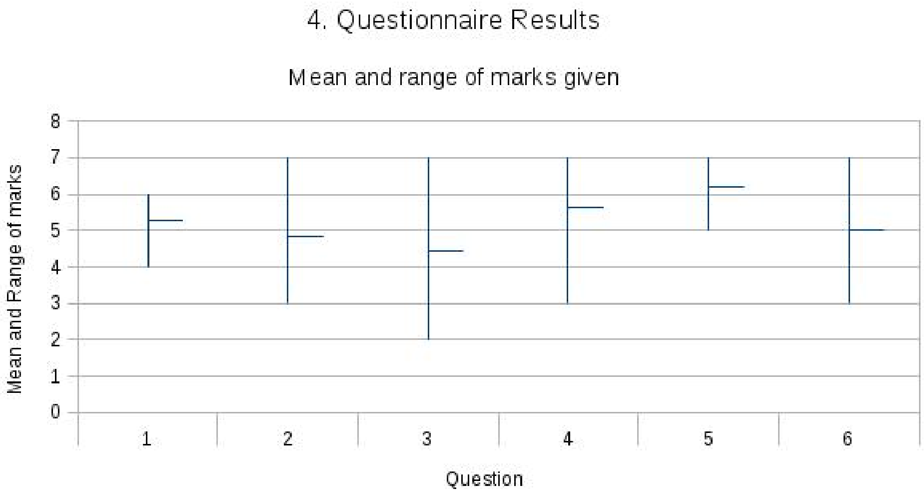


FIGURE 6.4: Results from Questionnaire

Chapter 7

Discussion

The results from the evaluation are generally positive while also highlighting a number of areas for improvement. The subjects performed better than chance: correct answers for sections 1-2 would be 1.25 by chance but the mean of the correct answers was in fact 2.54.

Correct answers by chance would be 0.3 for question 3 while the actual figure was 2.1. However the nature of question three, selecting of a value from a continuum, makes calculating the number of correct answers by chance more complex. This comparison to chance suggests that the program is definitely helping people understand these graphs to a certain degree. There is potentially an element of pre-existing ability which allowed some people to perform better than others overall. A high score in one test was a good predictor of high results in the others. You could however reasonably expect this to apply over the general population and so this causes no issues to the results.

7.1 Evaluation section one

The results for section one correspond generally with what was expected. As mentioned, the questions became harder as they moved along in order to test the limits of the program and this is borne out in the data. See fig 6.1 for information regarding correct answers by question.

7.1.1 Graph 1.1

The results for this section were somewhat surprising. It was expected that most people would be able to identify a linear progression graph out of the available options. What

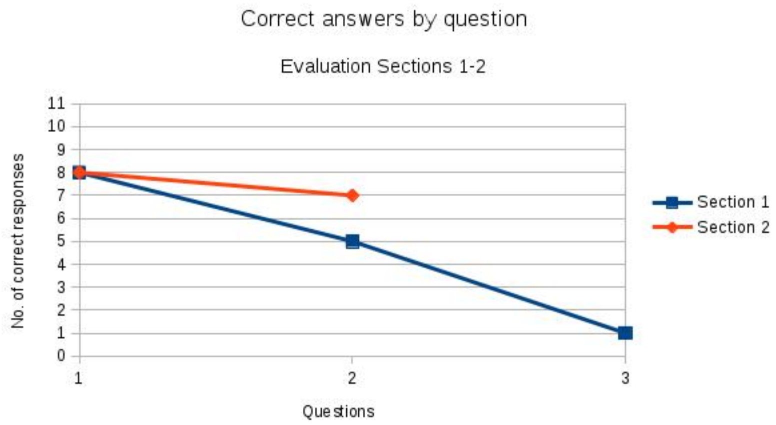


FIGURE 7.1: Correct answers by question

was unexpected was that the few people who got it wrong chose the graph that had no relation with the correct one, $y = sin(x)$. As a periodic waveform this shared no common ground with $y = x$. The best explanation that I can find is that, this being the first question, these subjects were too busy with the interface to correctly identify the graph. While it is true that the three people who got this incorrect did go on to be among the lower scorers overall, they did all correctly identify $y = x$ in from the audio in section 2.

7.1.2 Graph 1.2

While less than half or participants selected the "correct" graph, the difference between the correct answer and the most similar incorrect is very close: only the rate of change of the values of x differs between the two. This highlights one of the possible weaknesses of the interface: since the user is in control of the rate of change between the values of x, and since the highest and lowest pitch are constant for all graphs, the angle of the curve of a graph is basically impossible to determine. The audio just communicates that the graph is a curve. The choices regarding use of pitches and number of points to calculate were made with the scaling of data in mind but either some additional feedback. If the use of the program were to be purely for representing basic function shape (the second function being $20x^2$ rather than just x^2 then this may not be a problem. If more complex differentiation between data was required the more development would be required.

The third most common answer was a single period of $1 = cos(x)$ which shares a similar overall shape to $y = x^2$ but is inverted. The reasons for this are unclear and it doesn't seem wise to make up theories with little foundation at this point. As noted, this graph is a mirror image of the other and a mistake due to concentrating on the interface seems likely.

7.1.3 Graph 1.3

The outcome of this section is that the most incorrect graph was the most frequently chosen. Of course, this section was intended to be the most difficult one to judge. It's interesting to note that the most popular graph had an overall shape approximately opposite to that of the correct one. That is a small increase quickly rising opposed to a quicker change smoothing out. The second most popular choice had a similar rate of data change to the correct graph, but a more steep initial incline. In general it seems that participants could detect that the data progression wasn't linear but the direction of the curve wasn't clear. Although if that is the case, it's unusual that the completely linear graph y=x was chosen above the correct one which had a rate of change somewhere between the two extremes.

Some of the limitations of the interface come into play regarding these results. Two of the graphs from this section only begin at the vertical axis, yet the pitch of their initial data point is the same as that of the other two, which begin at the leftmost boundary of the graph. Once again this was implemented to allow a wider range of data to be displayed overall but has left some important elements of graphs impossible to represent. It also seems that while it is possible to detect that the distance between consecutive data points is not constant, if the changes are small enough it is hard to grasp other information regarding the shape of the graph.

7.2 Evaluation Section 2

The results from section 2 were a little surprising. It was unexpected that there should such a difference in performance between this test and the previous one which is demonstrated by figure 6.1. It is particularly interesting since the ease of navigating between graphs was rated lowest on the questionnaire at the end. There are a number of reasons why participants may performed better on this section. It's possible that the difficulty that people were having with the controls made them concentrate more closely at the interaction in general. However, the range of marks for that question shows that not all people were finding using the nunchuk that hard. Additionally, a person's rating of the nunchuk interface had no significant relation to how well they actually did in this part of the test. Since the majority of the information on this section was communicated through sound subjects had to concentrate more on their sense of hearing. That may have had an impact since the sight is the primary in sense in sighted people and does tend to overrule the others in most cases. The fact that all the participants were gaining more experience with the interface in general could also have had some impact on this result.

7.2.1 Graph 2.1

The three participants who chose incorrectly on this graph did chose another with a similar shape. It actually resembled one period of the waveform that was the actual answer. It may have been unclear that as they moved back and forth through the graph, that each end of the graph was actually a stopping point. Many people seemed to "get lost" in the array of graphs and may have chosen that because it sounded close enough. One participant did note in the comments that some indication of which graph they were listening to may have been of some assistance. Some additional feedback regarding position relative to the axes may also have been of some assistance.

7.2.2 Graph 2.2

The problems that hindered people with graph 1.3 seem to have been duplicated a little in this case. Most participants were able to identify that it was a totally linear graph, $y = x$. However, the other graphs in the array did share the overall direction and trend of data increase which may have confused some people. As suggested for graph 2.1, it may again be that some of the participants lost their way in the graphs and chose the one that seemed most likely at that point.

7.2.3 Evaluation Section 3

The result from this section seem quite negative upon first glance. It's certainly the case that only a few people selected the central point from any of the graphs correctly. There's another element to take into account however. Because this exercise was to select a value out of an array, then the differences between all of the values are not equal. You can determine the position of the "central" value by pitch or position in the array and either can be confused. In this case, the size of pitch change between certain values was a confounding factor.

The most extreme case in is graph 3.2 ($y = x^3$) where the values of $x = -1$ and $x = 1$ have pitches that are one tenth of one percent different to the pitch for $x = 0$. For the range of pitches implemented in this project, that equals to 0.75Hz. That is less than a quarter of the minimum audible frequency interval under laboratory conditions[34]. The next two values of $x = -2$ and $x = 2$ are 0.8 percent different which is 6Hz and is only barely detectable. Taking that into account means that 3 people picked what was audibly the same pitch, while six other selected one that was barely perceptible as being different. Graph 3.1,$y = \frac{1}{x^2}$, was easier with 10% difference on either side of the axis crossing. Graph 3.3 however had a similar issue to 3.2 in that values for $x = -1$ and $x = 1$ were only 1%, or 7.5 Hz, different in pitch to $x = 0$.

This factor has two important impacts on the project. Firstly, it does mean that the results for section three of the evaluation may not be as poor as they initially seem which is hopeful for the usefulness of the interface. This is also supported by the fact that the ease of marking points was rated third in the questionnaire. The other matter is the range of pitches implemented and how they might be altered in future iterations of development. There have been several issues that need to be examined with regards to the sound output of the interface.

7.3 Evaluation Section 4

The highest scored parts of the questionnaire with the smallest range of values are how easy it is to visualise the graph in your head, and how suitable the sounds/pitches are. This was rated highly across all participants. This supports previous work in soundgraphs which show that this is a useful and accessible way of presenting mathematical data. The lowest marked elements was navigating between soundgraphs from evaluation section 2. This is somewhat strange since this was also the area where participants performed better on the whole. It is possible that this is a very effective way of presenting the information whether or not people find the interface easy. However, this question is also the one with the largest range of marks given so extrapolations taken from the mean rating may not be reliable.

Other elements of the questionnaire had such wide ranges of values that the mean values are difficult to apply to general conclusions. This is hindered more by the limited sample size. The result with the smallest gap between the mean and maximum values is how easy did you find it to mark a point. disregarding the issues regarding the pitches as detailed above, this seems to have been an area that people found more easy than others.

Some values given in the questionnaire correlated with participant's demographic information or test results. On average, people who had never used a Wii remote marked the usability of the interface 1 point lower than those who had. Many comments also remarked that they thought the interface would become easier to use with practice. This does suggest that experience with the interface in general could have a significant effect on performance in the tests. Any novel interface takes time to learn and if it were possible, a follow up test with the same participants may show an overall improvement in performance.

Interestingly, the five top scoring participants gave the overall usability of the interface an average mark of 4.6 while the lower six gave it a mean mark of 5.3. This could suggest that people with a better ability to imagine sounds in a concrete way are able to do so regardless of their experience of the interface. Once again, however, it is difficult to be certain of such generalisations with the limited sample size of this study.

Some of the questions in the questionnaire could have benefited from more specific targeting. For example, the question "How easy did you find it to navigate between soundgraphs?" which was the lowest marked in the questionnaire. This confuses the physical interface of the nunchuk with the subjects ability to visualise the array of four graphs and their virtual position within it. Subjects definitely found both elements difficult in their own right. Additionally, when put together each element interfered with the subject's ability to successfully complete the other. It would be useful to identify which of the two parts of the problem was causing the most issues as each require different changes to rectify or improve problems with it. Other questions such as the ease of browsing a soundgraph and the ease of marking a point combine/confuse information about the subject's understanding of the audio and their use of the interface. Although some of these doubts can be eased by the relatively high ratings for visualising the shape of a graph and the choice of sounds and pitches, that is enough to remove the need for further investigation.

7.4 General points

There are a lot of confounding variables in the study. For example, as a person goes through the test they gain more experience with the controls which will influence their performance on the later parts. As mentioned before, the study combined questions regarding subject's audio understanding of the graphs, and their use of the interface. This does provide useful information regarding this particular execution of the interface but both would benefit from extra, individual testing.

Although it was highly marked by participants, the choice of sound and pitches is actually under some question. Using the maximum range of of available pitches for each graph provided the ability for maximum scaling of the data represented but causes a number of problems in the perception of certain elements of the graphs. For example, the rate of change in values between x^2 and $20x^2$ which is quite large represented as numbers, was not perceivable from the output of this program. Also details like $\sqrt{x}$ and $log(x)$ not starting until after the crossing of $x = 0$ were lost. However, for the communication of the basic shapes of functions, the interface performed very well. Participants selected graphs that were either correct or very close in most cases.

Some of the problems regarding the audio come from the discrete representation of the data. The requirement to scale the number of data points represented in order to include a wider range of data introduces a number of problems regarding minimum audible pitch intervals as seen above. The discrete representation was implemented originally to aid in giving an intuitive sense of function shape. Whereas in this execution some of the design choices made in order to implement additional interface functions have left some details of the graphs unclear. The central values and position relative to the axes, of particular graphs are not made explicit.

Perhaps one the question is, what is the main element of graphs that the interface needs to communicate, the shape or the data? This is not entirely clear from the initial specifications. This interface was mainly effective at representing information such as general data trends, and overall shape of curves in graphs. However, finer details such as rate of change, and difference between closely related data points could be lost. Was there any value in separately representing the 5 central values of $y = x^3$ when they differed by only 0.8 percent of the overall range of values? Once again that depends on the aim of the program. If you are trying to represent the data then the ability to zoom in on those values would be useful but doing so could make the overall shape of the graph ultimately unclear. While, as seen in this project, giving a wide enough range of data to fully communicate the shape of the graph can make the perception of the specific data more difficult. If the general shape of the graph is the most important part of the experience then perhaps returning to a continuous sonic representation will actually help people more than the discrete.

In sections one and two of the evaluation section of the project, subjects selected the graph they thought was being represented from a finite list. As noted in the design section, this choice was made to provide mainly quantitative data for analysis. This was achieved but still had some ambiguities due to the subject's different understanding of the shape of a graph. As mentioned, many previous studies have asked that subjects draw the shape of the graph they hear. Following this study, I can see that having a person concentrate enough on the shape of the graph to draw it can make them truly use the full potential of the soundgraph. By providing multiple choice answers, I had done some of the imaginary work for them. The problem regarding accurate representation of the data remains the same either way though. the difference between a graph starting at zero or continuing throughout the whole frame remains unclear from a purely audio representation.

Another problem with the audio representation of the data is the logarithmic nature of audio pitch. Each octave change in pitch is in fact a doubling or halving of the frequency of the wave. Therefore as the pitch increases, each step in the data needs to be scaled up slightly from the previous one in order for the distance between each to be equal in our perception. The reverse is also true for decreasing pitch, the actual size of the gaps between data points will appear to increase as the pitch lowers. If you were to take a totally linear data set and play a pitch at equal frequency differences for each data point, the output graph would actually look like a log graph, i.e. a sharp rise which quickly gets shallower then slowly gets closer and closer to horizontal. While this is a problem to take into account for a discrete representation, it would be an even more complex issue to solve in a continuous soundgraph.

Chapter 8

Conclusion

This project was to create an interface for people to browse audio representations of mathematical graphs. Doing so allows the blind and partially sighted greater access to mathematics which is not normally within their reach. It was to be implemented using a game controller as the primary input for this purpose. The intention was to provide a proof-of-concept for the usability of this kind of interface for the successful identification of soundgraphs and their elements. This development was a continuation of one already undertaken to prove the effectiveness of a discrete sound representation of audio graphs. In this way each data point of a graph was represented as a distinct sound with a pitch relative to the value and position it represents within a data series. This was in opposition to the more common way which utilises a continuous pitch to communicate the general shape and trend of a graph, generally ignoring the specific data involved. Following some initial changes to the design brief due to time and controller concerns the interface was successfully created and a small evaluation study run with eleven people. The outcome of the evaluation was broadly positive and highlighted a number of additional points that were not initially considered such as the minimum perceptible change in audio frequency and how that affected some of the soundgraphs.

On the whole I believe that this conceptual interface has promise and could be the basis of a successful full implementation of Soundgraph software. Users have found it reasonably simple to uses even when having no previous experience with the interface device. It also seems that with use will become even simpler with relatively small amounts of practice. Whether a discrete or continuous audio representation is better for this application remains to be seen. Future developments will need to include additional feedback cues. Some of those identified in this project are changeable timbre, spatialisation, and haptic feedback.

Chapter 9

Future Work

The outcome of this project has been a moderate success in meeting the initial brief, to develop and game controller interface for soundgraphs. I think that one of the most valuable things about the outcome has been to identify a number of new areas for exploration.

9.1 Further Wiimote study

The interface with the Wiimote has a great deal of potential but would need further and more targeted study. As mentioned in the discussion section, many of the points in the questionnaire evaluate both the audio and physical interfaces together. Therefore it's uncertain which part of the interface is having the most influence on the outcome. Most of the exercises in the evaluation also mix the outcomes of both the audio and physical interfaces. For instance the point navigating between soundgraphs that a participant was struggling to learn the new manual controls while also trying to conceptualise a virtual array of four soundgraphs, their own position within that array, and the shape of whichever graph they were interacting with at that point. While the study of using audio graphs to visualise mathematical functions has been quite well studied, further study into the other elements may be positive. Investigating whether people can successfully navigate a virtual array while being aware of their position within it would be of merit either with or without the Wiimote interface. Further work on navigating soundgraphs where the primary aim is to prove the usefulness of the interface, trying to eliminate the soundgraph as a variable as much as possible while still keeping it as part of the study.

9.1.1 I

nput C Due to the aims of this project all control inputs were taken from accelerometer data which is just a subset of all the inputs available. One of the study participants asked why the nunchuk thumbstick hadn't been used to shift between graphs rather than the wrist motion. Which is a fair question. Upon further development it may well be that some of the program functions would be would be better controlled by the more traditional buttons/joysticks. For example all of the functions regarding the navigating and marking of graphs could be controlled instead by the thumbstick and buttons on the nunchuk. That would remove the short term uncertainty felt by people as they start to use the program. These decisions would need to made as further operations were added to the program. There are only a finite number of control options in any one interface and it would be unfortunate to neglect some in order to avoid a small learning curve.

9.2 Representation of graph data

A large amount of further study should go into refining the ability of the interface to communicate more information regarding the rate of data change as well as the general trend. Looking further into making a user aware of the origin point of a graph and the values at which it crosses the axes would also be of benefit.

Regarding the audio output, it's possible that one specific pitch should represent the $x = 0$ point in the data. This does however run into problems of it's own. There's the problem of which pitch to chose for this. While the ideal human audio perception goes from 20Hz to 20kHz, most adults cannot hear that high and almost all will differ from each other in some way. Also, if a data set has an exponential increase then it will quickly run out of available pitches. The issue of perceptible differences in pitch will also be something to consider. The minimal value of 3.6 Hz was demonstrated between 1000Hz and 2000Hz[34] and will increase as the overall pitch does. Perhaps some other part of the audio could display these details. A greater or lesser level of a secondary waveform or modulation of the original wave included in the output could communicate this. Making the sonic output more texturally complex the data moves further from the zero point.

Spatialisation was initially only considered for use when concurrently browsing a number of different soundgraphs. For that reason it was not included in this implementation. However, many participants seemed to have a problem maintaining awareness of their position within a particular graph. If an element of spatialisation were added, with stereo positioning mapping to values of x, it could help users to maintain more of an awareness of their current point in the data. Spatialising the output in this way could also aid people in identifying the central point in the data

Adding haptic feedback into the interface could aid in this. Some feedback from the Wiimote as you cross one or other of the axes could be quite easily implemented. The Wiimote has the ability to provide haptic feedback in the form of vibration. A different duration of vibration could signify which axis you have crossed. One of the issues that this could solve was the mistake in telling the difference between a sine wave and a simple bell curve. The vibration would make it clear that the x axis had been traversed several times per pass, showing that the function was periodic.

One final option for dealing with identifying the zero point of the graph. The current implementation has the graph “fixed” in between two positions of the Wiimote and navigation begins at whichever point the Wiimote is at when initiated. IIt could be made so that whichever point the Wiimote is directed at when browsing is started is the zero point. A drawback to this however is the somewhat limited arc in which the Wiimote can detect the sensor bar.

It is likely that a combination of the above or other options will be included in any final implementation. Using several modes of communication, e.g. haptic and audio feedback, is a more effective way of communicating the information in the graph. As mentioned previously, for the purpose of this project only accelerometer input was used. Any future iterations should certainly use the other available controls in addition.

Appendix A

Consent Form

Using a Nintendo Wii remote to navigate soundgraphs.

Blind and partially sighted people have a hard time studying maths because so much information can be put into one small, visual formula or symbol that is difficult or lengthy to describe otherwise. A soundgraph is a way of representing the mathematical information that you would normally see in a line graph in an audio form. The aim of this experiment is to investigate the efficacy of using a Wii remote as an interface for this so we are gaining feedback from a number of people to help us do do. Following some demographic questions you will listen to some soundgraphs through headphones and perform a number of simple identification and interface tasks. I'll be running through this with you. Please remember that it's the interface being tested, not you. Following the test there is a short survey I will ask you to complete. You can ask any questions at any point and you are free to withdraw from the experiment at anytime. All data gather from this usability study will be treated in a confidential fashion and only used for the purposes of this evaluation. There are no known risks to participation in this experiment.

Do you agree to take part in this experiment?

`Date: _____________________ Signature: ____________________`

Bibliography

[1] Stevens R. D. Edwards, A. D. N. and I. J. Pitt. Représentation non visuelle des mathématiques. *Les D'eficit Visuel*, pages 169 – 178, 1995.

[2] Arthur I. Karshmer and Chris Bledsoe. Access to mathematics by blind students. In *Computers helping people with special needs.*, pages 471 – 476. Springer, 2002.

[3] Douglas L Mansur, Merra M Blattner, and Kenneth I Joy. Sound graphs: A numerical data analysis method for the blind. *Journal of Medical Systems*, 9(3): 163–174, 1985.

[4] Catherine M Wilson and Suresh K Lodha. Listen: A data sonification toolkit. Master's thesis, University of California, Santa Cruz, 1996.

[5] Suresh K Lodha, John Beahan, Travis Heppe, Abigail Joseph, and Brett Zane-Ulman. Muse: A musical data sonification toolkit. In *Proceedings of International Conference on Auditory Display (ICAD'97)*, 1997.

[6] Abigail J Joseph and Suresh K Lodha. Musart: Musical audio transfer function real-time toolkit. In *Proceedings of the International Conference on Auditory Display*, 2002.

[7] Florian Grond, Trixi Drossard, and Thomas Hermann. Sonicfunction: Experiments with a function browser for the visually impaired. In *Proceedings of the 16th international conference on auditory display*, 2010.

[8] Philip Harling, Robert Stevens, and Alistair Edwards. Mathgrasp: The design of an algebra manipulation tool for visually disabled mathematicians using spatial-sound and manual gestures, 1996.

[9] Robert Stevens and Alistair Edwards. Mathtalk: The design of an interface for reading algebra using speech. In *Computers for Handicapped Persons: Proceedings of ICCHP '94, Lecture Notes in Computer Science 860*, pages 313–320. Springer-Verlag, 1994.

[10] Robert D. Stevens, Alistair D. N. Edwards, and Philip A. Harling. Access to mathematics for visually disabled students through multimodal interaction. *Human-Computer Interaction*, 12(1):47–92, March 1997. ISSN 0737-0024.

[11] Philip Kortum. *HCI beyond the GUI: design for haptic, speech, olfactory, and other nontraditional interfaces*. Morgan Kaufmann, 2008.

[12] Hui Tang and D.J. Beebe. An oral tactile interface for blind navigation. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 14(1):116–123, March 2006.

[13] Wai Yu, Katri Kangas, and Stephen Brewster. Web-based haptic applications for blind people to create virtual graphs. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings. 11th Symposium on*, pages 318–325. IEEE, 2003.

[14] Wai Yu, Ramesh Ramloll, and Stephen Brewster. Haptic graphs for blind computer users. In *Haptic human-computer interaction*, pages 41–51. Springer, 2001.

[15] Jean-Luc Nespoulous, Paul Perron, and André Roch Lecours. *The biological foundations of gesture: Motor and semiotic aspects*. Psychology Press, 2014.

[16] Johnny Chung Lee. In search of a natural gesture. *ACM Crossroads*, 16(4):9–12, 2010.

[17] L. Greenemeier. Chipmaker races to save stephen hawking's speech as his condition deteriorates. *Scientific American*, 2013.

[18] Christine Youngblut, Rob E Johnston, Sarah H Nash, Ruth A Wienclaw, and Craig A Will. Review of virtual environment interface technology. Technical report, DTIC Document, 1996.

[19] Alistair DN Edwards, Heather McCartney, and Flavio Fogarolo. Lambda:: a multimodal approach to making mathematics accessible to blind students. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*, pages 48–54. ACM, 2006.

[20] Marion Hersh and Michael A Johnson. *Assistive technology for visually impaired and blind people*. Springer, 2010.

[21] P. Takacs. Resurrecting one of the world's 1st video games. scienceblogs.com/brookhaven/2010/12/14/resurrecting-one-of-the-worlds/, 2010. [Online; Accessed 20/09/2014].

[22] N. Nova and L. Bolli. *Joypads! The design of game controllers.* CreateSpace Independent Publishing Platform, 2014.

[23] C. Herold. The history of gesture gaming.

[24] Johnny Chung Lee. Hacking the nintendo wii remote. *Pervasive Computing, IEEE*, 7(3):39–45, 2008.

[25] Romulo Ochoa, Frank G Rooney, and William J Somers. Using the wiimote in introductory physics experiments. *The Physics Teacher*, 49(1):16–18, 2011.

[26] Andrej Gams and Pierre-André Mudry. Gaming controllers for research robots: controlling a humanoid robot using a wiimote. In *17th International Electrotechnical and Computer Science Conference (ERK08)*, 2008.

[27] Jilyan Decker, Harmony Li, Dan Losowyj, and Vivek Prakash. Wiihabilitation: rehabilitation of wrist flexion and extension using a wiimote-based game system. *Governor's School of Engineering and Technology Research Journal*, pages 92–98, 2009.

[28] Gustavo Saposnik, Robert Teasell, Muhammad Mamdani, Judith Hall, William McIlroy, Donna Cheung, Kevin E Thorpe, Leonardo G Cohen, Mark Bayley, et al. Effectiveness of virtual reality using wii gaming technology in stroke rehabilitation a pilot randomized clinical trial and proof of principle. *Stroke*, 41(7):1477–1484, 2010.

[29] Lindsay Evett, Allan Ridley, Liz Keating, Patrick Merritt, Nick Shopland, and David Brown. Designing serious games for people with disabilities: Game, set and match to the wii^{TM}. *International Journal of Game-Based Learning (IJGBL)*, 1 (4):11–19, 2011.

[30] Yang-Wai Chow. The wii remote as an input device for 3d interaction in immersive head-mounted display virtual reality. In *Proceedings of IADIS International Conference Gaming*, pages 85–92, 2008.

[31] Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, et al. Manifesto for agile software development. 2001.

[32]

[33]

[34] H.F. Olson. *Music, Physics and Engineering.* Dover Books. Dover Publications, 1967. ISBN 9780486217697.