Iterative Design for Approachable, Mobile-Based Collaborative Electronic Music Instruments

by

Timothy J. Barraclough

A thesis
submitted to the Victoria University of Wellington
in fulfilment of the
requirements for the degree of
Masters of Engineering
in Software Engineering.

Victoria University of Wellington
2015

Abstract

The steep learning curve for technology for electronic music creation enforces a barrier to entry for new electronic musicians that prevents them from enjoying the playful process of creating music, without significant investment of time, energy, or money. This thesis argues that this barrier to entry should be abolished by using creative technology as a platform in order to begin the democratisation of electronic music creation processes. By drawing upon specific values inherent to the playful process of creating music, this thesis suggests a framework for iterative design of social, electronic musical instruments. The combination of implementing this framework and informing design directions upon user feedback has resulted in the creation of Pyxis Minor, a new, smartphone and tablet based, electronic musical interface/instrument that allows users of any prior musical background to have a playful creative and social experience of making music. This is important for distinguishing that the apparent barrier to entry for electronic music creation may be unnecessary, and it is possible to develop instruments and interfaces for the playful creation of electronic music that fulfil the needs and requirements of a diverse range of users.

Acknowledgments

First and foremost, I would like to thank my advisors, Ajay Kapur and Dale Carnegie, without your guidance my work would not be what it is now, and more importantly, I would not be the artist and engineer I am today.

Thanks to Ted Apel, for giving me the critical skills and confidence to really consider myself an artist. By encouraging me to think more critically about my work, I feel that, thanks to you, I have found the meaning in my art.

Thanks to the whole of SELCT for helping with user studies, offering support and overall, creating some of my favourite works. It has been my privilege to be a part of the team. Special thanks Bridget and Diana for taking the time to read my work and critique my grammar and formatting.

Thanks to Paul for listening to and working through harebrained ideas, providing feedback and critiquing my general existence. I look forward to collaborating with you in the future.

Thanks to the Victoria University Scholarships Office for providing me the financial support to continue my studies. I am truly appreciative to have had this opportunity to further develop my academic skills by working on a project that is important to me.

Finally, thanks to my parents, Maureen and John, for giving me support and advice when I needed it most.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

> *Life means movement and action, and if we wish to find a meaning in it we must seek for activities which carry their own purpose and value within them, independently of any extraneous goals....There really are such activities...we must call them play.*
>
> Moritz Schlick [1]

It is no coincidence that we often use the word 'playing' synonymously with 'performing' when discussing a musician using a musical instrument. Although the level of skill required to truly 'play' an instrument is subjective, it can be argued that at least some familiarity with the instrument is required. In the realm of digital electronic music creation and performance the barrier to playfulness is built on a combination of computer literacy, digital electronic music theory and a veil of jargon. As such, finding a suitable entry point for a beginner can be a difficult task.

Inherent in its nature, electronic music creation requires a technological element; to become a proficient electronic musician requires a technical understanding of at least one chosen tool or instrument, similarly to how an acoustic music performer requires a technical understanding of their chosen instrument. One of the key differences that separates the two is that, unlike traditional acoustic instruments, electronic musical instruments require a loudspeaker for producing sound. In the first four of the categories of the Hornbostel-Sachs classification of instruments (idiophones, chordophones, membranophones and aerophones), the sound making mechanism is a physical object, able to be heard without electrical amplification [2]. A later revision supposed a fifth category of instrument, electrophones, that require a loudspeaker to present the sound [3]. The direct causal interaction with the first four categories means that we have a rich tradition of understanding how a performer's gestural interaction translates to sound. This is not necessarily the case with electronic music instruments; they are often 'black-box' technology, hiding the process required to transform the musician's physical interaction into sound or music. This can create a barrier to entry for creating electronic music as beginners are not provided with the transparent sound making mechanism inherent to acoustic instruments.

Naturally, the people that are more likely to spend money on purchasing new music technology are more likely to have prior musicianship training and, subsequently, have different requirements of a musical instrument/application than someone without such training. As such, the developers of the technology, due to market imperative, aim to implement features targeted to this demographic which naturally leads to more complex interfaces. Evidence of this can be seen in the jargon heavy marketing and press information released by large music technology companies, for example the website advertising the Korg MS-20 synthesiser [4]. This not only makes the interfaces more daunting to beginner musicians, but subsequently furthers the conceptual understanding schism between beginner electronic musicians and those with prior training.

The combination of these factors creates a barrier to entry for electronic music creation and performance, which can prevent novice musicians from experiencing the joy of 'playing' music, as a creative, social experience. By utilising iterative design methodology, user interface and interaction design and modern smart-phone technology, we can achieve the goal of this research; to create an electronic musical instrument that lowers the barrier to entry for electronic music and inspires musical creativity in potential and existing electronic musicians. This will simultaneously allow people with varying prior musical training, the social and creative experience of playing electronic music, but importantly, will not sacrifice the expressivity and technical features required and requested by more experienced musicians. By implementing sensible and accessible electronic musical instruments we can begin a discussion about the democratisation of electronic music processes.

As an artist who has designed and interacted with many electronic musical instruments and interfaces, based upon experience and prior knowledge, a research area was identified. This research aims to develop a framework that will be useful in designing an interaction method/interface for people of varying prior musical training in order to 'play' music and collaborate on music making together. This framework is based upon a combination of prior experience, artistic intention and analysis of the related work surrounding the research focus. This framework will guide the construction of an instrument/interface that achieves the research goals. The framework is comprised of a number of design values and constraints that will help to maintain both engineering and artistic integrity of the resulting instrument. These key values are:

- **User Informed Design:** The development of the instrument will be implemented and improved based on feedback from, and consultation with, users from a range of prior musicianship training. This will help to balance the requirements of a variety of musicians' needs and requirements.

- **Active Interaction:** The instrument will centre around the *active* process of creating electronic music. Furthermore, the existence of the instrument will favour the process of creating music rather than the resulting music itself. The instrument will encourage users to 'play', without imposing predetermined goals or musical outputs upon the user, allowing them to define their own intention and musical goals.

- **Easily Accessible:** The instrument will be easily accessible to potential users. This means it will need to be both inexpensive and widely available. A large amount of technology developed for the creation of electronic music is expensive or difficult to acquire, requiring a significant investment from the user (either in the form of learning curve or monetary investment).

- **Social Collaboration:** The instrument will allow for an element of collaboration between users to allow them to play music *together*.

This is based on the idea that musical complexity can naturally result from the interplay of two or more musicians and, perhaps more importantly, that this kind of social interaction is one of the most exciting and rewarding elements of 'playing' music.

- **Technical Transparency:** Where possible, the interface will be indicative of the underlying audio process. Although intended to allow novices to play music, the interface will have a potential pedagogical function to educate the user about the specific electronic music processes and jargon without requiring prior knowledge to play.
- **Artistically & Aesthetically Valid:** A focus on creating a coherent visual and audio aesthetic - the resulting instrument will look, feel and sound like a work of art in itself. The intended outcome is that users will be more inclined to feel as though their experience with the application is itself creative if the resulting instrument can be considered as a work of art.

iOS and Mac OS X have been chosen as the platforms for the development of a new instrument / interface for the creation of electronic music. This is due primarily to two main reasons. Firstly, the ease of distribution of the resulting instrument will be supported by the existing Apple application stores and the user base thereof. Although Android has the larger market share for smartphone devices [5], the capabilities for real-time audio on Android devices are not as developed as with Apple devices. This is predominantly due to the Apple Core Audio Framework, which is integrated into iOS and Mac OS X. Core Audio has a rich history of implementation for software instrument design, and subsequently, is well documented for the purpose of creating real-time audio instruments and devices. Building upon this, the resulting instrument will be limited to utilising the hardware inputs and outputs available on the devices themselves as to remain accessible to owners/users of iOS and Mac OS X devices without the need for any specialist equipment, in order to remain *easily accessible*.

1.2 Research Goal and Evaluative Criteria

The primary goal of this research is to create an electronic musical instrument that lowers the barrier to entry for electronic music, inspires musical creativity in new and experienced electronic musicians and incorporates a social element that will allow users to collaborate together. Due to the subjective nature of this goal, the success of this research in achieving the goal will be determined by user studies conducted over the course of the research. Confirmation of ethics approval and the user study documentation can be found in Appendix A.

The first criterion to be used will evaluate the *creative* aspect of the resulting application. A 5 point Likert scale will be constructed and used to question users as to whether they feel sufficiently creative using the application. This scale was chosen because it will limit user responses to predefined categories, that then give results for benchmarking user attitudes towards the application, as demonstrated by Poepel in 'On Interface Expressivity: a Player Based Study' [6]. Tabulating these results will give numerical data to determine the success of the application in regards to affording musical creativity. If the mean of these results is above 3.0 (indicating favourable creativity), we will determine the application as allowing users to experience creativity. Additionally, by testing this mean in relation to multiple stages of the iterative design process, we can determine if each design iteration has improves the system.

Similarly, the second criterion, user opinion indicating the *usability* of the resulting application, will also be determined by the use of a 5 point Likert scale. If the tabulation over the sampled population results in a mean above 3.0 (indicating generally favourable usability) with a small standard deviation the application will be considered successfully usable. s The third criterion is the *socially collaborative* aspect of the application. This is based upon the idea that the application will provide a more enriching experience with more than one user interacting simultaneously, socially. The first part of this criteria requires that users agree that the collaborative aspect of the application is, in fact, social. Again, a 5 point Likert scale will be used. A mean above 3.0, with a small standard deviation, will indicate that the collaborative aspect of the resulting application is *social*. An additional question will be asked for user preference between solo and collaborative interaction. If only a small number of users select their preference as solo interaction, reflecting a definite minority, we will determine that the social aspect of the application (if the application is deemed social) improves the overall user experience for the majority of users.

These three criteria together will determine whether the resulting application is successful at lowering the barrier to entry for electronic music, inspiring creativity in potential and existing electronic musicians and allows for a social collaboration between multiple users. For ease of reference, this evaluative criteria will be considered as a list, whereby the resulting application:

1. Allows and encourages creativity.

2. Is usable.

3. Allows users a degree of social collaboration.

1.3 Thesis Structure

The first section has provided the motivation for this research by situating the position taken in this research and suggesting a framework for the construction of a new, socially-collaborative electronic musical instrument. Furthermore, the specific goal of the research and the evaluative criteria that will be used to determine the success of this research at achieving the stated research goal will be discussed. An overview of the structure of this research document will follow.

Chapter two examines the related works in related fields. Firstly, this will focus upon the existing literature in the field of *collaborative musical instrument and interface design* and the technology afforded group-based performance paradigms involved (section 2.1). Secondly,, a historical examination of *interconnected musical performance* takes place in section 2.2. Thirdly, the contributing related work in the field of *musical instrument and interface design for smartphone applications* is explored in section 2.3. The inspection and analysis of these three related fields will validate the specifications of the framework established in section 1.1 whilst providing an academic context for this research.

Chapter three will discusses the iterative design based methodology of this research. After establishing and justifying the specific values that comprise the framework, chapter three establishes and documents the early stages of development of an application using the framework discussed in section 1.1 and justified in chapter two. This presents the first three iterations of the development process and the corresponding user studies that informed and directed the application construction and improvement. This development process focuses predominantly on establishing the concept for the interface/instrument and transforming the concept into a usable prototype whilst investigating specific usability issues that arise.

The later iterations of development that resulted in the iOS application, Pyxis Minor, are explored in chapter four. This covers the refinements and improvements necessary to release the resulting interface/instrument to the public. Furthermore, this chapter discusses the necessary requirements to add functionality to the interface/instrument, as informed by the user studies in chapter three and the aesthetic decisions that helped to shape the resulting application. This chapter also explores the addition of a collaborative element to the interface/instrument to make it sufficiently social to achieve the research criteria established in section 1.2.

Chapter five evaluates the success of Pyxis Minor at achieving the goal outlined in chapter 1. In order to determine this success, a final user study is conducted. The user study tests the various evaluative criteria described in section 1.2. If the data gathered from the user study indicates that all the criteria has been sufficiently fulfilled, then Pyxis Minor,will be deemed successful in achieving the goal of this research.

The final chapter concludes the research by commenting on the overall success of the research and the implications for the fields of study explored in chapter 2. Avenues for future work based upon, and related to, this research will be discussed. An additional discussion will include work that remains to be done to Pyxis Minor, that was avoided during this research due to the lack of relevance to the goal of the research, and the future of Pyxis Minor. A visual overview of this structure can be seen in figure 1.1.

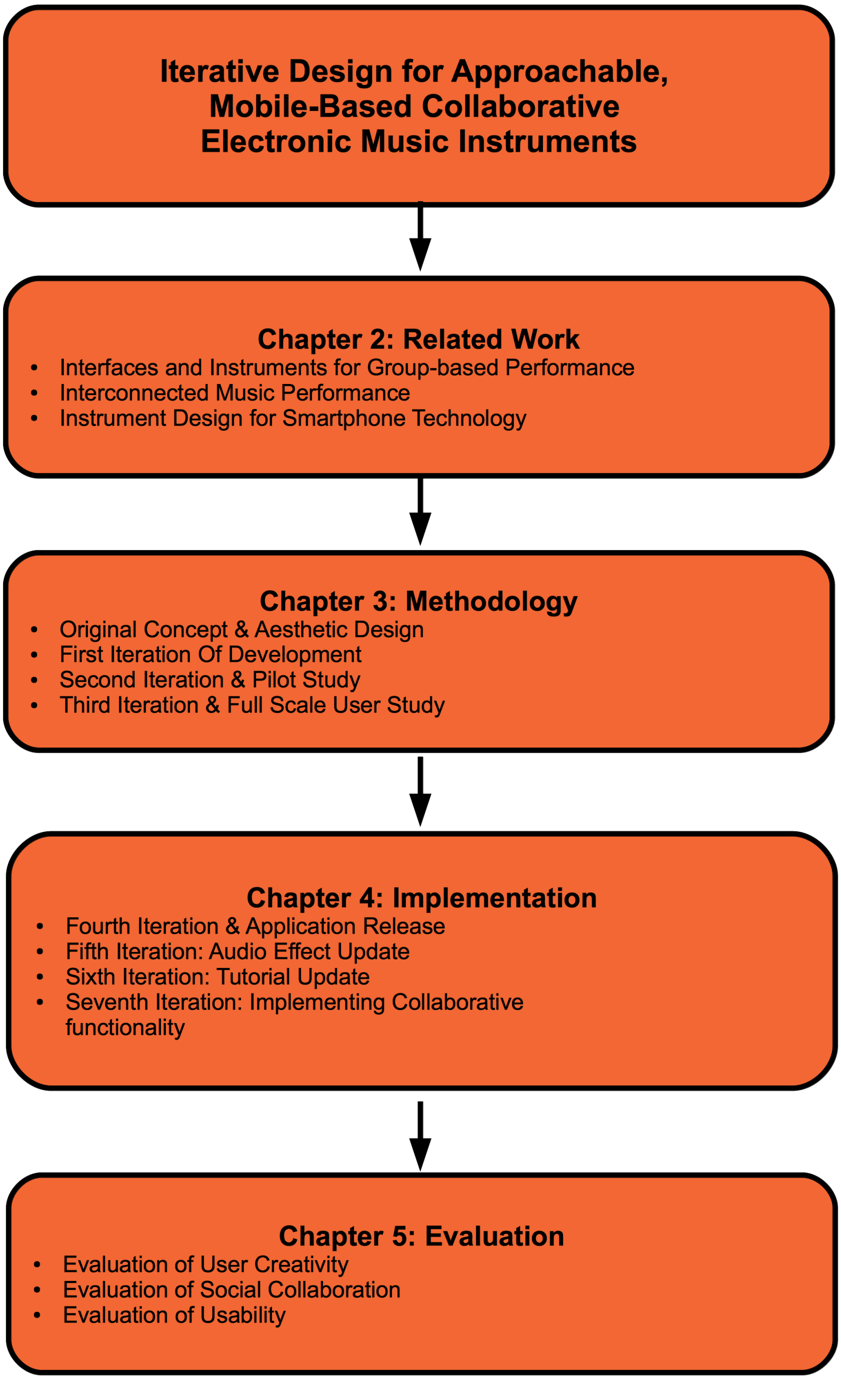


Figure 1.1: Overview of thesis structure.

Chapter 2

Related Work

This thesis builds upon multiple different fields of inquiry, theory and practices. By drawing on past work in related fields to this research we can create an informed framework for which to guide the development of a resulting electronic musical instrument. In addition to prior experience, many related works have informed the framework outlined in section 1.1. By analysing the surrounding literature we can understand key problems and identify potential solutions before development begins.

The primary approach of this design process will be based upon iterative design methodology. Nielson suggests that the most significant improvements to the usability of an interface will occur within the first few iterations of design [7]. This process of iterative design will result in a much more usable (and validated) interface. By building upon a foundation of iterations before adding upward creative capabilities (for more experienced musicians), the goals of this research will be achieved. Successive testing and iterating through designs, as informed by user testing and feedback, will result in an electronic instrument that achieves the goals of the research. The importance of this methodology is reflected in the first criterion expressed in the design framework - *user informed design*.

Following is a summary of related literature and works of the three main areas of study in which this research intersects. Each of these areas will be discussed in depth in order to provide a comprehensive academic context for this thesis. The areas are: interfaces and instruments for group based performance, instrument design for smartphone and tablet computers, and interconnected musical performance. Although there are frequent overlaps between the fields mentioned, the literature has been categorised in order to illustrate the primary contextual contribution to this thesis.

2.1 Interfaces and Instruments for Group Performance

Sacks argues that "A primary function of music is collective and communal, to bring and bind people together." [8]. The social experience of creating music collaboratively is perhaps one of the most exciting parts of creating music; however trying to recreate this social experience with novices involved is a more difficult task, as novices may not be aware of subtle intricacies of performer social cues or dynamics. As such, there is a plethora of academic work that highlights the primary issues in this field and positing strategies for group based performance in attempt to reproduce the social experiences normally reserved for expert musicians.

A seminal work in this field, 'Contexts of Collaborative Musical Experiences' by Tina Blaine and Sidney Fels compares notable works in the field of collaborative musical interface design, whilst simultaneously positing criteria for evaluating collaborative interfaces [9]. Through analysing the works in question, Fels and Blaine present a finding that some of the criteria appear to be inherently related to each other. For example, the ability for a user to become an expert performer with an interface/instrument is inversely proportional to the learning curve of the interface/instrument. Similarly, interfaces and instruments that have a low level of directed interaction, or capacity for users to decide their own musical intention, all feature relatively low learning curves in order to make them inclusive for novice users. From their analysis, Fels and Blaine argue that 'when designing for novice players ... the overall experience takes precedence over the generation of music itself.' [9] This idea informs the framework outlined, specifically, in the value of *active interaction*.

The idea of giving novices musical experience through group based instruments and interfaces has been explored with multiple different approaches. Interfaces such as Beatbugs [10] and PLOrk [11] provide directed interaction in the form of a conductor-type director leading musicians (either experienced or not) through a musical direction. This approach is useful in providing the users with instruction, effectively lowering the learning curve, but ultimately works to constrain their interaction. Figure 2.1 shows BeatBugs and figure 2.2 shows PLORk. The framework value of *ease of access* is informed by this in two ways. Firstly, the necessity of having a director of sorts would make the resulting instrument less accessible to users who do not have access to the director. Secondly, the direction helps to lower the barrier to entry. In order to maximise the ease of access for users, we must find a way to provide direction to the user, without impeding their musical freedom. A sensible way to do this would be to provide the user with a tutorial type option that does not require a human to control their musical direction, attempting to strike a balance between the benefits and pitfalls of strongly directed user interaction.

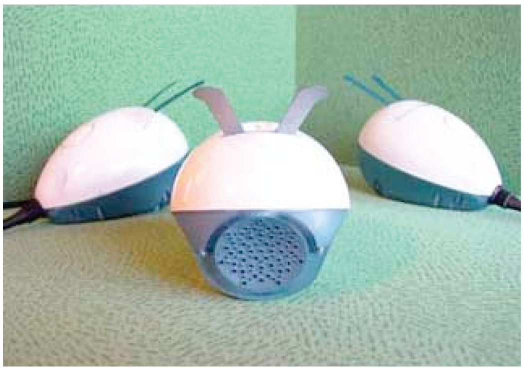


Figure 2.1: Beatbugs interface [10].

Figure 2.2: Princeton Laptop Orchestra (PLOrk) [11].

The specific *socially collaborative* element of the framework has also been informed regards to related work. The number of users an instrument is created for will affect the interface/instrument. For example, turn-based approaches to musical interface design have been shown to be successful as a pedagogical tool for familiarising users with the interface, this approach is useful for larger groups of users (above 4 or 5) as it limits the density of the resulting sound [12]. An analog of this would be akin to jazz musicians taking turns at performing a solo. Another approach is to delegate sonic responsibilities to users as was done with Musical Trinkets [13]. An analog of this would be how different musicians in a band play different instruments, each with their own role. All three of these separate approaches, however, limit the user's creative agency in different ways. For the sake of simplicity, the framework's value of *ease of access* will not enforce either the turn taking social approach or the delegation approach. However, by virtue of the *socially collaborative* aspect of the framework, these behaviours will be able to naturally arise through social interaction between the users. In order to facilitate the ability for these behaviours to naturally arise, the socially collaborative aspect will only work over a local network, encouraging a close physical proximity between the users to promote the *social collaboration*.

An approach to making large scale interactive musical experiences has been explored in Tweetdreams [14]. In this work, users 'tweet' words, via the social networking platform Twitter, which are then decoded into musical expressions. Although this way of approaching large scale collaborative interface design is novel, there is a lack of technological transparency; the user may not be able to understand the direct relationship between their interaction and the resulting musical output. This approach is also explored in 'Dynamic Social Interaction in a Collective Mobile Music Performance'. In this work, users are able to use smartphones to transform gesture data from the inbuilt accelerometers into musical data. However, the work also exhibits the problem of obscuring the causal relationship between interaction and output [15]. To make sure the users are able to engage with the work quickly and are able to understand the relationship their interaction has to the overall creative musical output, the group based aspect of this research will be limited to a small number of users. This is a contributing factor to the inclusion of the framework value of *technical transparency*. By providing users with direct audio feedback responsive to their interaction, they will be able to quickly learn how the interface is navigated, and ideally, learn basic concepts of electronic music theory.

The final framework value, *artistic and aesthetic validity,* is incredibly important in shaping what the resultant work will be like. The finished work should be able to stand alongside other musical works, as the development will feature a process of selecting and defining musical parameters that a user will be able to manipulate. This process bares a distinct similarity to the the creative constraining and selection in musical composition (or other forms of art) and arguably functions as a form of composition in itself. Robert Rowe discusses interactive music works as inhabiting a space between computer technology, performer and composer, whereby the creative output rests in the definition of the user interaction with the computer in order to experience music. Or more succinctly,

> "By transferring musical knowledge to a computer program and compositional responsibility to performers onstage... the composer of interactive works explores the creative potentials of the new technology at the same time that he establishes an

> engaging and fruitful context for the collaboration of humans and computers.” [16]

2.2 Interconnected Music Performance

Although strongly related to group based performance paradigms, this section on related work from the field of networked computer music performance will focus upon the more technical approaches and history of networked computer music performance. The League of Automatic Composers and the Hub are considered the earliest networked music performance groups, which set a precedent for the works that followed [17]. Weinberg remarks that the contribution of both the League and the Hub’s Network Computer Music introduced the personal computer as a resource for musical networks [18]. Before this time, the concept of an interconnected music performance was only in its infancy and as such, both groups required a large amount of technical understanding and ‘hacking’ in order to get things to work. The interconnectivity of the nodes in these networks were limited to sending instrument control data to one another. The second iteration of the Hub involved using the MIDI protocol to simplify the communication.

The type of connected musical networks that have been created since, due in part to developments within personal computers, internet technology and subsequently, smartphone technology, have demonstrated a plethora of different network topologies and strategies for creating interconnected musical networks. Emerging technologies for controlling expressivity with more precise control data have been explored with Open

Sound Control (OSC). Additionally, with increasing internet bandwidths and speeds, interconnected music networks have been able to send audio between computers, traversing continents, in real-time with minimal latency [19] or connect human performers with mechatronic instruments [20]. The possibilities of these networks, however, are beyond the scope of this research.

Due to the close proximity of performers, as prescribed by the suggested framework, network capabilities more closely resembling MIDI clock synchronisation and small amounts of control data will be sent between devices. This kind of loose networking will attempt to allow users' social interaction to be the primary networking method, with a technological networking solution functioning only to assist users. An example of this kind of technology can be seen with Korg's Wireless Sync-start Technology (WIST) [21]. This technology allows multiple iOS devices to synchronise metronomes between devices to emulate the MIDI clock synchronisation found on some hardware. This technology functions primarily for single users to create networked instruments between devices and appears to be predominantly for applications targeted towards more experienced musicians. However, this can be developed to allow multiple users (of any skill level) on multiple devices the same opportunity.

2.3 Instrument Design for Smartphone and Tablet

Over the last decade, the proliferation and technological advancements of smartphone and tablet computing has allowed for an unprecedented access to small applications or "apps" as a new form of digital media. Apps that attempt to create a new musical experiences for users can be commercially successful, as demonstrated by Smule's Ocarina [22], Smule's Magic Fiddle [23], SpeedDial [24] and Orphion [25]. However, there is a distinction distinction between active and passive musical experience applications. Although a multitude of apps exist that alter a person's listening experience, and the effect of augmented audio applications is both novel and exciting, this research is primarily focused upon applications where the user is actively shaping their personal musical experience through control of the creative process itself.

An inspection of the current climate of apps for active musical experience reveals that the market is predominantly fractured into four types of application, applications ported from hardware, skeuomorphic applications, game-based audio applications and smartphone specific applications. The following subsections discuss these different types of application, specifically the relevance, merit and flaws of each with special attention paid to notable examples from each of the categories. It is important to note that these categories are not mutually exclusive. In many cases, the applications discussed feature elements of multiple categories or transgress the boundaries completely. However, the utility in imposing these categories relies in understanding the benefits and pitfalls of features endemic to the categories and understanding them as design frameworks. This knowledge will be used in the creation of an application that utilises the best features of all the categories, and will aid in determining the values of the framework for designing interactive musical apps.

2.3.1 Applications Ported from Hardware

The first of these categories is *applications ported from other hardware*. Although similar technology to smartphones utilising touchscreen interfaces had been developed specifically for musical purposes before the advent of the mass proliferation of smartphone and tablet computers, such as the ReacTable [26] and Jazzmutant Lemur [27] these technologies only achieved limited success. This is primarily due to advent of smartphones and tablets as powerful computers with multi functionality (as cellphones, planners), existing distribution networks of App stores, and lower cost barrier to entry. This provides a much more accessible platform for creating a new musical experience. As such both ReacTable and Jazzmutant released iOS versions of the software that was previously locked to their proprietary hardware [28, 29]. The price of both these applications with their native platforms far exceeds the price of a smartphone with the ported software, however, there is a trade-off with this lower barrier to entry. A figure showing both Reachable and Lemur in their original hardware contexts can be seen in figure 2.3.

The reality of porting these kinds of applications is that the focus is on the inherent similarity between the two different hardware contexts (being the original hardware and the smartphone/tablet). This however, implies that that the application can be ported with only minimal differences between the hardware, or minor features lost in the transition, which may not be completely true. For example, the Reactable Mobile app does not allow for the use of the physical fiducial markers or the collaborative performance element that were both contributing factors to the success of the

Figure 2.3: ReacTable [26] and Jazzmutant Lemur [27] original hardware contexts. ReacTable is pictured on the left. Jazzmutant Lemur is pictured on the right.

original Reactable. This, combined with the smaller screen makes the performative aspect of the application dramatically different to that of the original. While only one example, this illustrates an issue with this category of applications. The focus being on the similarity between the two hardwares negates the point of difference that may have originally made the original context a better choice.

2.3.2 Skeuomorphic Applications

The second category is that of skeuomorphic applications that attempt to replicate existing instruments. Although strongly related to the prior category, a distinction can be made in that skeuomorphic applications are based on either traditional acoustic instruments or popular analog electronic instruments (such as drum machines and synthesisers). The allure of these kinds of instruments / applications is that they build upon strong foundations, frameworks and metaphors of musical understanding. Kell and Wanderley suggest that the most prevalent control schemes for musical apps are piano (accounting for 68 of the 337 apps they investigated) followed by DJ apps (accounting for 32 of the 337 apps they investigated) [30]. While these kinds of applications can be successful, due to their metaphors, they may also become problematic due to the difference in tactility between the physical instrument and the application. In many cases, the instrument modelled in the application has decades, if not centuries of haptic interaction that has guided user interaction. This history of haptic interaction is inevitably lost in the transition to touch screen. An example of this would be the comparison between Korg's hardware synthesiser, the MS-20 [4], and its iOS skeuomorph, iMS-20 [31]. Although these two instruments are visually similar, the iOS version (although significantly cheaper), features various user interface elements that do not translate smoothly to an iOS application. Figure 2.4 shows Korg's iMS-20 application.

Figure 2.4: Korg's iMS-20 Application based on the MS20 Hardware synthesizer [4].

Firstly, both the hardware and the application feature a piano-keyboard user interface, which is a sensible element for a hardware synthesiser to possess, in part because the user interacting with the hardware is able to delimit the keys with their sense of touch. This afforded precision is difficult to achieve on a touch screen without a visual referent. Secondly, the hardware and iOS interfaces both feature rotary knobs that require a circular motion. This motion is difficult to achieve without tactile feedback. Due to the difficulty of achieving the motion, there is no standardised method of interacting with these as user interface (UI) elements, yet a large amount of the skeuomorphic applications (not just Korg's iMS-20) feature rotary knobs (or similar UI elements). It is often the case that hardware interfaces are effective because they are purposely designed to achieve a specific goal. Translating these musical metaphors may allow for familiarity for the user (if the metaphor is sufficiently recognisable), however they may not be more expressive than metaphors created specifically for smartphone/tablet hardware.

By investigating some of the skeuomorphic user interface elements that do translate well to smartphone technology we are able to adopt metaphors of understanding that will allow users to engage with an application intuitively. This is a key concept in how this research will approach the design of an intuitive musical application.

2.3.3 Game-based Applications

The third category is musical applications that are strongly tied to game paradigms, language and semiotics, for example the work of Lucky Frame, specifically Pugs Luv Beats [32], Bad Hotel [33] and Wave Trip [34]. These hybrid works are successful in that they make musical experiences available to people of varying prior musical experience. However, the the primary goal of these applications is as a game, which is distinctly different to the rest of the related applications. For these reasons they all exhibit game structures, specifically in orienting the user towards achieving a goal as specifically mandated by the application in order to advance within the game structure. This is important in games because it provides the user with an incentivised experience, which encourages the user to play, although only with a single directive orientation. For this reason, understanding these applications as instruments is a potentially dubious territory. Although they allow the user to create music at their own leisure, by merely suggesting a direction for the user, the active process of play for the sake of musical output is pushed into the background, which we have argued should be the primary function of a musical instrument application. Figure 2.5 shows the main play screen for Pugs Luv Beats.

Additionally, the musical expression afforded by these types of applications is skewed in favour of providing a cohesive aesthetic more in line with the art direction of the game itself. Although this approach is good at giving new users a more rigid musical direction, it costs more experienced users a creative freedom. As Fels and Blaine discuss, "The trade-off in determining the appropriate balance of complexity and expressivity of an interface is not easily resolved." [9]. This illustrates that although one of our values in our framework is *aesthetic and artistic integrity*, these value is only useful if it does not noticeably affect the user experience in a way that prevents achieving the primary research goal.

Figure 2.5: Lucky Frame's Pug Luv Beats game-based music application [32].

2.3.4 Smartphone Specific Instrument Applications

The fourth category is applications designed specifically for the device itself, taking into consideration the in-built hardware and finding novel ways to utilise this technology to create new paradigms of musical interaction. This last category is by far the smallest category, despite being the most interesting from an instrument design perspective. The former three categories take their methods of interaction predominantly from other hardware or systems, but by drawing upon the tacitly standardised ways of interacting with the hardware itself. By subverting these standardised interfacing techniques, a handful of applications have become wildly commercially successful at creating new forms of instruments and interfaces for musical experiences. An example of this is Smule's Ocarina app, which utilises the microphone to detect users blowing into the inbuilt microphone, to simulate a real ocarina [22]. Although, Ocarina could be considered skeuomorphic, the additional musical parameter mapping to blow intensity represents an interface solution designed specifically with the hardware of a smartphone in mind.

Similarly, the authors of 'Characteristics of Pressure-Based Input for Mobile Devices' identified that the inability to sense touch pressure was limiting expressivity of musical instruments designed for touch interfaces and demonstrated ways to extract pressure information using user touch surface area [35]. Orphion implemented a similar technique by measuring the surface area of a users' touch in order to approximate touch gestures and essentially add an additional degree of musical expressivity by manipulating the specific hardware (and its perceived limitations) [25]. Both Orphion and Ocarina can be seen in figure 2.6.

Although these specific *extended techniques* of smartphone / tablet instruments as demonstrated by Ocarina and Orphion are perhaps not the leading factors that lead to their market success, they are definitely indicative of good user interface design by way of innovative problem solving. By extending and manipulating the commonly accepted modes of interaction with smartphones and tablets, there is the possibility of increasing the degree of expressivity of smartphone and tablet based instruments.

Figure 2.6: *Smartphone specific applications Ocarina [22] and Orphion [25]. Ocarina is pictured on the left. Orphion is pictured on the right.*

Atau Tanaka argues that this approach brings the realm of hidden musical affordances of the technology to a wider audience, who may have been previously unaware that the technology could be manipulated in such a way, which will naturally make for a more intriguing user experience [36].

2.4 Summary

In summary, the intersection of multiple fields of study has provided a suitable basis for the construction of a new musical instrument/interface application. By understanding the fields surrounding the research goal, a framework has been created that will guide the development of this instrument that lowers the barrier to entry for electronic music and inspires musical creativity in potential and existing electronic musicians. Applying this framework and understanding the potential issues that surround the choice in adopting specific approaches to the application design will result in an application that is able to achieve the goals of this research. The resulting original concept for an application based upon this investigation of related work, that will implement the framework described, in order to achieve the goal of this research is discussed in section 3.1.

Chapter 3

Methodology

This chapter will cover the original concept and aesthetic design for the application central to the research. It will discuss the first three iterations of the application development, including the designs, concepts and justification that will guide further application development leading to a publicly available distribution of the resultant application. This methodology is primary entered around an iterative design process informed by user studies and user feedback to shape the direction of the application development.

3.1 Original Concept

The original concept for the application is to have a visual grid upon which users are able to place nodes. These nodes will represent musical notes (or musical events), with their co-ordinates within a defined grid to express the musical parameters of pitch (upon the x-axis) and velocity/volume (upon the y-axis). This mapping decision was informed by Kell and Wanderley's mapping analysis of musical applications [30]. In their analysis, they found that the mapping of pitch horizontally from left to right and volume from bottom to top were the most common mapping strategies for musical applications. Furthermore, they suggest that these mappings are common for applications because they are intuitive to users, which is an aim of this interface. Additionally, informed by their mapping analysis, user node inputs will be quantised to the tics of the grid in order to create a discrete musical mapping. This means that users do not have to be as precise in their node placement (and subsequent associated musical event), which naturally inhibits the musical range of the application. However, by adjusting the amount of tics on the grid, iterations of the application will aim to find a balance between musical range (of notes and volumes) and quantisation area.

As nodes are input into the system they will be recorded to allow for users to create collections of these nodes. The user will be able to turn these collections of nodes will into looping sequences of the defined musical events. The looped sequences will make use of a temporal pulse to playback a temporally quantised version of the originally input sequence. By quantising the user input in the time domain, the user is able to easily create music with a strong metre without requiring impeccable musical timing. This will echo the strong metre of popular electronic music in order to present familiar musical forms to novice users and further define the aesthetic of the application.

The user will be able to create up to 4 separate sequences simultaneously. Each of these sequences will be distinguished by a different colour and different sound sample for their sound events.

3.1.1 Aesthetic Design

The core of the original concept is tied to a strong aesthetic vision for the application. This is in order to present the application as a work of art in itself. The application will adhere to a minimalist aesthetic to prevent confusing users with needless visual clutter or complex interactions. An exception to this will be if an interaction or visual elements obfuscates an underlying electronic music creation process. Where possible, the application will attempt to balance the minimal user interface (UI) and user experience(UX) aesthetic with elements that seek to inform and educate the user to the underlying electronic music processes involved. This will help to engage novice musicians with the application and ideally inform them about key concepts involved in electronic music creation.

The primary aesthetic theme of the application is based upon outer space. The theme will work to evoke an exploratory attitude from the user. Simultaneously, the resulting audio will bear similar musical tropes from science fiction depictions of space. This theme is a metaphor for the expansiveness of human creativity and will aim to engage the user with the application, whilst also creating an aesthetic musical and visual direction to the user, ideally, with only minimal effect on the creative element afforded by the application.

In addition to the instruments being distinguished by colour, modified planetary symbols were assigned to further relate to the theme, whilst helping the user distinguish between the the instruments. The symbols chosen are variations of the symbols for Pluto, Jupiter, Saturn and Ceres. Where possible, if words will obfuscate meaning from the user or otherwise make it more difficult for the user to grasp a concept, symbols related to the theme of the application will be used. Although not one of the key values in our framework, this deference to aesthetic theme is tied into both the values of *aesthetic and artistic integrity* and *ease of accessibility*.

3.2 First Iteration

The first iteration of the application was developed for Mac OS X. It was built with a combination of openFrameworks[1] for the user interface, ChucK [37] for the application logic and Ableton Live 8[2] for the sound design. The underlying logic of this iteration provides proof of concept for the game-type logic that will be developed upon in later iterations.

The user interface features four coloured buttons to allow the user to select and delete sequences. A short press on the coloured button will complete an active sequence, or change the active sequence to the chosen colour. A long press on the coloured button will remove the sequence corresponding to that colour.

The application features two different methods for interaction. The first method of interaction utilises the inbuilt trackpad of a computer to allow users to change between instrument sequences and place the nodes for the sequences. The second method of interacting features an iOS application running on an iPod Touch using accelerometer data to control a cursor on the Mac OS X application.

In addition to the primary application, an iOS application was devel-

[1] `www.openframeworks.cc`
[2] `www.ableton.com`

oped to provide a more novel and expressive way for users to interact with the application. The iOS application used Open Sound Control (OSC) protocol [38] over a User Datagram Protocol (UDP) network to communicate to the OS X application. The visual aesthetic of the iOS application was developed to mirror the Mac OS X version. A screenshot of the application running on an iPod Touch can be seen in figure 3.1. The iOS application featured a similar control scheme to the Mac OS X application, with short button presses to change instrument and long button presses to delete sequences. An additional two buttons were added to the iOS interface to allow the user to add nodes and to complete sequences (symbolised with an addition and a loop symbol, respectively).

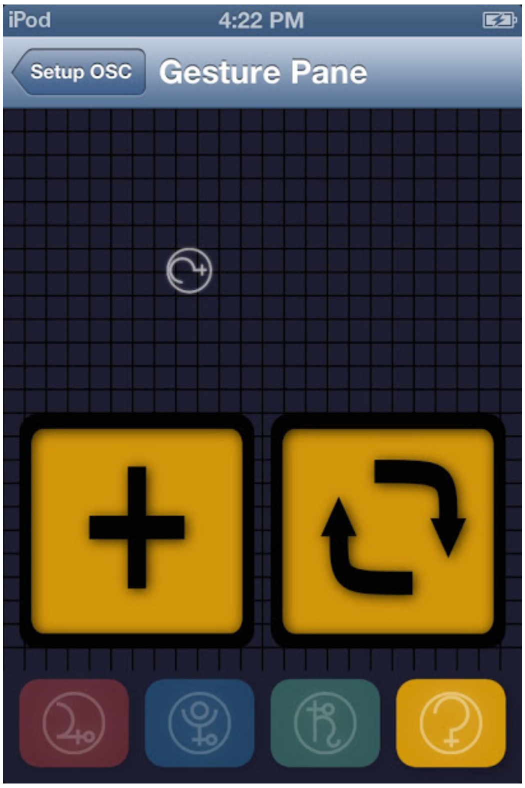


Figure 3.1: Screenshot of the first iteration with accelerometer control on iPod Touch.

3.2.1 Conformal Mapping Accelerometer Input

Controlling a cursor with the raw accelerometer data received from the iOS device proved to be difficult and unintuitive. The main issues contributing to this difficulty are excessive jitter and the difficulty of reaching the corners of the plane the cursor could traverse (as can be seen in the heat map in figure 3.2. These issues can be mitigated by processing the raw data. To smooth the accelerometer input to avoid jitter the following low pass filter is used:

$$x_t = \alpha \times x_{in} + (1 - \alpha) \times x_{t-1} \tag{3.1}$$

The value for α determines the balance between responsiveness and smoothness. As α approaches 1 the resulting output will respond smoothly, but will take longer to move the cursor and require more deliberate physical gestures. Conversely, as α approaches 0, the cursor will appear more jittery, but will respond much more quickly. After experimenting with different values for α the final chosen value was $\alpha = 0.8$, with an accelerometer reading interval of 33 milliseconds. This value gives a good balance of responsiveness to the gestural input, whilst simultaneously smoothing the accelerometer enough to avoid jitter.

Considering it is difficult to get readings of maximum acceleration due to gravity on perpendicular axis simultaneously (as can be seen in figure 3.2), a conformal mapping was used to convert the x and y axis accelerometer readings to co-ordinates on a square plane.

The different methods for conforming a circle to a square focus on either preserving the angle or the magnitude of the input vector, but not

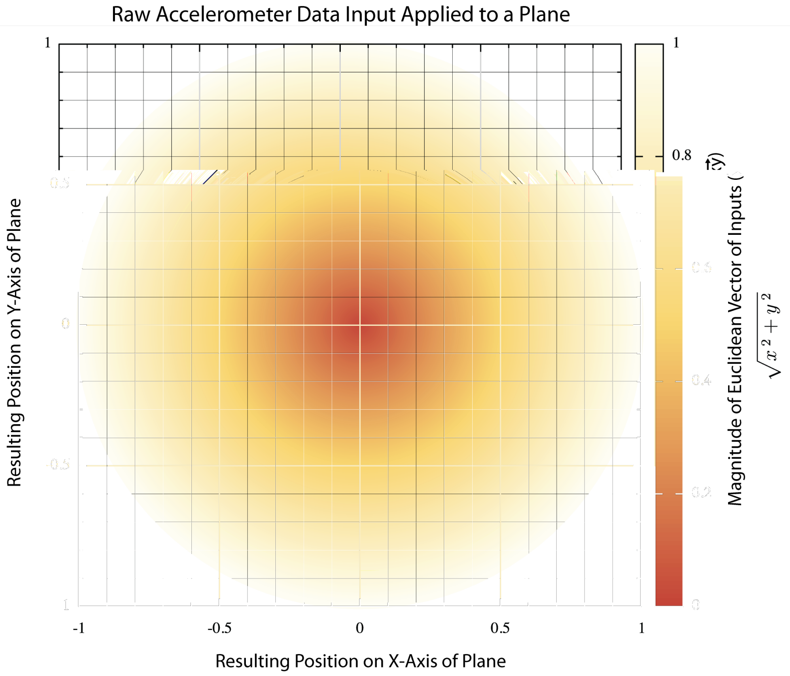


Figure 3.2: Heatmap of raw accelerometer input. This graph shows how the magnitude of the x-axis & y-axis accelerometer readings translate to a plane. This shows that using the accelerometer for tilt leaves areas in the corners of the plane that are difficult to navigate.

both simultaneously. The preservation of angle was considered more important in this application than the preservation of magnitude and, as such, the mapping that was used loses magnitude precision of the device tilt, whilst preserving the tilt angle. The mapping used is based upon the upon the equation by Nowell in 'Mapping a Square to a Circle' [39]. The mapping of the x-coordinate (x') from the filtered data of the accelerometers x-axis (x) is:

$$x' = f(x) \cdot \sqrt{\frac{x^2}{1 - \frac{y^2}{2}}} \tag{3.2}$$

Where,

$$f(x) = \begin{cases} 1.0 & \text{if } x \geq 0; \\ -1.0 & \text{if } x < 0. \end{cases}$$

The mapping of the y-coordinate (y′) from the filtered data of the accelerometers y-axis (y) is:

$$y' = f(y) \cdot \sqrt{\frac{y^2}{1 - \frac{x^2}{2}}} \tag{3.3}$$

Where,

$$f(y) = \begin{cases} 1.5 & \text{if } y \geq 0; \\ -1.0 & \text{if } y < 0. \end{cases}$$

$f(y)$ compensates for the increase in difficulty of tilting the device away from the user and to make sure the tilt does not exceed the viewing angle by skewing the mapping. A heat-map for visualising the conformal mapping can be seen in figure 3.3.

The code implementation of the conformal mapping can be seen in figure 3.4. The `calculatePosition` method is set up as the callback method for the accelerometer reading. The method argument is the data passed from the accelerometer, of type CMAcceleration. The accelerometer data is constrained before the mapping is performed. The result of the mapping is unsigned, so the sign of the accelerometer values must be applied again. If the y value is negative, it is scaled by 1.5, as per the mapping algorithm. The result of this is then constrained to between -1–1 and translated to a cursor position.

If the equation for $f(x)$ is replaced with a similar skewing of the input data as afforded by $f(y)$, the output could potentially compensate for

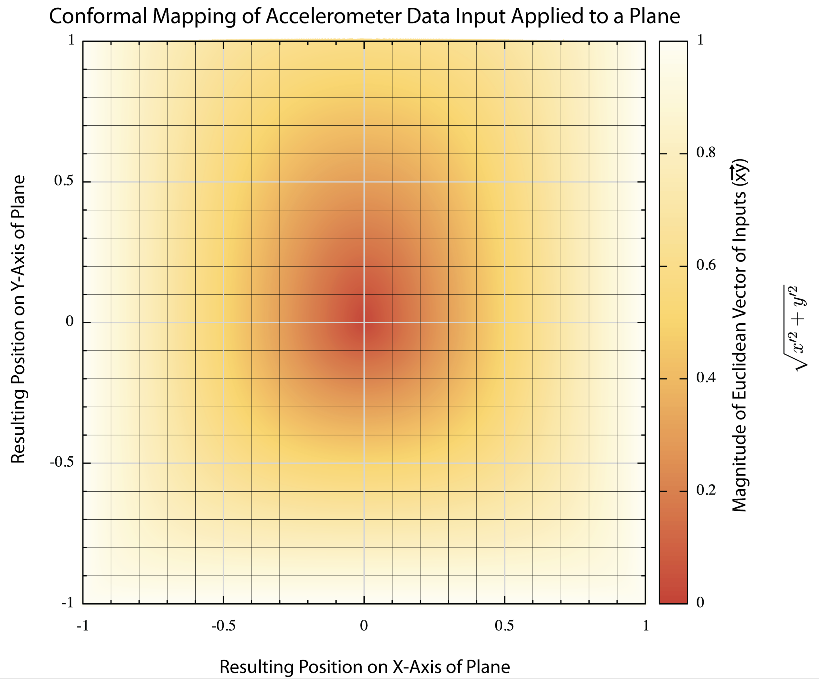


Figure 3.3: Heatmap after conformal mapping of accelerometer input. This graph shows how the areas of the plane that are difficult to access with raw accelerometer data (as shown in figure 3.2) are easily accessible by the conformal mapping, which transforms the magnitude of the x-axis & y-axis accelerometer readings to allow easier traversal of the plane.

the difference in wrist mobility related to the handedness of the user (for example, tilting a right hand (palm facing upwards) to the right is considerably harder than tilting a right hand to the left).

Using the accelerometer in this way does not account for shaking the device or moving it quickly. Some of this is resolved by the filtering of the raw data, but by employing an additional layer of processing (using the gyrometer on the device in combination with the accelerometer) it is possible to filter out acceleration that does not occur due to a change in the tilt of the device.

```
-(void)calculatePosition:(CMAcceleration)acceleration{
    //Receive a CMAcceleration that has
    // been smoothed by the low pass filter

    //Constrain the acceleration
    _acceleration.x = fminf(_acceleration.x, 0.95);
    _acceleration.y = fminf(_acceleration.y, 0.95);

    //Conform this to a square
   float _positionX = sqrt(pow(_acceleration.x,2)
            /(1.0 - ((pow(_acceleration.y,2))/2.0)));

   float _positionY = sqrt(pow(_acceleration.y,2)
            /(1.0 - ((pow(_acceleration.x,2))/2.0)));

    //make sure it is signed correctly...
    //constrain between 1 & 0.
    if (acceleration.x < 0) _positionX *= -1;
    if (acceleration.y < 0) _positionY *= -1.5;

    //Constrain values between -1 & 1
    _positionX = [Utilities constrain:_positionX
                 betweenMinimum:-1.0
                     andMaximum:1.0];
    _positionY = [Utilities constrain:_positionY
                      betweenMinimum:-1.0
                          andMaximum:1.0];

    //map value to the screen position (from centre of screen)
    _positionX = (_positionX * _screenWidth/2.0)
        + _screenWidth/2.0;
    _positionY = (_positionY * _screenHeight/2.0)
        + _screenHeight/2.0;

    //Position the cursor at the value
    _cursor.position = CGPointMake(_positionX ,_positionY);
  }
```

Figure 3.4: Code of the method developed for conformal mapping accelerometer input to cursor position on a plane.

Overall, using an accelerometer in this way is not often explored in musical instrument design. A more common way of utilising the accelerometer is to map the input to audio effect parameters, for example the resonance and cut-off of a filter. An example of this can be seen in apps such as Musyc[3] and flail[4]. The application of accelerometer in this way will be explored in the future work section.

3.2.2 Iteration Analysis

The first iteration, as a proof of concept, reveals issues with the original idea of the application and refinements necessary for further development. The most prominent issue is that the user interaction is split between the iOS device and the Mac OS X device. A consequence of this is that the obviousness of how to interact with the application is substantially more complicated with the addition of the iOS device. Personal testing revealed that due to simplicity, interaction with the Mac OS X device alone is preferred, despite the novel method of interaction with the application afforded by the inclusion of the iOS device.

Another issue is that the accelerometer mapping is also unintuitive to users. When presented with a grid on an iOS device screen, a more intuitive way of interacting with the grid is to directly touch the screen. This is because many applications use touch interfacing as the primary method of interaction, with the accelerometer mostly used for detecting screen orientation, rather than as an active user input. As such, using the accelerometer in this way goes against the overall ease of use expressed in the objectives of the design process. For this reason, the possible implementation of the data input processing designed for the accelerometer will be explored in relation to different musical mappings.

The final issue raised by this iteration is that in both methods of interac-

[3]http://www.fingerlab.net/website/Fingerlab/Musyc.html
[4]http://heartofnoise.com/products/flail/

tion the cursor based interaction negatively impacts the ability to respond to musical events in real-time (as there is an inherent delay required in moving a cursor across a screen between UI elements). For this reason, it became difficult to make both quick and precise inputs simultaneously. Based upon this, the next iteration will test whether the application may be better suited to a multi-touch surface (such as an iOS device), rather than with a mouse and cursor.

3.3 Exploration of Accelerometer Mapping

In tangent to the primary application development, different approaches to applying the previously developed conformal mapping of accelerometer data were explored. The result of this exploration was an iOS application that took the output of the conformal mapping, in combination with the outputs of multiple touch gestures to create parameter values. The touch gestures mapped one finger swipe vertically, one finger swipe horizontally, two finger swipe vertically, two finger swipe horizontally, two finger rotate gesture and three finger rotate gesture. The parameter values represented have output ranges from 0–1023. Any change in parameter values is sent over OSC via UDP. This data was received in a ChucK application and downscaled to a value between 0–127, before being sent out as MIDI data to be received in Ableton Live and mapped to any chosen musical parameter (predominantly in the form of audio effect parameters).

The visual design of this application features concentric arcs mapped to each of the parameters. The length of each arc in degrees indicates the value of each parameter (between 10°–360°mapped to the output values between 0–1023). As the values change, the lengths of the arcs are updated to provide visual feedback to the user. Figure 3.5 displays a screenshot of this application.

Figure 3.5: Screenshot of iOS application to transform gesture data to MIDI outputs.

This application results in a much more intuitive mapping of the data inputs from the conformal mapping process. This suggests that the approach of using accelerometer data for audio effect manipulation is a better approach than the previous cursor based interaction for providing expressive and intuitive musical control. This iOS application, however, requires prior knowledge of how to set up the MIDI, OSC and the UDP connection. Consequently, this makes it relatively inaccessible for novice users. This application favours manipulation of existing audio (predominantly through effect parameter manipulation), rather than the process of actively generating sound, which is the focus of this research. As a result, this approach to mapping accelerometer and gesture data may be implemented in a later development iteration, but by itself would function as a tool targeted only to people with an understanding of the technological requirements.

3.4 Second Iteration

The second iteration of application development resolved the issue of splitting user interaction between devices by limiting the user interaction to one device. This iteration is a complete rebuild of the application for two separate platforms - iOS and Mac OS. The goal of this rebuild was to create a functionally identical application that had two distinct methods of interaction in order to compare them. There are three hardware contexts for the application, iPad and iPod Touch (for iOS) and Macbook Pro (with mouse input) for Mac OS X. The interaction methods are mouse input for Mac OS and touch screen input for iOS. The use of a touchscreen is an attempt to resolve the issue raised by the cursor based user interface in the previous iteration.

The application is built with the Cocos2D[5] game engine and libPD audio engine [40]. These engines were chosen as they work on both operating systems. This limits any differences between the application running on the different hardware contexts to their specific interaction methods

[5] `www.cocos2d-swift.org`

and screen sizes. A flowchart detailing the architecture and core mechanisms for the application can be seen in figure 3.6. The user interaction flows through to the core logic of the application, which is built upon the game engine. The game engine handles the interpretation of user interaction and the visual output. There is a two way communication between the the logic of the application and the metronome and quantisation engine. The audio engine of the application is controlled by the game engine and main logic, which, in turn outputs audio directly to the hardware of the device. A screenshot of the OS X version of the application resulting from the second iteration of development can be seen in figure 3.7.

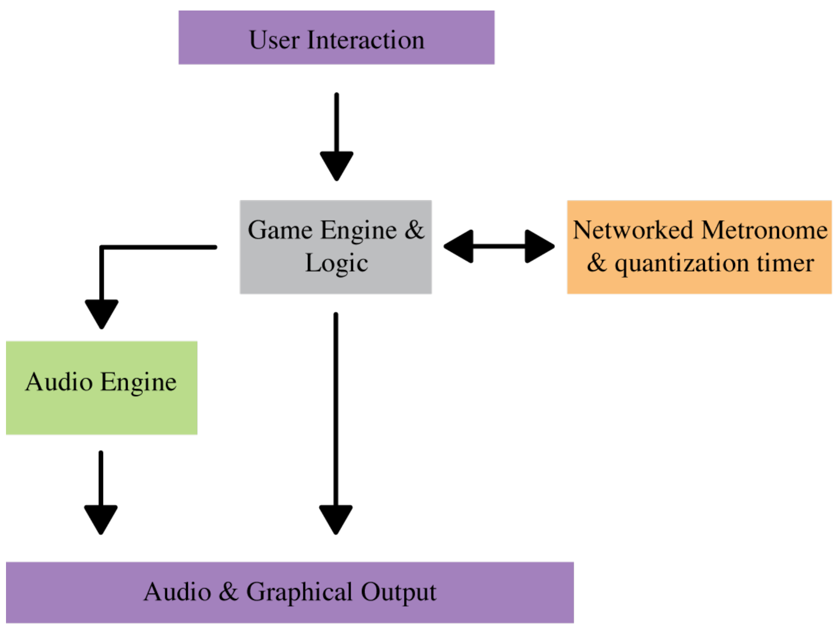


Figure 3.6: Flow chart of application architecture designed for the second iteration of application development.

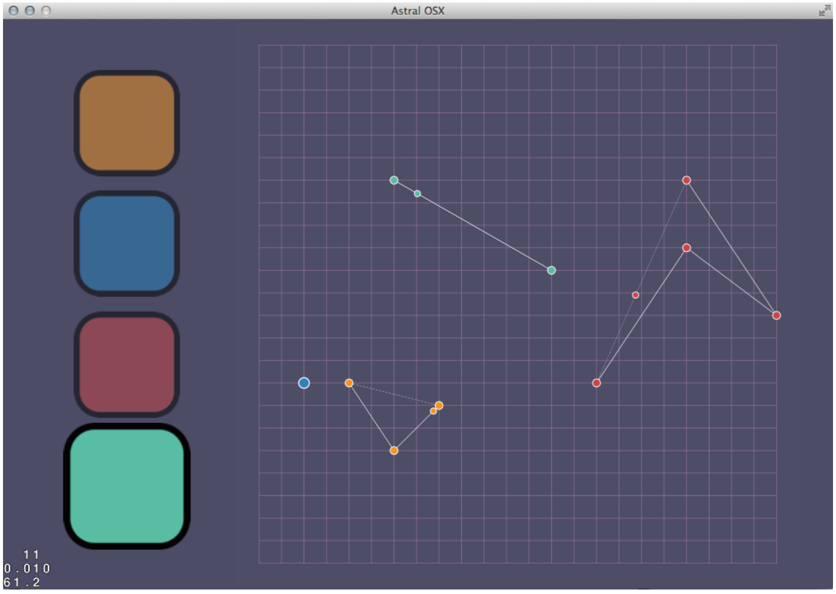


Figure 3.7: OS X version of the second iteration of application development.

Audio Generation

The audio generation in this iteration features rudimentary frequency modulation (FM) synthesis. The implementation, and specifically the overhead of utilising embedded Pure Data (in the form of libPD), consumes a majority of the processing power used by the application. The code for invoking the patch and configuring libPD is shown in figure 3.8. After initialising the `PdAudioController` object, which connects to the hardware, the `audioController` is configured as a playback device. The sample rate of 22050 and mono output were used because the application had significant overheating and audio drop outs on the iOS devices when higher sample rates and stereo audio were used. As such, the audio generation is rather limited and the resulting expressive capabilities of the audio are adversely impacted by the decision to use Pure Data instead of a lower level audio engine. The PdBase class uses a resource path pointing to the location of the 'synthesiser.pd' patch. The patch was created with the pure data application, using monophonic wavetable synthesis. In order to develop the application further, a new audio engine will have to be utilised to both allow for more musical expressivity and to lower the power demands of the application.

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions↩
    :(NSDictionary *)launchOptions
{
    self.audioController = [[PdAudioController alloc] init];
    [self.audioController configurePlaybackWithSampleRate:22050 numberChannels↩
        :1 inputEnabled:NO mixingEnabled:NO];
    [self.audioController configureTicksPerBuffer:4];
    [PdBase openFile:@"synthesizer.pd" path:[[NSBundle mainBundle] resourcePath↩
        ]];
    [self.audioController setActive:YES];

        return YES;
    }
```

Figure 3.8: Method invoked as the application begins, in order to load a Pure Data patch and run libPD as the audio engine.

The audio generation is confined to one musical scale with no ability to play notes outside of that scale or change the scale. The F# Dorian scale was chosen based upon musical aesthetics.

Metronome and Quantisation Engine

In order to temporally quantise the user input, a solid metronome and quantisation engine was required. When the application launches, a global eventScheduler is created that can either be listened to, or manipulated from elsewhere in the application architecture. The implementation for this global class can be seen in figure 3.9. The eventScheduler class has a method named `sharedBeatClock`. The `sharedBeatClock` is a class method that returns a static instance of an `eventScheduler` class if it has been initialised, otherwise it creates and initialises a new one. This has the benefit of only having one instance of the `eventScheduler`, that is accessible from anywhere else in the application that imports the `eventScheduler` header.

```
+(eventScheduler *)sharedBeatClock{
    static eventScheduler *sharedBeatClock;

    @synchronized(self)
    {
        if (!sharedBeatClock)
            sharedBeatClock = [[eventScheduler alloc]init];
        return sharedBeatClock;
    }
}
```

Figure 3.9: Code for method used to create a global instance of the eventScheduler object named sharedBeatClock that can be accessed from elsewhere in the application architecture.

The scheduler itself utilises Apple's Grand Central Dispatch[6] (GCD) to create a repeating timer that will fire at the beats per minute (BPM) rate of the application. GCD was utilised because it provides Apple's lowest-level mechanisms for creating timers and threading processes. This is required as the temporal rigidity of the metronome is important for creating a strong metric pulse, which will be crucial when syncing multiple devices together to create electronic music. The code for the scheduling of the repeating timer can be seen in figure 3.10.

This code works by creating a dispatch queue and timer. The timer is

[6] `https://developer.apple.com/library/prerelease/ios/documentation/Performance/Reference/GCD_libdispatch_Ref/index.html`

```
-(void)startTimer{
    //Create dispatch queue
    dispatch_queue_t queue = dispatch_queue_create("rhythmQueue",NULL);
    //Create a timer
    _timer = dispatch_source_create(DISPATCH_SOURCE_TYPE_TIMER, 0, 0, queue);

    //Dispatch timer with the length of the BPM in Milliseconds
    dispatch_source_set_timer(_timer,
                              dispatch_time(DISPATCH_TIME_NOW,0),
                              [self pulseInMillis] * NSEC_PER_MSEC,0);

     //Set the timer to call the broadcast method for the timer.
    dispatch_source_set_event_handler(_timer,
                                      ^{ [self broadcastPulse];});

    dispatch_resume(_timer);
}

-(void)broadcastPulse{
//Increment counter for number of beats that have occurred
    _beatcount++;
       //Create user data for this beat count to be received by the listeners
        NSDictionary  * dict = [NSDictionary dictionaryWithObject:[NSNumber ↩
            numberWithInt:_beatcount] forKey:@"beatCount"];
    //Every four beats post a whole note notification
    if (_beatcount % 4 == 0){
        [[NSNotificationCenter defaultCenter]
        postNotificationName:@"Beat" object:nil userInfo:dict];}
    //Otherwise post a quarter note notification
   else [[NSNotificationCenter defaultCenter]
    postNotificationName:@"clockPulse" object:nil userInfo:dict];
}
```

Figure 3.10: Code for methods used to invoke a repeating timer using Apple's Grand Central Dispatch (GCD) and broadcast the messages to all listeners in the application.

then added to the queue with a repeating time of the Beats Per Minute (BPM) parameter in nanoseconds. The event handler for the timer is set to the method `broadcastPulse` and the timer is resumed. The method `broadcastPulse` works by first incrementing an integer, `beatcount`. This integer is then stored in an NSDictionary, which provides the userInfo argument for an NSNotification broadcast. If the beatcount integer is a multiple of 4 (or a whole note in standard tempo), the `defaultCenter`

NSNotificationCenter will distribute a notification labelled beat. Otherwise, the defaultCenter will post a notification named clockPulse. This is because different objects within the application require different notification frequencies.

The sharedBeatClock allows users to set the BPM parameter, which is then calculated as a time interval in seconds. The BPM of the application is set at 120 BPM, as this is a standard tempo for electronic music. Additional methods in the sharedBeatClock will return the BPM value in milliseconds or seconds as requested.

The quantisation engine features methods for broadcasting pulses (as timer firings) in order to communicate to other components that subscribe to the broadcasts. The pulses that are broadcast are of two varieties; quarter notes and whole notes. Additionally, the quantisation engine records the time of the last pulse and the time of the next pulse scheduled. When an event needs to be quantised, a method compares whether the event is closer to the previous pulse or the next pulse in order to allow for both forward and backward quantisation. This is important because if the quantisation engine only allowed backward quantisation, notes that are played a fraction of a second before a pulse will be quantised to the previous pulse, which would be contrary to the user intention.

3.4.1 Pilot Study

A pilot study was conducted with this iteration of the application in order to highlight any key usability issues with the application or hardware contexts before undertaking full scale user study. The pilot study also functions to give preliminary findings as to the suitability and preference of the hardware contexts for the application. To complete the user study, users are asked to play with the application on all three hardware contexts at their leisure, after receiving a brief demonstration of how to use the application. After users have finished playing with all three contexts they are asked to complete a questionnaire pertaining to the hardware contexts, the application design and the usability of the application. This questionnaire can be found in the appendices of this thesis. Five users took part in the pilot study, reflecting a range of prior musical experience.

3.4.2 Iteration Analysis

The results of the pilot study indicate that the iPod device, in its current iteration, is not the preferred hardware context for the application. The predominant issue noted by the users is that the screen size is too small for the number of grid divisions in the iOS implementation of the application. Subsequently, this negatively affected the score for the iOS categories of “ease of achieving the full range of available interaction” and ”simplicity of interaction”. One user indicated that despite the iPod being substantially more difficult to use, the hardware itself could be used in more expressive ways, as afforded by its compact, hand-held size. While the Macbook and the iPad remained stationary, the iPod was often picked up in the user’s hands.

One user noted that at first they were unaware of the quantisation occurring in the application. They remarked that the quantisation mechanisms allowed them to feel ‘virtuosic’ with the application after only a short period of time.

Four of the five users surveyed suggested that they would like more ability to control parameters of the sound produced. Audio effects and different sound generation possibilities were the suggestions offered by the users for how to accomplish this. In addition to this suggestion, specific usability issues of the application not pertaining to any specific hardware were raised.

Users noted an inability to remove a sequence that is in the process of being drawn. Another issue is that users are unable to move nodes after placing them. The combination of these two issues impacts negatively on the user experience. If the user is making a long sequence and accidentally records an incorrect note, they are forced to finish the sequence (which deletes the previous sequence of that colour) before removing the sequence, in order to begin recording the sequence over.

An additional issue raised is the difficulty of completing and looping a sequence. In order to loop a sequence, a user must click/touch the first node of a sequence, however, if this node is obscured for any reason it becomes difficult to achieve the interaction. Additionally, with sequences with a larger number of nodes, identifying the first node of the sequence may be difficult.

The rigidity of the quantisation and the inability to change the BPM was also noted by two users (who were both experienced musicians). This can be resolved by creating global options to address application wide parameters.

Overall, the pilot study revealed that a further iteration of application development is necessary to fix these specific usability issues before a full-scale user study should be conducted.

3.5 Third Iteration

The third iteration of application development is focused upon resolving the usability issues revealed in the pilot study and improving the application stability, responsiveness and speed. Usability issues in the application have been resolved by the inclusion of more extensive options for each instrument. If a user selects an instrument, an additional menu replaces the instrument selection button. A screenshot displaying this menu can be seen in 3.11. The menu is comprised of four buttons:

- A delete button for deleting a sequence in the process of being created, or if no sequence is being created, the sequence belonging to that instrument. The delete button is symbolised by a cross.
- An add button for beginning a new sequence. If a sequence is being created this will be replaced with a loop button for finishing the current sequence. The add button is symbolised by an addition symbol. The loop button is symbolised by a looping arrow.
- A mute and un-mute button. This button will mute or un-mute the instrument from sending the notes of the sequence to the audio engine. These buttons are symbolised by a speaker indicating sound and no sound (for mute and un-mute, respectively).
- A settings menu for revealing an additional menu with more extensive options to alter sound parameters of the given instrument. This

button is symbolised with a cog. While the settings menu is active, either pressing the settings menu again or the delete button will return to the main application screen.

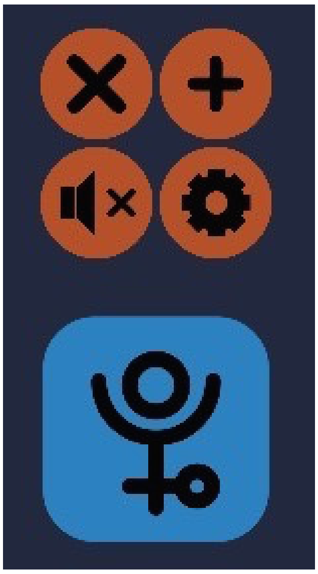


Figure 3.11: Per instrument settings added to the third iteration of the application. Bottom symbol indicates an un-selected instrument button. Once selected, the button reveals a the new instrument menu button (as seen in the top four buttons).

Additional buttons were placed on the right side of the application for global functions. A button with a dotted quaver reveals a menu for adjusting parameters of the quantisation engine. A trashcan button deletes all sequences in the application. A question mark button allows the user to watch a 20 second tutorial illustrating the core functionality of the application. These buttons can be seen in figure 3.12.

The only specific hardware change was the decrease in grid tics for the iPod Touch hardware. This makes it easier for users to select nodes on the smaller screen, however further confines the musical expressivity

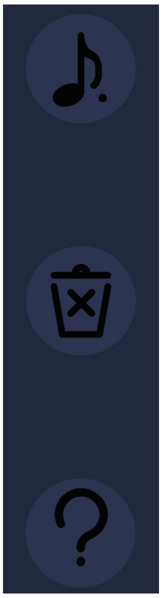


Figure 3.12: Global function buttons added to the third iteration of the developed application.

afforded to the user. In order to improve the usability and performance of the application significant changes were made to both the quantisation engine and the audio engine. A screenshot of the application from this iteration can be seen in figure 3.13.

Adding Swing to the Quantisation Engine

Based upon how Roger Linn implemented the swing functionality to the original MPC60, adding a groove or swing element to the quantisation engine would allow the resulting music to sound less metrically precise. The decision to implement Roger Linn's technique for swing is based upon a combination of aesthetic reasoning and the historical significance of the MPC in relation to hip-hop and electronic music [41]. The swing function was implemented by adding a delay to every second pulse in the quan-

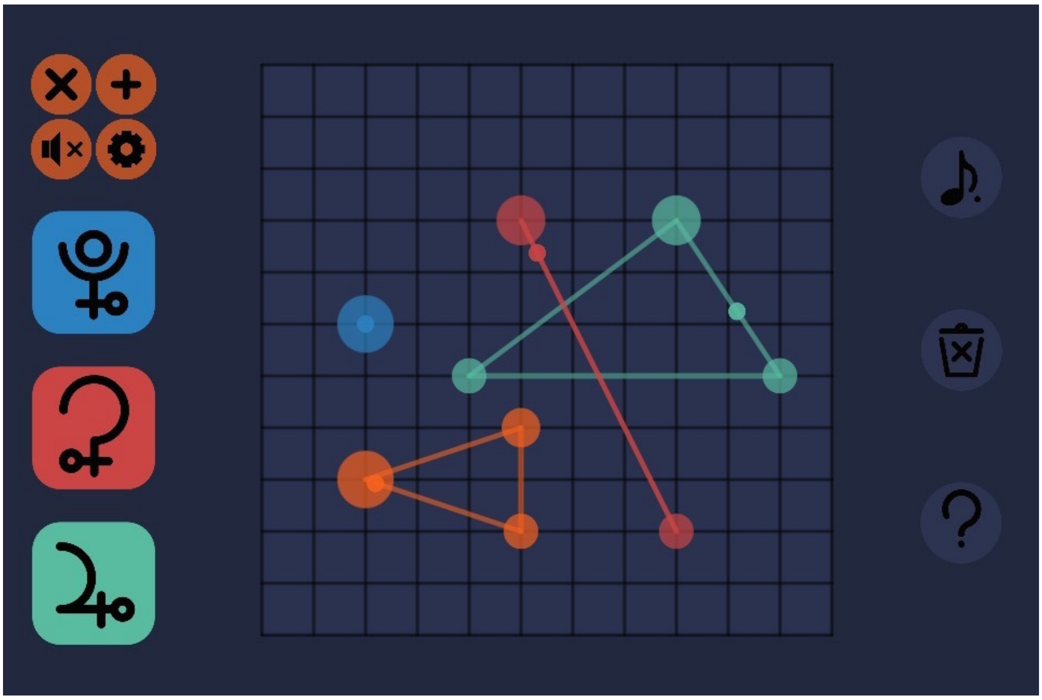


Figure 3.13: Screen shot of the result of the third iteration of application development (pictured on iPod Touch). The planetary symbols used to differentiate the instruments can be seen on the left.

tisation engine. The amount of delay is calculated as half the duration of each pulse multiplied by a swing factor (between 0–1). A fader is provided to the user in the global settings menu to allow them to choose how much swing is applied.

The code realisation of the swing mechanism can be seen in figure 3.14. It works by modifying the `broadcastPulse` method shown in figure 3.10. After the `beatcount` is incremented, a variable, `grooveDelay`, is set to 0. If the value of `beatcount` is odd, `grooveDelay` is set to equal the groove amount multiplied by the BPM in milliseconds. The pulse is then scheduled to occur after a delay of the `grooveDelay` amount (which is 0 if `beatcount` is even). Additionally, when the pulse occurs, the eventScheduler records the time of the pulse occurring and the time that the next pulse should occur. These values are used by the quantisation engine to determine whether any specified time is closer to `lastbeat` or `nextBeat`, allowing for both forward and backward quantisation.

```
-(void)broadcastPulse{
//Increment beat counter
    _beatcount++;
  //initialise variable for delay amount (in ms)
    float grooveDelay = 0;
           //Create user data for this beat count to be received by the ↩
               listeners
        NSDictionary  * dict = [NSDictionary dictionaryWithObject:
        [NSNumber numberWithInt:_beatcount] forKey:@"beatCount"];
            //Every four beats post a whole note notification
    if (_beatcount % 4 == 0){
        [[NSNotificationCenter defaultCenter] postNotificationName:@"Beat" ↩
            object:nil userInfo:dict];
    }
    //For every odd beat, calculate delay to be half the groove
    //times the BPM in milliseconds
    if (_beatcount % 2 == 1){
        delay = [self pulseInMillis] * (_groove);
    }
    //Post the notification after a delay amount (which may be 0)
     dispatch_after(dispatch_time(DISPATCH_TIME_NOW,grooveDelay * NSEC_PER_MSEC↩
         ),beatQueue,^{
        _lastBeat = [Utilities timeStamp];
        _nextBeat = _lastBeat + ([self pulseInMillis]) - grooveDelay;
    [[NSNotificationCenter defaultCenter] postNotificationName:@"clockPulse" ↩
        object:nil userInfo:dict];
    });
}
```

Figure 3.14: Code showing updated method from figure 3.10 to allow for swing to be added to the metronome.

A fader situated above the swing fader allows users to change the BPM of the quantisation timer, to allow more musical expressivity to the user. The range of this fader is 60–140 BPM, which encompasses a large variety of tempos from popular electronic music.

Audio Engine

The audio engine was written using Apple's Core Audio Framework[7]. Core Audio provides low-level access to the audio functionality on iOS and Mac OS X and was utilised because of the performance enhancements provided over the higher-level audio provided by libPD.

The audio engine works by constructing an "audiograph" - a collection of audio nodes that provide utilities for creating and processing real-time audio. The "audiograph" was constructed from four audio unit sampler nodes (AUSampler) connected to a audio mixer node (AUMixer). Each AUSampler corresponds to a separate instrument. The AUMixer sends audio to the generic audio in and out node (IONode) which connects the audio inputs and outputs to the hardware of the device. Each AUSampler features five sample presets. The first four of these are each comprised of a single sample that is pitch modulated to create the desired note (from MIDI input sent from the game logic). The samples are a sine wave, a triangle wave, a saw-tooth wave and a square wave. The final sample preset is constructed from a collection of band-pass filtered white noise samples. This preset attempts to replicate percussive sounds. These samples were chosen because they represent the building blocks of audio synthesis [42].

Each AUSampler has parameters that can be manipulated by the user in the instrument settings menu corresponding to each sampler. The user is able to select which sampler preset they will use by selecting from buttons with the corresponding depiction of the sample type (for example a

[7]https://developer.apple.com/library/ios/documentation/MusicAudio/Conceptual/CoreAudioOverview/Introduction/Introduction.html

sine-wave symbol). This mapping of sample sound to the symbol is intended to encourage users to understand the relationship between the visual shape of the waveform and the resulting sound. Figure 3.15 shows these waveform symbols/buttons. Other parameters able to be adjusted are stereo pan, octave base (allowing the user to scale the pitch of the instruments notes in octave increments), note envelope and instrument gain. The note envelope allows users to change the placement of three nodes upon an xy-grid that determines the note onset, duration and release times for the instrument. Each of the UI elements corresponding to these parameters are labelled with the correct terminology. This is to encourage users to understand what the terminology means by changing the parameters and hearing the audible result. The octave base for the iPod Touch features an additional two octaves to counteract the limiting of expressivity imposed by the smaller number of grid tics on the iPod Touch platform.

Figure 3.15: Waveform symbol buttons added to the instrument settings menu of the application developed.

3.5.1 User Study

The third iteration of development has been demonstrated on all three hardware contexts at the Sonic Engineering Expo 2014 at Victoria University of Wellington on the 17th of October 2014. A photo of the application as displayed in the expo can be seen in figure 3.16. The application attracted a significant amount of interest and a large amount of informal feedback was received. Users were also asked to participate in a user study by completing a questionnaire pertaining to the application, the hardware contexts and their personal preference of contexts. Those that agreed to take part in the study received a brief introduction to the application before being shown the in-application tutorial. Users were then asked to spend approximately 10 minutes using the application on all three hardware contexts before filling out the questionnaire. This questionnaire can be found in Appendix A. A total of 20 users completed the questionnaire.

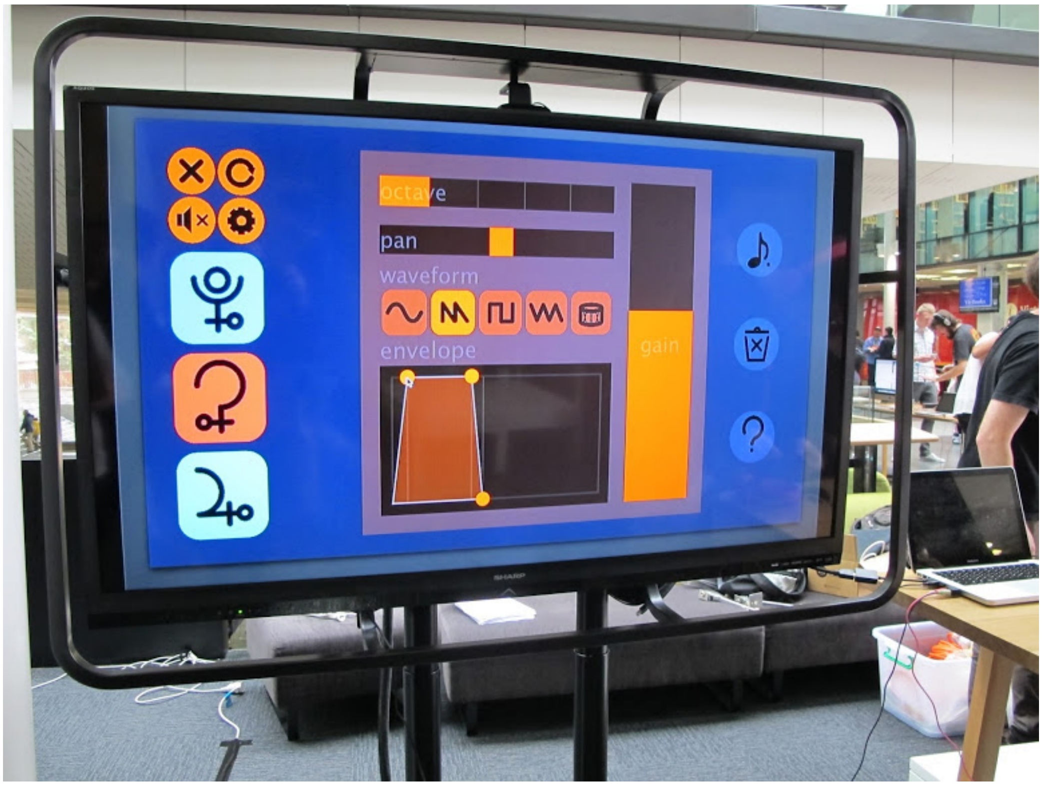


Figure 3.16: OS X version of the third iteration of application development as displayed at the Sonic Art Engineering Expo 2014.

Both the iPad and iPod hardware contexts performed better in this user study compared to the pilot study. The improvements to the application however did not translate as well to the OS X interface. Overall, the iPod Touch marginally out performed the Macbook Pro. Remarks from users suggest that the increased amount of menus to navigate (most of which were added in this iteration) makes the usage of a cursor based interaction cumbersome, in comparison to the touch based interaction of the other two devices.

3.5.2 Iteration Analysis

The questionnaire asked users as to which platform, given equal access to all three, they would use most frequently. Results indicate the iPad as the preference, followed by the iPod Touch. This was suggested to be due to the relative portability of these devices over the Macbook Pro. Although the smallest of the devices, the iPod Touch appeared to still have usability issues related to its size - especially in the new instrument menu added in this iteration. As such, the iPad appears to be the best balance of portability, comfortability and usability for this iteration of the application. The device preferences of the earlier 20 surveyed users can be seen in table 3.1.

Table 3.1: Results of the full scale user study showing the number of users who selected each hardware context as their preference for the developed application.

Category	Mac	iPad	iPod
Ease of Use	2	17	1
Most fun to use	2	16	2
Allowed for most creativity	3	13	4
Overall Preference	2	16	2
Which platform would you use most frequently?	2	12	6

A combination of user study results and informal feedback reveals issues and ideas for the next phase of development. The recurring issues and ideas are:

- Users have requested more in depth audio manipulation tools to increase the expressivity of the application. It is important to note that this was requested from users from the whole range of prior musical experience.
- It has been suggested that the added menu buttons that replace the instrument buttons are too small. This is evident predominantly on the iPod Touch device, but rethinking their position and size could improve the overall interface for the other hardware contexts simultaneously.
- Multi-touch capability has been requested for the iOS devices. Currently, users are only able to play one note at a time. By implementing multi-touch, users will be able to play more complex harmonic structures with chords and progressions.
- A more comprehensive tutorial has been requested. Some specific usability issues, such as, users difficulty in creating and looping sequences appears to be related to briefness of the existing tutorial. By altering the tutorial to be more comprehensive, users will be able to gain a quicker understanding of the core mechanics of the application.

With the early development and testing that has been shown in this chapter, the following chapter will focus on the next stages of development that will lead toward a publicly available iteration of the application.

Chapter 4

Implementation

This chapter discusses the additional development required before the application can be deployed and further iterations that will subsequently occur as software updates to the publicly available application. Additionally, the final sections will discuss the implementation of a social element to the resulting application and the final user study used to determine the applications success at achieving the goal of this research.

4.1 Fourth Iteration: Pyxis Minor

The overall results from the prior user study indicated that users had a stronger preference for the iOS version of the application. This preference has lead to the decision to develop, optimise and deploy the application primarily favouring iOS devices over the Mac OS X devices.

Multi-touch capability was added to the application to allow users to play chords (multiple notes occurring at the same quantised time) and harmonies (without the necessity of using multiple instrument-sequence pairs). This affords more musical expressivity to the user.

An AUReverb reverberation audio effect, provided by the Apple Core Audio Library, was added to the application. The user is able to adjust the ratio of signal with the reverb applied to the original signal (wet/dry mix) within the global settings menu. This is afforded by a fader located below the groove/swing fader. This reverb effect helps to blend the audio elements from separate instruments together and is a small experiment into how audio effects could be added to the application.

In response to the users' difficulty interacting with the instrument menu (added in the previous iteration) the menu was enlarged and placed beside the instrument selection buttons. A screenshot of this can be seen in figure 4.1

Additionally, some minor aesthetic details have to be decided upon before the application is suitable for public release.

Application Name

The name 'Pyxis Minor' was chosen for the application as it fits within the primary aesthetic theme. 'Pyxis Minor' is a combination of Pyxis, a constellation named after a mariner's compass, symbolically representative of exploration, and Minor, establishing a connection to both small constellations and musical composition techniques. This ties together the concepts of exploration, outer space, and music within the application.

Figure 4.1: Instrument settings menu buttons modified from prior iteration in order to be easier to use.

Application Logo

A splash screen with the logo will be added to the application. The logo consists of the name of the application, with a slightly modified depiction of the Pyxis constellation connecting the diacritics of the i's in 'pyxis' and 'minor'. This depiction features the same colour and shape of a constellation drawn with the red instrument (Ceres) within the app itself. A simplified logo consists of just the depiction of the constellation, to be used as the application launch icon. Both the logo and the launch icon can be seen in figure 4.2.

Figure 4.2: Pyxis Minor launch icon (left) and logo (right).

4.1.1 Application Release

Pyxis Minor was published to the Apple App Store as a universal iOS application on the 28th of October 2014. A screenshot of the first publicly available version can be seen in figure 4.3. The decision to publish Pyxis Minor as a universal app means that users will be able to download the app for all iOS devices. The minimum iOS the application will run on is 6.1. By investigating download reports, it can be determined which devices and iOS versions Pyxis Minor is downloaded to and use feedback gathered from the App Store reviews in order to improve and iterate through updates to the application.

Initial Results of Release

As of the 24th of February 2015 the number of downloads for all platforms exceeds 1500. iPad downloads account for approximately 61% of the total downloads (938 of 1540). Additionally, all of the total downloads come from the music app category of the App store. This indicates that the primary user base of the application is more likely to be musically inclined people, as they are the people more likely to be browsing the music app category of the App store. More extensive documentation of download

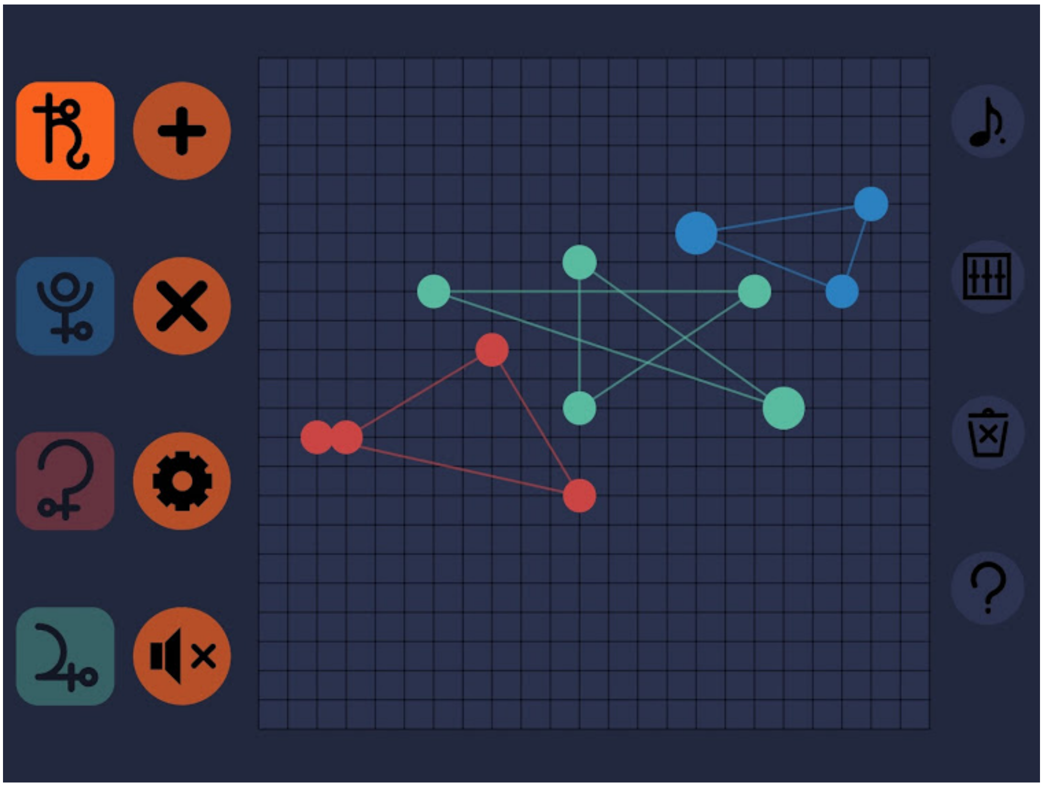


Figure 4.3: Screenshot of the first published version of Pyxis Minor (pictured on iPad).

statistics can be found in Appendix B.

Multiple reviews and comments on various internet forums have suggested that the inclusion of inter-application audio or communication protocols (such as AudioBus, Inter-App Audio or Core MIDI) would increase the user-base, however this functionality would primarily target users that have a prior knowledge of these concepts, which could potentially alienate novice users or diminish their experience. Although this kind of functionality would improve the satisfaction of the user base, preference for improved functionality will be focused on the suggestions from the user study from the prior iteration of development. This is to make sure that the development process does not favour users with prior musical experience, by adding functionality that only appeals to them and not users without prior musical experience.

4.2 Fifth Iteration: Audio Effect Update

Pyxis Minor 1.1 featured, amongst various bug fixes and tweaks, the inclusion of audio effect processing to the audio signal chain. This update was released on the 26th of November, 2014. Originally, the audio effects were going to be added by dynamically changing the AUReverb effect in the signal chain. Attempting to do this without creating significant audio dropouts and creating a uniform way of interfacing with audio effect parameters was problematic. Due to this, in combination with the limited *types* of audio effects provided by Apple audio units, a different way of implementing audio effects was developed. The audio effects were implemented in the audio signal chain by incorporating some of the effects from the Synthesis Tool Kit (STK) [43] to allow each instrument to have a separate audio effect. This is achieved by applying the effect processing to the audio render callback on the AUMixerNode inputs; the mixer node applies the effect on the audio it pulls from each AUSampler before it gets processed by the mixer itself, with each input channel/effect having parameters that can be adjusted by the user. The STKChorus and JCReverb effects are unmodified implementations from STK. The flanger effect is a chorus effect from STK modified to feed the original signal back into the effect input. The tap delay effect, similarly is a STKDelay effect modified to allow the original signal to feedback. The low pass and high pass filters are implementations of algorithms of the LPF and HPF Unit Generators implemented in ChucK [37]. These effects are accessible from an additional menu bar item (symbolised by an audio mixer) on the right side of the main screen, between the global settings menu button and the delete all sequence button. This will reveal an audio effects menu, which can be seen in figure 4.4.

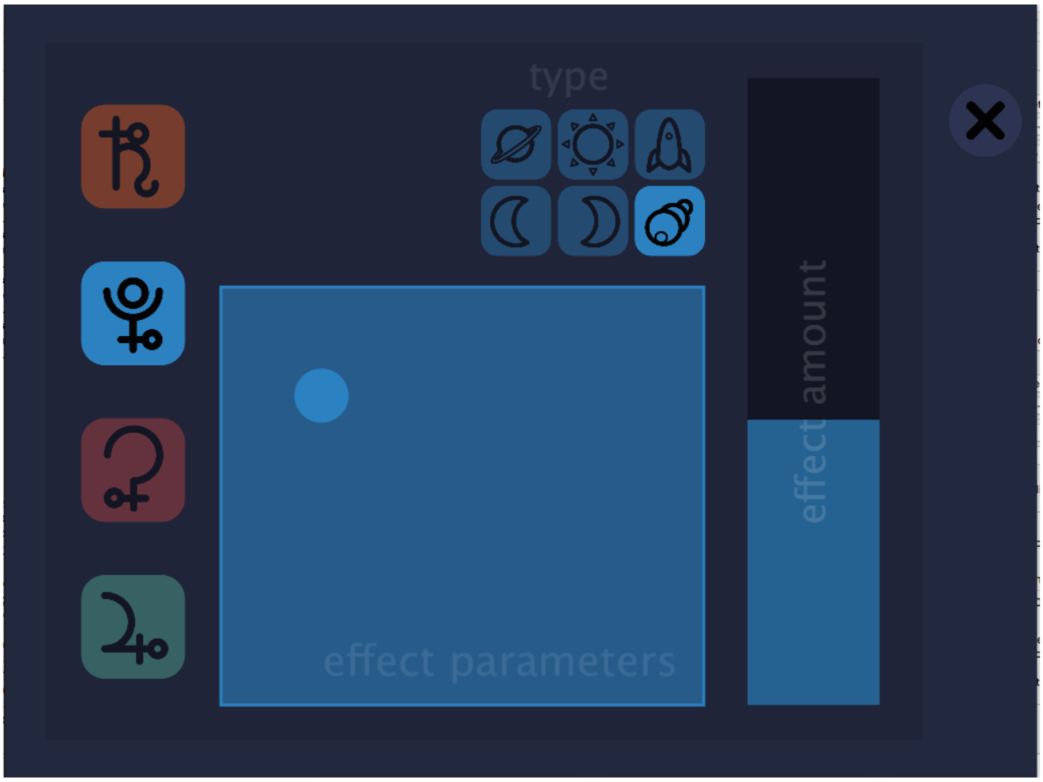


Figure 4.4: Effect menu screen added to Pyxis Minor 1.1.

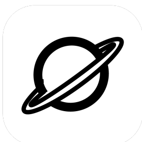
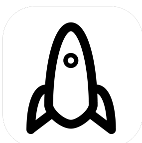
The audio effect menu has multiple UI elements for users to apply effects. The user selects the instrument from a panel that is identical to the instrument selection on the main screen. The top of the effects menu allows the user to select the kind of effect they would like to apply to the instrument. An x-y-pad in the centre of the menu allows users to adjust two parameters of the audio effect simultaneously. These parameter mappings vary between effects. This approach is based upon how the Korg Kaoss Pad hardware[1] and iKaossilator application[2] both allow the user to adjust effect parameters. A table showing the effect types, symbolic representation and the parameters adjusted by the xy-pad can be seen in table 4.1. The symbols were chosen to reinforce the aesthetic thematic link to outer space. Additionally, the effects themselves were chosen based upon stereotypical sonic depictions of outer space (especially the chorus and tap delay effects). An additional fader on the right hand side of the menu adjusts the amount of the chosen effect, mixing between the amount of effected and uneffected audio, with the exception of the flanger effect. In that case, the fader adjusts the attenuation of the signal fed-back into the flanger effect.

Two other significant developments, in addition to the effect menu and processing, were incorporated into the updated Pyxis Minor 1.1 application. Firstly, some users reported issues hearing the audio when the application was started. After investigating, it was discovered that some of the speakers in iPhone devices were unable to reproduce the lower octaves of sine wave notes (which is the default when the application loads). This confused some users into believing that the audio did not work for the application at all. This issue was resolved by making the initial pitch of the application two octaves higher.

The second issue concerned users being unable to discern how their

[1] `http://www.korg.com/us/products/dj/kaoss_pad_kp3_plus/`
[2] `http://www.korg.com/us/products/software/ikaossilator/`

Table 4.1: Visual symbols depicting effect types implemented into the Pyxis Minor 1.1 update.

Effect Type	**Symbol**	**Symbol Description**	**X Parameter**	**Y Parameter**
Low Pass Filter		Waxing crescent	Frequency Cutoff	Filter Resonance
High Pass Filter		Waning crescent	Frequency Cutoff	Filter Resonance
Tap Delay		Aligned planets	Delay time	Delay Feedback
Chorus		Ringed planet	Chorus width	Chorus depth
Reverb		Sun / Star	Decay time	High frequency attenuation
Flanger		Rocket Ship	Flanger width	Flanger depth

interaction was being temporally quantised. To mitigate this issue, an audible metronome was added to the application. The metronome is an instance of the percussion preset of an AUSampler, which feeds into the AUMixerNode. This AUSampler receives notes from the quantisation engine clock. An option was added to the global settings menu to allow users to mute/unmute the metronome. The metronome plays hi-hat notes in order to fit with the sonic aesthetic of the application. A flowchart visually depicting the Audiograph of Pyxis Minor 1.1 can be seen in figure 4.5.

4.3 Sixth Iteration: Tutorial Update

Pyxis Minor version 1.2 was released on 16th of January, 2015. The primary aim of this update is to add a more extensive tutorial option to guide users. Previous feedback indicated that users enjoyed the application more once they were comfortable with the interaction methods. Originally, the concept for the application was to be intuitive enough that users would not need instruction - ideally, the process of figuring out the application (by exploration) would be one of the main activities a user would engage in with the application. In practically, this approach has been difficult for a number of reasons. Firstly, additional functionality and concepts that have been added to the application necessitate more interface complexity, which decreases the capacity for the application to be comprehensively understood intuitively. Secondly, delineating the goal orientation of the application as specifically user-defined is a difficult task, which is only exacerbated by excluding explicit instruction to indicate this concept to the user. During the user studies, on a few occasions, users approached the

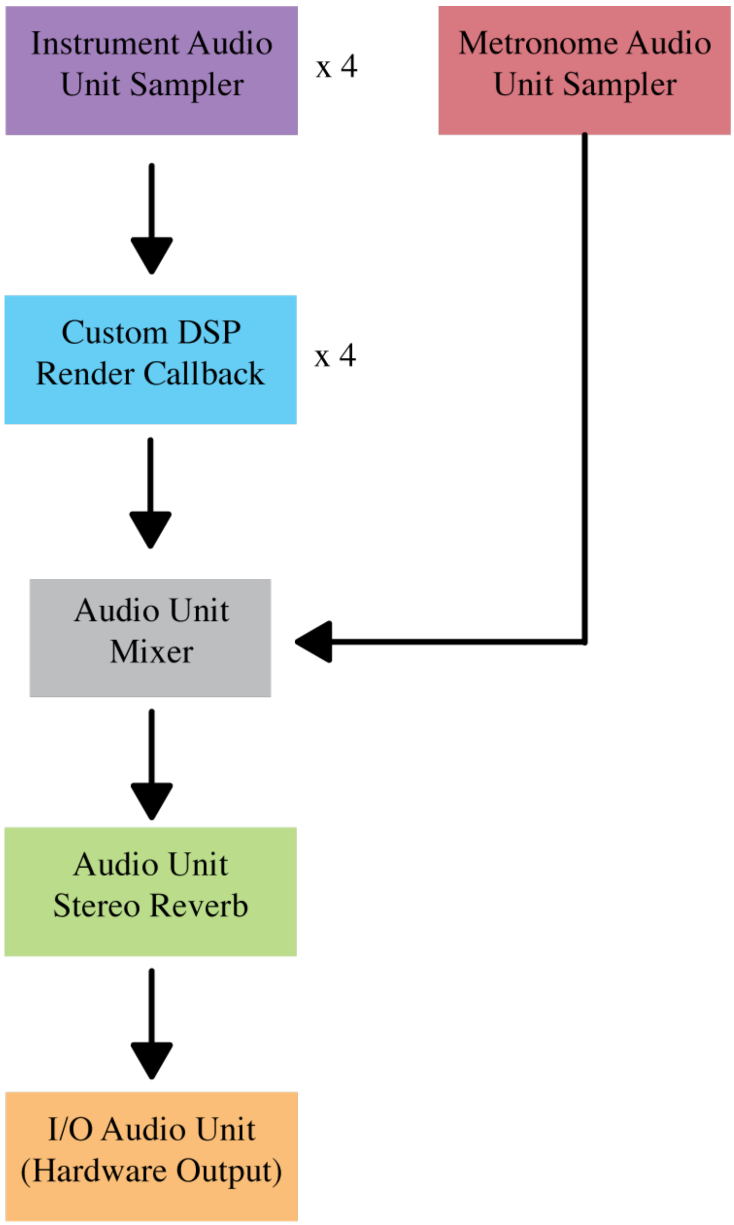


Figure 4.5: Flowchart illustration the Audiograph of Pyxis Minor 1.1.

application by asking the investigator 'What do I do?', which illustrates the common approach to understanding an application as functionally defined. Apps are often understood by the utility they serve the user. By illustrating the core functionality of the application without enforcing strict goal orientation, Pyxis Minor allows users to define their own intention with the application. The previous approach to instructing the users was to provide a tutorial focused upon brevity as to not bore or annoy the user. The result, however, was a tutorial that lacked complete description of interactions, interface, and musical concepts to the user. This issue has been mitigated by developing and designing a navigable tutorial that allows users to understand specific facets of the application at their own pace.

An additional splash menu was added to the application to allow users to navigate to the tutorial menu before playing with the application. Screenshots of this splash page and the main tutorial screen can be seen in figures 4.6 and 4.7, respectively. The tutorial itself is divided into four main aspects of the application, indicated by a corresponding button. Selecting one of these buttons will reveal to the user screenshots of the application that focus on specific elements of the user interface, using text to describe the functionality of the specific elements. A return button will return the user to either the main screen or the splash screen, depending on how they launched the tutorial.

An update to the percussive sample bank for the audio engine has been included in Pyxis minor 1.2. The sample bank is comprised of a collection of percussive elements including traditional and electronic kick drums, cymbals, and snares, as well as non traditional percussive elements such as water drops, clock ticks, and machine noises. This sample bank was crafted with reference to the desired musical aesthetic of the application in order to provide the ability to create more traditional rhythmic structures found in electronic music than what is possible with the original noise-

Figure 4.6: Splash screen added to Pyxis Minor 1.2.

based sample bank.

4.4 Seventh Iteration: Collaboration

The research until this point focussed upon building and developing the original concept into a fully functioning application with intuitive user interfacing mechanics. This is because it is important to get the initial design, mechanics and usability of the application to a suitable level before complicating the interaction with collaborative functionality. The final stages of development address the socially collaborative aspect in order to fulfil the research criterion of *social collaboration*.

In order to add social collaboration, a networked metronome functionality is added to the application. This allows multiple users (with multiple

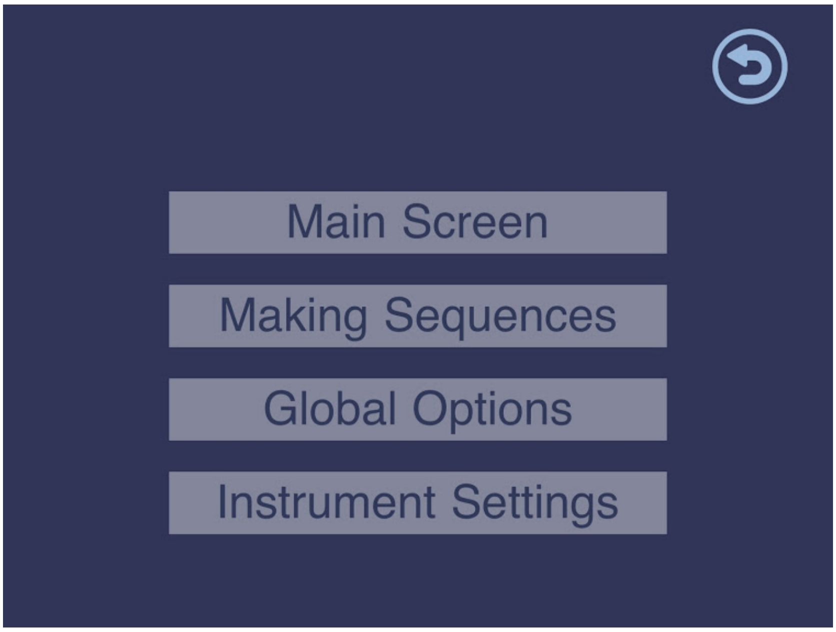


Figure 4.7: Tutorial screen added to Pyxis Minor 1.2.

devices) to connect the metronomes together (and the subsequent quantisation engine). This means that all the user interaction of multiple users will is quantised to the same metric pulse, so that users are able to make music collaboratively. The intention is to add only a minimal networking functionality, with minimal data sent between the devices, in order to place the focus of the collaborative and social interaction on the communication between users (either verbally or non-verbally. This allows the responses they make to one another's musical interactions to be similar to how musicians of traditional acoustic musical instruments would.

Due to the rigidity necessitated by the application, as shown in the `sharedBeatClock`, networking using OSC[38] over a UDP connection resulted in a poor syncing of the metronomes. This method of networking the metronome suffers from significant jitter, despite latency calibration developed to mitigate this issue. In order to proceed with the concept of networked metronome, Korg's Wireless Sync Start Technology[3] (WIST) has been implemented. WIST functions by synchronising an initial control function to start the metronome at the same time on two devices with the same BPM. WIST is primarily implemented in applications targeted towards experienced musicians[4], the implementation in this application poses the question as to whether these types of functionality can be suitably deployed on applications with a target demographic inclusive of novice users. Although only the initial start time of the metronomes are synchronised, the design of the metronome and quantisation functionality has resulted in metronomes that do not get out noticeably out of sync even after periods of approximately 20 minutes. One limitation of the implementation of WIST is that the networked functionality is limited to two devices simultaneously.

The user interface for setting up the wireless technology was designed to be simple and non-exclusive of beginner users. Two buttons were added to the global settings menu. The first of these is a pause/resume button, which pauses and resumes the application playback. The second of these buttons features a networking icon. Selecting this button will open a dialogue, whilst searching for other devices for Pyxis Minor to connect to. A user can select any devices found from a drop down menu to attempt to connect. The other device receives a notification that another device is attempting to connect, in order to either allow or deny the request. If the user accepts the request, they will take on the role of 'master' whilst

[3] `http://www.korguser.net/wist/`

[4] See footnote 3.

the device that sent the request will become the 'slave'. The slave device loses ability to control the BPM slider and the pause/resume button. These changes to the global settings menu can be seen in figure 4.8. In order to test whether this minimal networked technology improves the Pyxis Minor by adding a social element and whether the social element is sufficient for this application, a user study will be conducted.

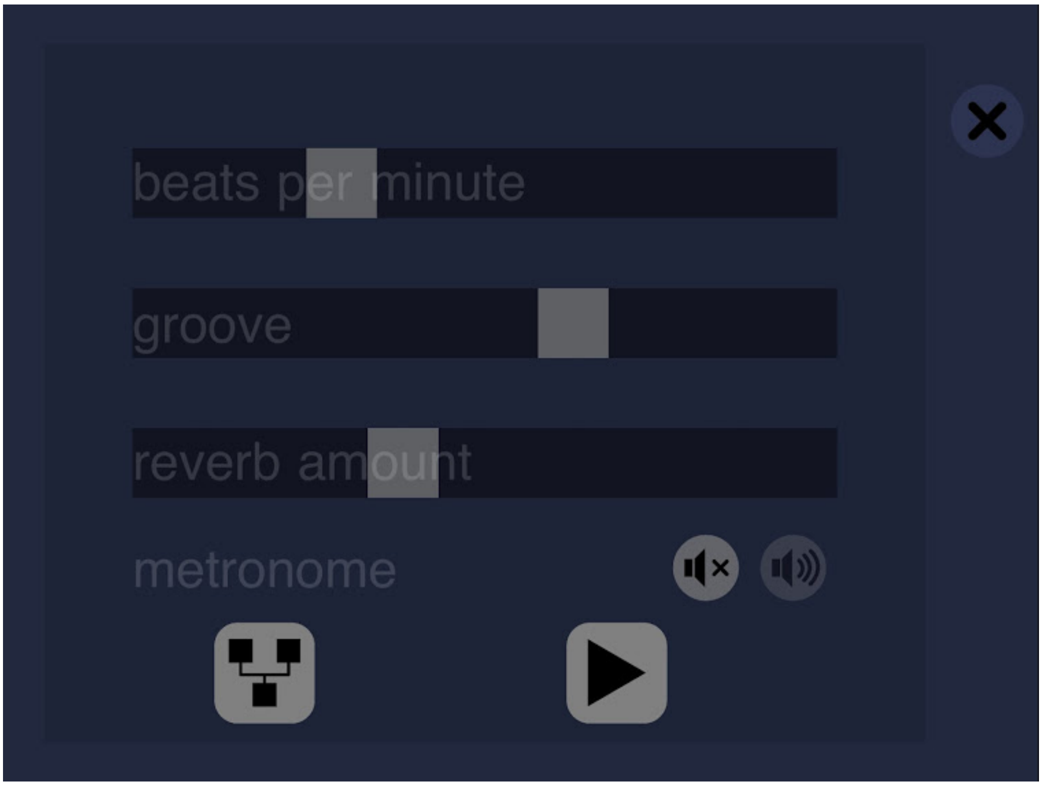


Figure 4.8: Screenshot of the added pause/resume and networking buttons added to the global menu to allow for a networked metronome.

4.5 Social Collaboration User Study

A user study was conducted to test the effect the collaborative element added to Pyxis Minor. Additionally, questions were asked pertaining to the application itself in order to determine the success of Pyxis Minor in achieving the goal set out for this research. Users were provided with an iPad and headphones and asked to familiarise themselves with the application using the tutorial menu for guidance, before engaging free play for approximately 5–10 minutes with the application. Users were then paired together (with an iPad each) and introduced to the networked metronome functionality before being asked to engage in free play for an additional minimum 5–10 minutes, although they were advised they could spend more time if they wanted. Users were asked to complete a questionnaire about their experience with the application in both the solo and collaborative operations and their preference of interactions. The full questionnaire can be found in Appendix A.

4.5.1 User Study Feedback

Selected quotes from the user study indicate that the comprehensive tutorial added to Pyxis Minor helped users to understand the core mechanics of the application quicker, with one user remarking that '[The] help guide was very valuable! Otherwise I would've been a bit confused.' Of the elements not explained in the tutorial, one user stated 'Some of the buttons aren't self explanatory, but it only takes a moment to work out what they mean', which indicates that there is still an element of exploration required of the application, but it does not inhibit the overall usability of the application, as did the prior iterations without the comprehensive tutorial.

User preference for the socially collaborative aspects far outweighed that of solo interaction. One user stated that 'It's very fun and much more conducive to creativity than I expected. The networking is surprisingly smooth.' One pair of participants requested to continue playing on the application after completing the user study. They continued to collaborate for over half an hour beyond what was requested for the study.

4.6 Summary

This chapter has discussed the development involved in transitioning the application discussed in chapter 3 into Pyxis Minor - a publicly available musical app for electronic music creation. Since being released, Pyxis Minor has been downloaded 1540 times. Through continuing iterative design after public deployment based upon feedback from various forums outside of specific user study situations (such as internet forums, app store reviews, and music app blogs) the development of Pyxis Minor has improved the user satisfaction, making significant steps towards achieving the goal of this research.

Chapter 5

Results & Evaluation

This chapter will provide an evaluation as to the success of Pyxis Minor in achieving the goal of this research. In order to do this, each element of the criteria established in section 1.2 will be evaluated using data from the user studies discussed in chapters 2 and 3. This chapter will make reference to specific questions asked in the questionnaires provided to users taking part in the user study. These questionnaires can be found in Appendix A.

5.1 Evaluation

The final user study, in addition to prior user studies conducted allowed for the gathering of data, upon which the analysis will be performed to ascertain the effectiveness of the developed application. The following section is broken into three parts covering the specific criteria established in section 1.2.

5.1.1 User Creativity

All three of the user studies asked whether the users agreed they experienced creativity by using the tested iterations of the application. A 5 point Likert scale was used, where the users could indicate their level of agreement to the statement 'Using this application made me feel creative'. The scale was tabulated using the numbers from 0–4, with 0 indicating the absence of feeling of creativity and 4 indicating the user felt creative. The null hypothesis is that there is user answers to this question would result in a uniform distribution. A value less than or equal to 2.0 would indicate users did not feel sufficiently creative using of the application. If it can be demonstrated that the mean of sampled user preference is greater than 3.0 (indicating generally favourable creative experience), and there is a small standard deviation, (implying the majority of users would feel this way) we will consider the application successful at achieving the creative element of the research goal (as specified in research criteria 1).

Analysis of the given answers to this question in the final user study, results in a chi squared value of 24.77. The expected value indicating that this is not a uniform distribution, for a confidence level of 99% is 25.5 for the sample size of 12. This indicates the results of this question are statistically significant. The average value for the final iteration and corresponding user study is 3.75 with a standard deviation of 0.42. The average values from this user study are above the threshold for which we have defined the success of this application at allowing the user to experience creativity by using the developed application. Additionally, even the mean minus the standard deviation is above our success threshold of 3.0. This indicates that we can be reasonably sure that most users will feel creative by using this application.

This result can be used to make comparison to previous iterations. Notably, over the course of the three user studies no users selected values for 0 or 1, indicating further that the application allows for user creativity, with the minimum user result from all three user tests indicating at least a moderate feeling of creativity was experienced. The mean for the pilot study is 3.2 with a standard deviation of 0.83. The mean for the first full scale user study is 3.4 with a standard deviation of 0.50. Due to the limited size of the pilot study, statistical significance cannot be ascertained, however a single tailed T-test comparing the results of the two full-scale user studies shows that the increasing mean is statistically significant with a confidence level of 95%. This shows that the successive iterations have increased the consensus that users felt creative whilst using the application. Simultaneously, the decreasing standard deviation indicates that more users are likely to agree. A graph of this trend can be seen in figure 5.1. Overall, this gives strong indication that using an iterative design process informed by user feedback has led to an application that allows more users to feel creativity, justifying the use of this design process itself.

5.1.2 Social Collaboration

To ascertain the success of the application in regards to the socially collaborative research criterion (criterion 2 specified in section 1.2), it must be shown that the collaborative functionality of the application is indeed social. The final user study asked users whether they agreed that 'The

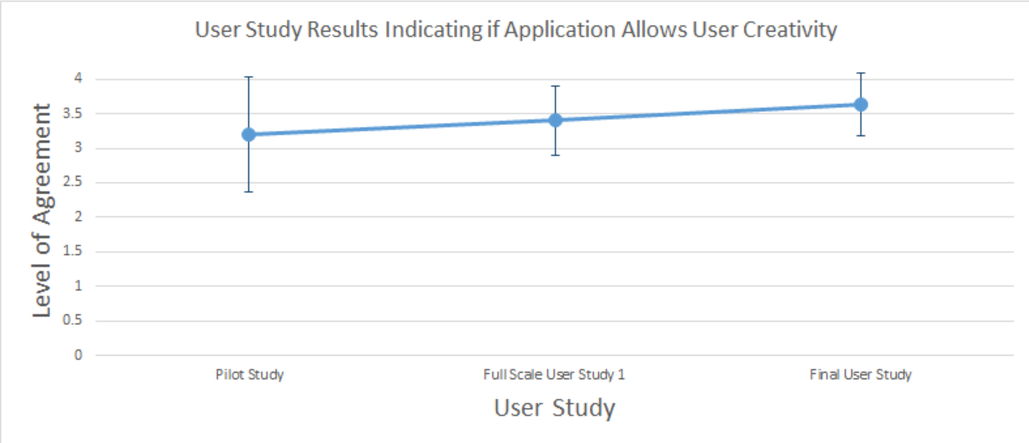


Figure 5.1: Graph showing positive trend of increasing consensus that the application allows for creativity over development iterations and successive user tests.

collaborative aspect of the application feels like a social experience.' The null hypothesis is that the results will fit a uniform distribution (with an approximate mean of 2.0). The tabulated values of this question (using the same Likert scale and process as for user creativity), resulted in a chi squared value of 14.5. The expected value with a 95% confidence level for a test with four degrees of freedom is 9.49. This means that the results of this tabulation are statistically significant. The mean is 3.27 with a standard deviation of 0.46. This shows that the the users reasonably believe that their collaborative interaction in the user study was a social experience.

Additionally, users were asked their level of agreement to the statement that '[When using the collaborative aspect of the application] it feels as though we are making music together.' Analysis of the results from this question gives a chi square value of 11. This is, again, above the expected value for a confidence value of 95% for a test with four degrees of freedom, indicating statistically significant results. The mean for this question is 3.17 with a standard deviation of 0.717. Although the mean is less than the previous question, with a larger standard deviation, this still indicates positive results for the application. The results suggest that users agreed as though their collaborative interaction felt like they were playing music *with* their paired participant.

The final user study questioned users as to whether they preferred their solo interaction or the collaborative interaction for a number of criteria. The question 'Which interaction do you think sounded the best?' has users unanimously agree that the collaborative interaction sounded better than the solo interaction, with all 12 surveyed users preferring collaborative interaction. For the question 'Which interaction do you think resulted in the most fun?', 10 of the 12 surveyed users indicated preference for the collaborative interaction, with the remaining 2 indicating no preference. Similarly, for the question 'Overall, which interaction did you prefer?', 10 of the 12 surveyed users indicated preference for the collaborative interaction, with the remaining 2 indicating no preference. Of the sampled users, no user indicated a preference for solo interaction in any of the three categories. This, in combination with the previous analysis revealing that the collaborative interaction is both social and results in an experience of playing music *together*, demonstrates that the evaluative criterion of providing a social aspect in the application has been achieved.

Additionally, participants were also questioned about what amount of users, do they think, the application would most benefit from allowing to interact simultaneously. Participants were able to select from 1–5 (or above) users for each question regarding this field of inquiry. In each case, the null hypothesis is that the results would indicate a uniform distribution, so the expected value would be 3. The results of the question 'What number of simultaneous users do you think would result in the most fun?' gives a chi-square value of 33.2 which is outside the expected value of 13.27 for a confidence of 99%, indicating statistically significant results. The mean for this question is 2.08, with a mode of 2. The chi-square value of the question 'What count of simultaneous users do you think would result in the best sounding music?' is 25.2 - which is also statistically significant with a confidence of 99%. The mean for this question is 2.50 and the mode is 2. The results of the third question 'What amount of simultaneous users do you think would result in the most creative experience?' gives a chi square value of 13.2, which means this result is statistically significant, above the expected chi-square value of 11.14 for a confidence level of 97.5%. The mean of these results is 2.75 with a mode of 2.5. No users indicated that solo interaction is preferable for any of the categories. Overall, the results of this show that users were *mostly* satisfied with the application only allowing a maximum of two people to collaborate. Considering that of the three questions, the medians and modes were all between 2–3, the potential benefits of adding functionality to allow more than two users to collaborate simultaneously does not outweigh the necessary time investment at this stage of development.

5.1.3 Usability

The final criterion for evaluating the success of the application outline in section 1.2 is the usability of the resultant application (criterion 3). In the final user study, a 5-point Likert scale was used to determine users' level of agreement to the statement 'The application is easy to use.' In this case, the null hypothesis is that results would indicate a uniform distribution, where a mean above 3.0 would indicate that the application is easy to use. Tabulating the results of user answers to this question gives a chi square value of 14.5. This is above the expected value of 9.5 for a confidence level of 95%. This indicates that the results are statistically significant. The mean of this question is 3.27 with a standard deviation of 0.46. This value is above our target of 3.0 for the level of usability, indicating that the resultant application has achieved the criterion of usability for determining the success at achieving the research goal. However, this mean minus the standard deviation is slightly below the target. This means that additional developments should be made to improve the overall usability for some users of the application. This will be discussed in the future work section of this research (section 6.2).

The decision to not compare usability over successive iterations has been made because the iterative process will successively add functionality to the application to allow for more creativity. This upward functionality will necessitate modifying the user interface which has implications for comparison of usability different iterations of the application. However, this indicates that the user informed design process is integral to achieving both the evaluative criteria for *usability* and *creativity* of the application.

Chapter 6

Conclusion

To conclude this thesis, this chapter will summarise the research conducted within the thesis and the results that have been obtained. Additional work to Pyxis Minor and future avenues of research resulting from this thesis will be discussed.

6.1 Summary

The first chapter of thesis has presented a framework for the development of a new electronic musical instrument in order to lower the barrier to entry for electronic music (in section 1.1) and presented evaluative criteria for which we can determine the success of a resulting application (in section 1.2). By drawing upon previous works in the fields related to this research, in chapter two, the framework has been justified. Furthermore, the principles of the framework established as appropriate criteria for steering the development of an application. The specific criteria, as defined in section 1.1, are driven primarily by *development informed from users*, to result in an application that focuses upon *active interaction*, is *easily accessible*, *socially collaborative*, *technically transparent* and *artistically & aesthetically valid*.

The third and fourth chapters of this thesis documented the implementation of the framework. Relying on user feedback through a combination of a pilot study and two full scale user studies the development went through multiple iterations, resulting in Pyxis Minor, an application for iOS devices. These chapters presented the movement from an original application concept to functional prototype in chapter three. Subsequently becoming a fully developed, publicly available iOS application in chapter four. The final full scale user study was then conducted. This user study helped to determine future development necessary for the continued success of Pyxis Minor, while also ascertaining the success of the application (and by extension, the framework implemented) at achieving the goal of this research.

Chapter five applied the evaluative criteria to the results generated from the user studies conducted over the previous two chapters in order to reveal the effectiveness of the resulting application, the development process and the framework established in the section 1.1. Despite the user studies revealing specific future work that can benefit the resulting application, the overarching result of the evaluation is that Pyxis Minor has achieved the goal established for this research in section 1.2.

6.2 Future Work

This following section is split into two specific facets of future work. The first will focus upon the specific features and improvements that will be made to the application resulting from this research. The second will discuss future avenues that could be explored in relation to the results of this research, or tangential fields of inquiry revealed by this research.

6.2.1 Pyxis Minor Future Work

Feedback from the user studies and personal aesthetic taste has revealed future improvements to be made to Pyxis Minor:

- Additional sampled instruments will be added to the application. User feedback indicated that the addition of the percussion samples (detailed in section 4.3) dramatically improved the resulting sound of Pyxis Minor. By adding additional sampled instruments, an additional layer of timbral complexity will be afforded, tailored to the desired aesthetic, that will allow for more creative flexibility than the existing simple waveform-based instruments.
- Additional effects will be added to the instrument, with more descriptive names to indicate the type of effect to the user. In the final user study, one user indicated that they would like to know what the specific effects were. By adding the effect names, beginner users will be able to gain an understanding of what each effect sounds like, whilst more experienced users will be able to materialise sonic intention faster.
- The conformal mapping technique and effect based accelerometer control discussed in sections 3.2.1 and 3.3 will be added to the application. This will allow users to control the xy-pad of the effects

menu using accelerometer data. This will afford an additional layer of creative expression to the users and increase the level of physical performativity of the application.

- An iOS 8.0 specific bug is preventing the public release of the socially collaborative aspect of the application developed in section 4.5. The issue prevents users from seeing the full list of available devices that they may connect to. Once this issue is resolved, the publicly available Pyxis Minor application will be updated to allow users collaborative functionality.
- Some users have expressed a difficulty with understanding how their interaction is temporally quantised, which makes it difficult to make loops with their intended number of pulse counts and notes at their intended beat times. This issue can be resolved by adding an additional menu that allows the user to visualise and manipulate their sequences in the time domain, rather than in the pitch/volume domain offered by the main menu in Pyxis Minor. This will appear and behave similarly to how a traditional step sequencer would.

6.2.2 Related Future Work

Additionally, areas of interest related to this research may be explored in future work. Specifically, the adoption of a game engine (Cocos-2d), with the combination of a capable real-time audio engine has provided a unique platform for the resulting application. Beyond the scope of this research is an analysis of the semantic baggage that comes with the adoption of a video game engine. Future research could seek to answer questions related more closely to the intersection of video games and music technology, for example, could this software architecture design benefit from exploring other game engine designs (such as 3D engines)? Could the adoption of other video game concepts (not just the technical structures) allow for new forms of hybrid interactive media, between electronic musical instruments and video game entertainment?

Exploration of the pedagogical function of Pyxis Minor has only briefly been discussed within the scope of this research. Beyond this scope, questions naturally arise as to the degree that this work (and related work) functions as pedagogical tools. By basing the focus of future work on this pedagogical function, research can be conducted into the creation and analysis of novel and enjoyable tools for teaching more theoretical musical concepts and techniques, not necessarily specific to electronic music.

The development and implementation of the conformal mapping techniques expressed in this research could be suitably adopted to other applications. Specifically, mapping accelerometer movement could be beneficial in aiding physical therapy and rehabilitation by simultaneously monitoring user input, whilst providing feedback to the user (afforded by multimedia on the device) through musical or visual stimuli. Existing applications, such as PhysioAdvisor[1], provide rehabilitation exercise demonstrations to users, however, by monitoring user input and utilising interaction methods based upon more advanced conformal mappings of accelerometer (and gyrometer) inputs, this field of research could be further explored.

As can be seen, this research is not only built upon a foundation of multiple fields and related works, but may provide basis for future works

[1]https://itunes.apple.com/app/physioadvisor-exercises/id397276649?mt=8

in these related fields.

6.3 Final Remarks

The results of the user studies conducted over the iterative process involved in creating and developing Pyxis Minor indicates that this research has achieved the stated goal. Pyxis Minor succeeds at achieving the criteria outlined in section 1.2. This means that Pyxis Minor is, indeed, an electronic musical instrument that lowers the barrier to entry for electronic music, inspires musical creativity in potential and existing electronic musicians and incorporates a social element that will allow users to collaborate together. Furthermore, in succeeding to achieve this goal, the research demonstrates that the framework set out in section 1.1 was beneficial to the process and can suitably be utilised in the construction of these types of instruments.

Bibliography

[1] M. Schlick, "On the Meaning of Life," in *Life and Meaning* (O. Hanfling, ed.), pp. 60 – 73, New York: Basil Blackwell, 1988.

[2] E. M. v. Hornbostel and C. Sachs, "Classification of Musical Instruments: Translated from the Original German by Anthony Baines and Klaus P. Wachsmann," *The Galpin Society Journal*, vol. 14, pp. 3–29, Mar. 1961.

[3] C. Sachs, *The History of Musical Instruments*. Dover Publications, Inc., 2006.

[4] Korg Inc., "MS-20 mini MONOPHONIC SYNTHESIZER | Synthesizers / Keyboards | KORG." [Online] Available: http://www.korg.com/us/products/synthesizers/ms_20mini/.

[5] K. Pratap, "Android Grabs 81 Percent Market Share With Over 1 Billion Phones Shipped in 2014: Report." [Online] Available: http://gadgets.ndtv.com/mobiles/news/android-grabs-81-percent-market-share-with-over-1-billion-phones-shipped-in-2014-report-655583.

[6] C. Poepel, “On interface expressivity: a player-based study,” in *Proceedings of the 2005 conference on New interfaces for musical expression*, pp. 228–231, National University of Singapore, 2005.

[7] J. Nielsen, “Iterative user-interface design,” *IEEE Computer*, vol. 26, pp. 32–41, Nov. 1993.

[8] O. Sacks, *Musicophilia*. Vintage Books, 2008.

[9] T. Blaine and S. Fels, “Contexts of Collaborative Musical Experiences,” in *Proceedings of the 2003 conference on New interfaces for musical expression*, pp. 129–134, National University of Singapore, 2003.

[10] G. Weinberg, R. Aimi, and K. Jennings, “The Beatbug Network: A Rhythmic System for Interdependent Group Collaboration,” in *Proceedings of the 2002 Conference on New Interfaces for Musical Expression*, NIME ’02, (Singapore, Singapore), pp. 1–6, National University of Singapore, 2002.

[11] D. Trueman, P. Cook, S. Smallwood, and G. Wang, “PLOrk: the Princeton laptop orchestra, year 1,” in *Proceedings of the international computer music conference*, pp. 443–450, 2006.

[12] G. Weinberg and S. Driscoll, “iltur: connecting novices and experts through collaborative improvisation,” in *Proceedings of the 2005 conference on New interfaces for musical expression*, pp. 17–22, National University of Singapore, 2005.

[13] J. A. Paradiso, K.-y. Hsiao, and A. Benbasat, "Tangible Music Interfaces Using Passive Magnetic Tags," in *Proceedings of the 2001 Conference on New Interfaces for Musical Expression*, NIME '01, (Singapore, Singapore), pp. 1–4, National University of Singapore, 2001.

[14] L. Dahl, J. Herrera, and C. Wilkerson, "Tweetdreams: Making music with the audience and the world using real-time twitter data," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, pp. 272–275, Citeseer, 2011.

[15] K. Tahiroglu, "Dynamic Social Interaction in a Collective Mobile Music Performance.," *Proceedings 12th IEEE International Conference on Computational Science and Engineering*, pp. 854–859, 2009.

[16] R. Rowe, "The aesthetics of interactive music systems," *Contemporary music review*, vol. 18, no. 3, pp. 83–87, 1999.

[17] S. Gresham-Lancaster, "The Aesthetics and History of the Hub: The Effects of Changing Technology on Network Computer Music," *Leonardo Music Journal*, vol. 8, pp. 39–44, 1998.

[18] G. Weinberg, "The aesthetics, history, and future challenges of interconnected music networks," in *Proceedings of the 2002 International Computer Music Conference*, International Computer Music Association, 2002.

[19] A. Cart, P. Rebelo, and A. Renaud, "Networked music performance: State of the art," in *Audio Engineering Society Conference: 30th*

International Conference: Intelligent Audio Environments, Audio Engineering Society, 2007.

[20] A. Kapur, M. Darling, D. Diakopoulos, J. W. Murphy, J. Hochenbaum, O. Vallis, and C. Bahn, "The Machine Orchestra: An Ensemble of Human Laptop Performers and Robotic Musical Instruments," *Computer Music Journal*, vol. 35, pp. 49–63, Dec. 2011.

[21] Korg Inc., "KORG WIST | Wireless Sync-Start Technology." [Online] Available: http://www.korguser.net/wist/, 2014.

[22] G. Wang, "Designing smules iphone ocarina," in *Proceedings of the International Conference on New Interfaces for Musical Expression. Pittsburgh*, vol. 291, 2009.

[23] G. Wang, J. Oh, and T. Lieber, "Designing for the ipad: Magic fiddle," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, vol. 30, 2011.

[24] G. Essl, "Speeddial: Rapid and on-the-fly mapping of mobile phone instruments," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, pp. 270–273, 2009.

[25] S. Trump and J. Bullock, "Orphion: A Gestural Multi-Touch Instrument for the iPad," *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2014.

[26] M. Kaltenbranner, S. Jorda, G. Geiger, and M. Alonso, "The reacTable*: A Collaborative Musical Instrument," in *15th IEEE*

International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006., pp. 406–411, June 2006.

[27] G. Largillier, "Developing the first commercial product that uses multi-touch technology," *Information Display*, vol. 23, no. 12, p. 14, 2007.

[28] Reactable Systems SL, "Reactable Mobile." Retrieved from: https://itunes.apple.com/en/app/reactable-mobile/id381127666?mt=8, Apr. 2011.

[29] Liine, "Lemur." Retrieved from: https://itunes.apple.com/us/app/lemur/id481290621?mt=8, Dec. 2011.

[30] T. Kell and M. M. Wanderley, "A Quantitative Review of Mappings in Musical iOS Applications," in *Proceedings of the Sound and Music Computing Conference 2013*, pp. 473–480, 2013.

[31] Korg Inc., "iMS-20." [Online] Available: https://itunes.apple.com/nz/app/korg-ims-20/id401142966?mt=8 , Nov. 2010.

[32] Lucky Frame Inc., "Pugs Luv Beats." Retrieved from: https://itunes.apple.com/us/app/pugs-luv-beats/id488112663?mt=8, Dec. 2011.

[33] Lucky Frame Inc., “Bad Hotel.” Retrieved from: https://itunes.apple.com/nz/app/bad-hotel/id548665066?mt=8, Aug. 2012.

[34] Lucky Frame Inc., “WaveTrip.” Retrieved from: https://itunes.apple.com/nz/app/wave-trip/id581914455?mt=8, Dec. 2012.

[35] C. Stewart, M. Rohs, S. Kratz, and G. Essl, “Characteristics of Pressure-Based Input for Mobile Devices,” in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 10–15, 2010.

[36] A. Tanaka and others, “Mapping out instruments, affordances, and mobiles,” in *Proceedings of the 2010 Conference on New Interfaces for Musical Expression*, NIME, 2010.

[37] G. Wang, P. R. Cook, and others, “ChucK: A concurrent, on-the-fly audio programming language,” in *Proceedings of the International Computer Music Conference*, pp. 219–226, Singapore: International Computer Music Association (ICMA), 2003.

[38] M. Wright, A. Freed, and A. Momeni, “Open Sound Control: State of the Art 2003,” 2003.

[39] P. Nowell, “Mapping a Square to a Circle.” [Online] Available: http://mathproofs.blogspot.co.nz/2005/07/mapping-square-to-circle.html,
2011.

[40] P. Brinkmann, P. Kirn, R. Lawler, C. McCormick, M. Roth, and H.-C. Steiner, "Embedding pure data with libpd," in *Proceedings of the Pure Data Convention*, 2011.

[41] G. Scarth and R. Linn, "Roger Linn On Swing, Groove & The Magic Of The MPC's Timing." [Online] Available: http://www.attackmagazine.com/features/roger-linn-swing-groove-magic-mpc-timing/.

[42] M. Puckette, *The Theory and Technique of Electronic Music.* World Scientific Publishing, 2007.

[43] P. R. Cook and G. P. Scavone, "The Synthesis ToolKit (STK)," in *Proceedings of the International Computer Music Conference*, pp. 64–166, 1999.

Appendix A

User Study Documents

A.1 Project Ethics Approval

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

MEMORANDUM

Phone 0-4-463 5676
Fax 0-4-463 5209
Email Allison.kirkman@vuw.ac.nz

TO	Timothy Barraclough
COPY TO	Ted Apel Ajay Kapur
FROM	Dr Allison Kirkman, Convener, Human Ethics Committee

DATE	2 July 2014
PAGES	1

SUBJECT	**Ethics Approval: 21033** Analysis and Creation of Group Based Digital Music Performance

Thank you for your application for ethical approval, which has now been considered by the Standing Committee of the Human Ethics Committee.

Your application has been approved from the above date and this approval continues until 3 March 2015. If your data collection is not completed by this date you should apply to the Human Ethics Committee for an extension to this approval.

Best wishes with the research.

Allison Kirkman
Human Ethics Committee

A.2 Pilot Study

A.2.1 Information Sheet for Pilot Study

VICTORIA UNIVERSITY OF WELLINGTON

Interface Interaction User Study Information Sheet

Introduction

The objective of this user study is to gather impressions of a new interface for musical expression. The interface is specifically designed to be of interest to people of varying prior musicianship training and offers varying levels of control over musical parameters. The primary goal of this study is to understand the extent of user's engagement with the interface in order to refine and improve future iterations of the interface design.

This user study has been approved by Victoria University's Human Ethics Committee.

Participation

We are inviting participants to explore an interactive musical and visual installation by using two specific methods of interaction on multiple devices. We will then provide a short questionnaire with questions pertaining to the usability of the interface at providing a satisfactory level of control and enjoyable experience with the installation and which method of interacting with the interface was preferred.

Should any participants feel the need to withdraw from the project, they may do so without question at any time before the data is analyzed. If a participant wishes to withdraw they may do so before August 1st, and should contact the primary investigator via the email address provided below.

Responses collected will be evaluated and may be quoted anonymously in a written report. It will not be required to be identified personally. All material collected will be kept confidential. No other persons, besides the investigation team, will have access to the questionnaire data.

The interaction and questionnaire survey will take approximately 10-15 minutes to complete.

The Installation Context

Participants will be asked to evaluate three different control schemes for a new musical instrument / application. In the application, users are able to make and loop sequences of notes to create rhythms and melodies. The first control scheme participants will utilises a computer keyboard and trackpad to create the sequences and loops. The second control scheme uses an iPod Touch device with the accelerometer and various touch gestures to interact. The third control scheme uses an iPad with various touch gestures for interaction. The investigators are interested predominantly in user opinions of which scheme and device is most enjoyable, intuitive and relevant to the specific installation context and what changes would benefit the installation as a whole.

Results

Results from the study may be published in academic journals, conferences, and technical reports. The results will also be published in the primary investigator's thesis at the end of his Masters. Any participants who are interested in seeing the overall results of the study can provide their email address on the consent form, and information will be sent to them.

We would like to sincerely thank you for your participation in this user study.

Researchers

Principal Investigator
Timothy Barraclough
Masters Candidate
School of Engineering and Computer Science
Victoria University of Wellington
Office: CO239, Cotton Building
Victoria University of Wellington
Email: barractimo@myvuw.ac.nz

Investigator
Ajay Kapur
Supervisor
School of Engineering and Computer Science
Victoria University of Wellington
ajay.kapur@ecs.vuw.ac.nz

Investigator
Ted Apel
Supervisor
New Zealand School of Music
Victoria University of Wellington
ted.apel@nzsm.ac.nz

Investigator
Dale Carnegie
Supervisor
School of Engineering and Computer Science
Victoria University of Wellington
dale.carnegie@ecs.vuw.ac.nz

A.2.2 Questionnaire for Pilot Study

,

VICTORIA UNIVERSITY OF WELLINGTON

USER INTERFACE QUESTIONNAIRE

This questionnaire is designed to survey user preference for a custom installation and two methods of interaction on multiple devices. Please fulfil this questionnaire after interacting with the installation using both the computer keyboard and mouse and the supplied iOS devices. Feel free to ask any further questions or request clarification of the questions asked. Your answers may be used as part of a publication regarding the effectiveness of this installation and the presented methods of interaction.

Please check one box per statement to indicate your level of agreement.

Computer Interaction

After using the keyboard and trackpad to interact with the installation please answer the following questions:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The method of interacting with the installation using the keyboard and trackpad was simple.	☐	☐	☐	☐	☐
It was made obvious how to interact with the installation (using the keyboard and trackpad).	☐	☐	☐	☐	☐
It was simple to access the full range of interaction available.	☐	☐	☐	☐	☐
It was obvious how my keyboard and mouse interactions and gestures were represented **sonically**.	☐	☐	☐	☐	☐
It was obvious how my keyboard and mouse interactions and gestures were represented **visually**.	☐	☐	☐	☐	☐

If there are any specific interaction or gestures you thought should have been available, please list them here:

If there were any interactions or gestures that were difficult to accomplish, please list them here:

iPad Interaction

After using the supplied iPad to interact with the installation please answer the following questions:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The method of interacting with the installation using the iPad was simple.	☐	☐	☐	☐	☐
It was made obvious how to interact with the installation (using the iPad).	☐	☐	☐	☐	☐
It was simple to access the full range of interaction available.	☐	☐	☐	☐	☐
It was obvious how my interactions with the iPad were represented **sonically**.	☐	☐	☐	☐	☐
It was obvious how my interactions with the iPad were represented **visually**.	☐	☐	☐	☐	☐

If there are any specific interaction or gestures you thought should have been available, please list them here:

If there were any interactions or gestures that were difficult to accomplish, please list them here:

iPod Touch Interaction

After using the supplied iPod to interact with the installation please answer the following questions:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The method of interacting with the installation using the iPod Touch was simple.	☐	☐	☐	☐	☐
It was made obvious how to interact with the installation (using the iPod Touch).	☐	☐	☐	☐	☐
It was simple to access the full range of interaction available.	☐	☐	☐	☐	☐
It was obvious how my interactions with the iPod Touch were represented **sonically**.	☐	☐	☐	☐	☐
It was obvious how my interactions with the iPod Touch device were represented **visually**.	☐	☐	☐	☐	☐

If there are any specific interaction or gestures you thought should have been available, please list them here:

If there were any interactions or gestures that were difficult to accomplish, please list them here:

Can you identify any key differences that affected your interaction between the two iOS Devices (iPod Touch & iPad) :

Interaction Method Comparison

The following questions look to compare the methods of interaction with the installation:

	iPod Touch	iPad	Keyboard Trackpad	None of them
Which method of interaction was the easiest to use?	☐	☐	☐	☐
Which method of interaction was the most fun?	☐	☐	☐	☐
Which method of interacting allowed you to be more creative?	☐	☐	☐	☐
Overall, which method of interacting did you prefer?	☐	☐	☐	☐

User Satisfaction

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I was satisfied in the amount of control over the **sonic** elements of the system	☐	☐	☐	☐	☐
I enjoyed the sonic result of the system	☐	☐	☐	☐	☐
I was satisfied in the amount of control over the **visual** elements of the system	☐	☐	☐	☐	☐
I enjoyed the visual result of the system	☐	☐	☐	☐	☐
The combination of audio, visual and interactive elements was coherent	☐	☐	☐	☐	☐
I felt as though I was having a creative experience	☐	☐	☐	☐	☐

If the answer to the last question was not Agree or Strongly agree, please skip to the Additional Questions section.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I felt in control of my creative experience	☐	☐	☐	☐	☐

Which method of interacting, in your opinion, allowed more creative control? (please circle)

iPad Device   iPod Device   Keyboard & Trackpad   Neither

Additional Questions

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I have had prior musicianship training with traditional acoustic instruments	☐	☐	☐	☐	☐
I have had prior musicianship training in electronic music	☐	☐	☐	☐	☐
I have seen or used systems similar to this before	☐	☐	☐	☐	☐

If you have used similar systems to this, please list them here (if you remember their names):

Do you have any suggestions for improving the installation or interaction design?

Please use the space below for any further comments or remarks on your experience with the interface.

A.3 Full-Scale User Study

A.3.1 Information Sheet for Full Scale User Study

VICTORIA UNIVERSITY OF WELLINGTON

Interface Interaction User Study Information Sheet

Introduction

The objective of this user study is to gather impressions of a new interface for musical expression. The interface is specifically designed to be of interest to people of varying prior musicianship training and offers varying levels of control over musical parameters. The primary goal of this study is to understand the extent of user's engagement with the interface in order to refine and improve future iterations of the interface design.

This user study has been approved by Victoria University's Human Ethics Committee.

Participation

We are inviting participants to explore an interactive musical and visual installation by using three specific methods of interaction on multiple devices. We will then provide a short questionnaire with questions pertaining to the usability of the interface at providing a satisfactory level of control and enjoyable experience with the installation and which method of interacting with the interface was preferred.

Should any participants feel the need to withdraw from the project, they may do so without question at any time before the data is analyzed. If a participant wishes to withdraw they may do so before August 1st, and should contact the primary investigator via the email address provided below.

Responses collected will be evaluated and may be quoted anonymously in a written report. It will not be required to be identified personally. All material collected will be kept confidential. No other persons, besides the investigation team, will have access to the questionnaire data.

The interaction and questionnaire survey will take approximately 10-15 minutes to complete.

The Installation Context

Participants will be asked to evaluate three different control schemes for a new musical instrument / application. In the application, users are able to make and loop sequences of notes to create rhythms and melodies. The first control scheme participants will utilises a computer keyboard and trackpad to create the sequences and loops. The second control scheme uses an iPod Touch device with the accelerometer and various touch gestures to interact. The third control scheme uses an iPad with various touch gestures for interaction. The investigators are interested predominantly in user opinions of which scheme and device is most enjoyable, intuitive and relevant to the specific installation context and what changes would benefit the installation as a whole.

Results

Results from the study may be published in academic journals, conferences, and technical reports. The results will also be published in the primary investigator's thesis at the end of his Masters. Any participants who are interested in seeing the overall results of the study can provide their email address on the consent form, and information will be sent to them.

We would like to sincerely thank you for your participation in this user study.

Researchers

Principal Investigator
Timothy Barraclough
Masters Candidate
School of Engineering and Computer Science
Victoria University of Wellington
Office: CO239, Cotton Building
Victoria University of Wellington
Email: barractimo@myvuw.ac.nz

Investigator
Ajay Kapur
Supervisor
School of Engineering and Computer Science
Victoria University of Wellington
ajay.kapur@ecs.vuw.ac.nz

Investigator
Ted Apel
Supervisor
New Zealand School of Music
Victoria University of Wellington
ted.apel@nzsm.ac.nz

Investigator
Dale Carnegie
Supervisor
School of Engineering and Computer Science
Victoria University of Wellington
dale.carnegie@ecs.vuw.ac.nz

A.3.2 Questionnaire for Full Scale User Study

,

VICTORIA UNIVERSITY OF WELLINGTON

USER INTERFACE QUESTIONNAIRE

This questionnaire is designed to survey user preference for a custom installation and two methods of interaction on multiple devices. Please fulfil this questionnaire after interacting with the installation using both the computer mouse and the supplied iOS devices. Feel free to ask any further questions or request clarification of the questions asked. Your answers may be used as part of a publication regarding the effectiveness of this installation and the presented methods of interaction.

Please check one box per statement to indicate your level of agreement.

Computer Interaction

After using the mouse to interact with the installation please answer the following questions:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The method of interacting with the installation using the keyboard and trackpad was simple.	☐	☐	☐	☐	☐
It was made obvious how to interact with the installation (using the mouse).	☐	☐	☐	☐	☐
It was simple to access the full range of interaction available.	☐	☐	☐	☐	☐
It was obvious how my mouse interactions and gestures were represented **sonically**.	☐	☐	☐	☐	☐
It was obvious how my mouse interactions and gestures were represented **visually**.	☐	☐	☐	☐	☐

If there are any specific interaction or gestures you thought should have been available, please list them here:

If there were any interactions or gestures that were difficult to accomplish, please list them here:

iPad Interaction

After using the supplied iPad to interact with the installation please answer the following questions:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The method of interacting with the installation using the iPad was simple.	☐	☐	☐	☐	☐
It was made obvious how to interact with the installation (using the iPad).	☐	☐	☐	☐	☐
It was simple to access the full range of interaction available.	☐	☐	☐	☐	☐
It was obvious how my interactions with the iPad were represented **sonically**.	☐	☐	☐	☐	☐
It was obvious how my interactions with the iPad were represented **visually**.	☐	☐	☐	☐	☐

If there are any specific interaction or gestures you thought should have been available, please list them here:

If there were any interactions or gestures that were difficult to accomplish, please list them here:

iPod Touch Interaction

After using the supplied iPod to interact with the installation please answer the following questions:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The method of interacting with the installation using the iPod Touch was simple.	☐	☐	☐	☐	☐
It was made obvious how to interact with the installation (using the iPod Touch).	☐	☐	☐	☐	☐
It was simple to access the full range of interaction available.	☐	☐	☐	☐	☐
It was obvious how my interactions with the iPod Touch were represented **sonically**.	☐	☐	☐	☐	☐
It was obvious how my interactions with the iPod Touch device were represented **visually**.	☐	☐	☐	☐	☐

If there are any specific interaction or gestures you thought should have been available, please list them here:

If there were any interactions or gestures that were difficult to accomplish, please list them here:

Can you identify any key differences that affected your interaction between the two iOS Devices (iPod Touch & iPad) :

Interaction Method Comparison

The following questions look to compare the methods of interaction with the installation:

	iPod Touch	iPad	Mouse	None of them
Which method of interaction was the easiest to use?	☐	☐	☐	☐
Which method of interaction was the most fun?	☐	☐	☐	☐
Which method of interacting allowed you to be more creative?	☐	☐	☐	☐
Overall, which method of interacting did you prefer?	☐	☐	☐	☐

User Satisfaction

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I was satisfied in the amount of control over the **sonic** elements of the system	☐	☐	☐	☐	☐
I enjoyed the sonic result of the system	☐	☐	☐	☐	☐
I was satisfied in the amount of control over the **visual** elements of the system	☐	☐	☐	☐	☐
I enjoyed the visual result of the system	☐	☐	☐	☐	☐
The combination of audio, visual and interactive elements was coherent	☐	☐	☐	☐	☐
I felt as though I was having a creative experience	☐	☐	☐	☐	☐

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I felt in control of my creative experience	☐	☐	☐	☐	☐

Which method of interacting, in your opinion, allowed more creative control? (please circle)

iPad Device iPod Device Mouse

Given the obvious differences in portability of the control schemes, if you had access to all 3 schemes, which, do you believe, you would use most frequently? (please circle)

iPad Device iPod Device Mouse

Additional Questions

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I have had prior musicianship training with traditional acoustic instruments	☐	☐	☐	☐	☐
I have had prior musicianship training in electronic music	☐	☐	☐	☐	☐
I have seen or used systems similar to this before	☐	☐	☐	☐	☐

If you have used similar systems to this, please list them here (if you remember their names):

Do you have any suggestions for improving the installation or interaction design?

Please use the space below for any further comments or remarks on your experience with the interface.

A.4 Final User Study

A.4.1 Information Sheet for Final User Study

VICTORIA UNIVERSITY OF WELLINGTON

Interface Interaction User Study Information Sheet

Introduction

The objective of this user study is to gather impressions of Pyxis Minor, a new interface/instrument intended for collaborative musical experiences. The interface is specifically designed to be of interest to people of varying prior musicianship training and offers varying levels of control over musical parameters. The primary goal of this study is to understand the extent of user's collaborative engagement with the interface.

This user study has been approved by Victoria University's Human Ethics Committee.

Participation

We are inviting pairs of participants to explore an interactive, collaborative musical and visual installation by using provided iOS devices. We will then provide a short questionnaire with questions pertaining to the usability of the interface at providing a satisfactory level of control, an enjoyable, musical experience and the quality of the collaborative aspects of the experience.

Should any participants feel the need to withdraw from the project, they may do so without question at any time before the data is analyzed. If a participant wishes to withdraw they may do so before March 3rd 2015, and should contact the primary investigator via the email address provided below.

Responses collected will be evaluated and may be quoted anonymously in a written report. It will not be required to be identified personally. All material collected will be kept confidential. No other persons, besides the investigation team, will have access to the questionnaire data.

The interaction and questionnaire survey will take approximately 10-15 minutes to complete.

The User Study Method

Participants will be asked to first familiarise themselves with the instrument / application to gain an understanding of how their interactions create sounds.. After participants are familiar with the application two instances of the instrument will be networked - one for each participant. In the application, users are able to make and loop sequences of notes to create rhythms and melodies. The application platform will be iPad devices. The investigators are interested predominantly in the user satisfaction of the user using the application by themselves vs in a collaborative manner. Additionally, questions will be asked as to the effectiveness of the application user interfacing and suggestions for improving either the solo or collaborative aspects of the application.

Video demonstrations of the application can be found at the following addresses:

`https://vimeo.com/109660076`

https://vimeo.com/113478071

Pyxis Minor can be downloaded from the Apple App Store at:

https://itunes.apple.com/us/app/pyxis-minor/id931393910?ls=1&mt=8

Results

Results from the study may be published in academic journals, conferences, and technical reports. The results will also be published in the principal investigator's thesis at the end of his Masters. Any participants who are interested in seeing the overall results of the study can provide their email address on the consent form, and information will be sent to them.

We would like to sincerely thank you for your participation in this user study.

Researchers

Principal Investigator
Timothy Barraclough
Masters Candidate
School of Engineering and Computer Science
Victoria University of Wellington
Office: CO239, Cotton Building
Victoria University of Wellington
Email: barractimo@myvuw.ac.nz

Investigator
Ajay Kapur
Supervisor
School of Engineering and Computer Science
Victoria University of Wellington
ajay.kapur@ecs.vuw.ac.nz

Investigator
Ted Apel
Supervisor
New Zealand School of Music
Victoria University of Wellington
ted.apel@nzsm.ac.nz

Investigator
Dale Carnegie
Supervisor
School of Engineering and Computer Science
Victoria University of Wellington
dale.carnegie@ecs.vuw.ac.nz

A.4.2 Questionnaire for Final User Study

,

VICTORIA UNIVERSITY OF WELLINGTON

COLLABORATIVE APPLICATION INTERACTION QUESTIONNAIRE

This questionnaire is designed to survey user satisfaction for a new instrument / application intended for collaborative use. Please fulfil this questionnaire after interacting with the application both by yourself and with your paired participant. Feel free to ask any further questions, request clarification of the questions asked or ask for assistance if necessary. Your answers may be used as part of a publication regarding the effectiveness of this installation and the presented methods of interaction.

Please check one box per statement to indicate your level of agreement.

Solo Interaction

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I enjoyed the sonic result of the system	☐	☐	☐	☐	☐
I enjoyed the visual aesthetic of the system	☐	☐	☐	☐	☐
I felt as though I had a creative experience	☐	☐	☐	☐	☐
The application is easy to use	☐	☐	☐	☐	☐

If there are any problems or difficulties you had please note them here:

Collaborative Interaction

After interacting with the application with your paired participant, please fill in the following:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
It was easy to setup the collaborative technology	☐	☐	☐	☐	☐
I enjoyed the collaborative aspect of the application	☐	☐	☐	☐	☐
I felt as we were making music **together**	☐	☐	☐	☐	☐
The collaboration felt like a social experience.	☐	☐	☐	☐	☐
I preferred the **sonic** result of the collaboration over my solo interaction	☐	☐	☐	☐	☐
The sonic result was **messier** as a result of the collaboration	☐	☐	☐	☐	☐
I felt in control of my creative experience	☐	☐	☐	☐	☐
We were verbally communicating to discuss musical ideas	☐	☐	☐	☐	☐
We were verbally communicating to teach each other techniques or interactions	☐	☐	☐	☐	☐
Working together allowed me to be more creative	☐	☐	☐	☐	☐
I was more focussed on my own interaction and music, rather than the collaborative interaction and music.	☐	☐	☐	☐	☐
This application inspired my creativity	☐	☐	☐	☐	☐
Playing collaboratively inspired my creativity more than playing solo	☐	☐	☐	☐	☐

Was there any aspect of the application that was particularly difficult or hard to accomplish during your collaborative interaction?

Do you have any suggestions for improvements to the collaboration or additional ways of collaborating?

Interaction Comparison

The following questions look to compare the methods of interaction with the installation:

	Solo	Collaborative	Neither
Which interaction, in your opinion, sounded the best?	☐	☐	☐
Which interaction was the most fun?	☐	☐	☐
Overall, which method of interacting did you prefer?	☐	☐	☐

	1	2	3	4	5+
What number of networked devices / players do you think would result in the most fun experience?	☐	☐	☐	☐	☐
What number of networked devices / players do you think would result in the best sounding music?	☐	☐	☐	☐	☐
What number of networked devices / players do you think would result in the most creative experience?	☐	☐	☐	☐	☐

Additional Questions

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I have had prior musicianship training with traditional acoustic instruments	☐	☐	☐	☐	☐
I have had prior musicianship training in electronic music	☐	☐	☐	☐	☐
I have seen or used systems similar to this before	☐	☐	☐	☐	☐

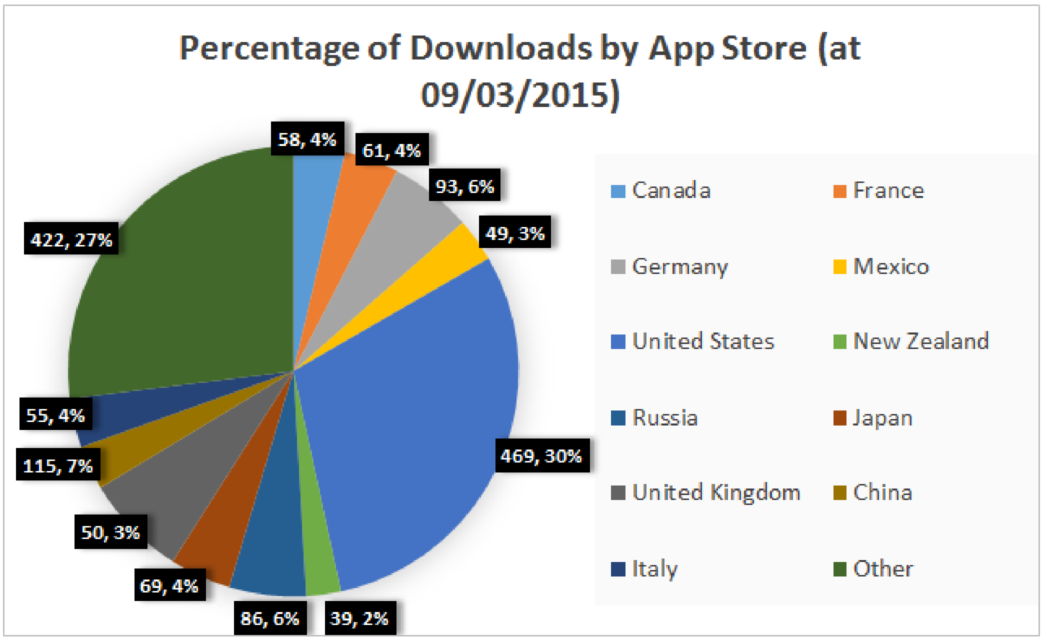
If you have used similar systems to this, please list them here (if you remember their names):

Do you have any suggestions for improving the application or any further comments?

Appendix B

Pyxis Minor Download Reports

B.1 Downloads by App Store



B.2 Pyxis Minor Downloads Per Month