

Pyxis Minor: App Design for Novel Social Music Experiences

Timothy J. Barraclough(1), Dale A. Carnegie(1), Ajay Kapur(2)

Victoria University of Wellington (1), California Institute of the Arts (2),
Wellington, New Zealand, Valencia, CA. USA.

barractimo@myvuw.ac.nz, dale.carnegie@vuw.ac.nz, akapur@calarts.edu

Abstract

Pyxis Minor is a social musical application built for iOS and Mac OS devices. It is intended to be of use for people of varying musical backgrounds and knowledge levels by providing a low barrier to entry for the creation of electronic music and by emphasizing the playful nature of music making. Pyxis Minor achieves disruption of existing electronic music performance and creation paradigms in order to posit a democratization of electronic music processes. This paper outlines the user interfacing principles and the design architecture which includes the repurposing of a game engine in order to create a unique social music experience.

Keywords

Sound synthesis, instrument design, app design, social music experiences, digital signal processing, user interfacing, game design.

Introduction

“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 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.

This paper presents Pyxis Minor, a new, social musical application that disrupts established models of electronic music creation and performance. It does this by providing an enjoyable, creative platform for people of any background to explore and create electronic music by focussing on the playful aspect of the creation process in a way that was previously only available to electronic musicians. Pyxis Minor does not attempt to provide shortcuts for beginners to create electronic music expertly, instead it offers a unique platform for users of

any experience level to experience an intuitive way of thinking about and constructing electronic music and subsequently attempt to democratize these electronic music processes.

Although the processes used to create Pyxis Minor are based on engineering principles, the primary conception is that the application itself is a work of software art. Pyxis Minor, as an artwork, seeks to give users an aesthetic experience in order to challenge established paradigms of electronic music creation. It does this by allowing the user to shape their own musical experience through unstructured, free play. This kind of playful user interaction is the primary medium through which Pyxis Minor becomes a unique interactive artwork. As an artistic medium, application development has an interesting issue; the success of an application as an artwork is inevitably tied to the structural integrity of the design and system architecture. If Pyxis Minor fails to present a cohesive and intuitive user experience, the success of the application both as a work of art and as an attempt to democratize electronic music processes will be compromised. For this reason, the usage of engineering principles is paramount to the success of the Pyxis Minor.

Detailed in the paper are aesthetic and user interfacing (UI) principles that guided the design of Pyxis Minor, the justification behind the principles used. This will be accomplished by establishing the context for the application before illustrating the main aesthetic decisions that have made Pyxis Minor an effective application.

Related Work

One of the main aims of Pyxis Minor is to create a playful musical experience for one or more users that can be enjoyed regardless of prior music performance experience. The key difficulty here is creating sensible user interfacing that appeals to users with varied expectations of what musical interaction should entail. Blaine and Fels discuss this problem as a matter of balancing complexity and expressivity. By increasing the expressivity of a musical controller or interface requires an increase in complexity - which inevitably makes these interfaces more difficult for a novice to engage with. [2] For this reason, in the field of electronic, collaborative musical interface/instrument design, frequently instruments or interfaces are made specifically for demographics based upon their prior experience or training, with few interfaces that cater to people of varied music performance experience. For example, an

early example of collaborative electronic performance, The Hub, required the participants to have a strong knowledge of computer music technology to participate. [3] Alternatively, interfaces such as Beatbugs, are used primarily as pedagogical tools for teaching rhythmic concepts to children who may not possess knowledge of music theory through the process of collaborative electronic music performance. [4] Although Beatbugs does make an effort to connect novices and experts, it makes an explicit distinction between the two, instead of recognizing the difference as separate ends of a scale. Additionally, the experiential value for the expert is predominantly in the pedagogical process, rather than the musical process. [4,5]

We can overcome this difficulty with several strategies. Some facets of this dichotomy can be resolved by relying on familiar, extramusical physical interactions to form the basis of an instruments user interaction. The user is able to focus more upon the causal result of their interaction, rather than the process of interaction.

The increasing prevalence of smartphones and tablets over the last decade has created a niche platform for music technology. The lower cost to entry for developers (over creating hardware) has translated to a lower cost of entry for musicians (beside the initial cost of the hardware device), allowing for music technology to appeal to a larger demographic. Additionally, the distribution avenues of the existing smartphone and tablet app stores has provided a platform for simple distribution of new musical instruments and interfaces. These reasons and the ubiquity of these devices make smartphones and tablets an excellent platform for the development and distribution of a social musical performance interfaces.

By examining the existing market, we can see there is an issue in the market of social musical performance applications. For example, Korg's WIST (Wireless Sync-Start Technology) allows inter-device syncing and communication and provides a strong platform for the development of a collaborative musical experience, however the majority of the applications listed on the WIST website are applications that emulate existing hardware instruments. [6] Kell and Wanderley observed that 282 of the 337 musical applications for iOS they examined used musical metaphors based on existing instruments or interfaces. [7] This is problematic for multiple reasons.

Unfortunately many of the modes of interaction of existing, physical interfaces & instruments translate poorly to a flat touch-screen. For example, rotary knobs are a common feature on hardware synthesizers that require a circular motion, which is difficult to achieve without tactile feedback. Due to the difficulty of achieving the motion, there is no standardized method of interacting with these as UI elements, yet a large amount of the emulations feature rotary knobs. It is often the case that hardware interfaces are effective because they are purposely designed for the specific hardware. Translating these musical metaphors may allow for familiarity for the user (if the metaphor is sufficiently recognisable), however they might not be more expressive than metaphors created specifically for touch-screens.

Most of these emulations are presented as tools, with a primary function of aiding in the creation of music as a product.

In doing this, however, they ignore the creation of music as a process with it's own purpose and value that can be enjoyed for its own sake, instead focusing on the musical result. The result of this is that the majority of these emulations are proposed as tools, rather than toys.

The emulations alienate new users who are unfamiliar with the hardware counterpart. This leads us to believe that novices are not the target demographic of these hardware emulations. These emulations prioritize affordability and portability over novel experience and as such, contribute to maintaining the barrier for electronic music performance as unapproachable for users without prior, related experience or knowledge.

Pyxis minor combines affordability, portability, novel experience and low barrier to entry in a single application that is accessible to users of any prior music performance experience in order to democratize electronic music performance processes. This democratization disrupts the standard electronic performance models by providing a distributed, democratized creative platform for electronic music creation and exploration.

Pyxis Minor

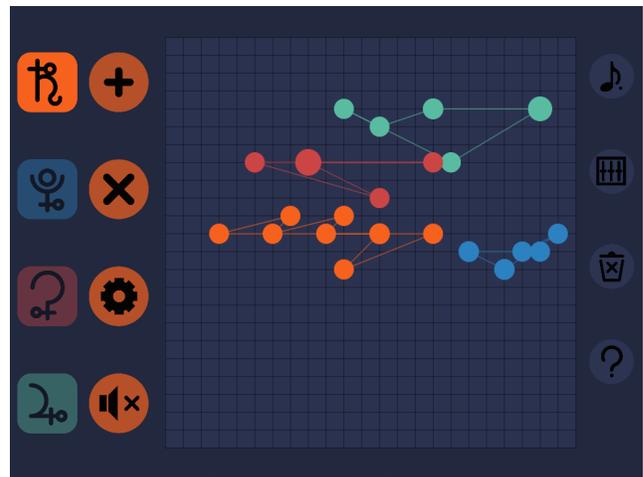


Figure 1: The main screen of Pyxis Minor with 4 sequences playing back

Pyxis Minor is a social music application built for iOS and Mac OS devices. By existing as both an application, and a work of art, Pyxis Minor gives users a chance to explore an aesthetic in a playful manner, in the hopes of suggesting the movement towards the democratization electronic music processes. The visual aesthetic is based upon an outer-space theme to echo a sense of vastness and exploration as a parallel for human creativity.

In Pyxis Minor users create sequences of musical notes (defined by pitch and velocity pairs) by defining nodes upon a grid, where the x-axis defines the pitch and the y-axis defines the velocity. Users input the nodes by touching the screen or using a mouse cursor to click the grid. As the nodes are input they are quantized to clock pulses in real time. As the

user selects the first node of the sequence, or alternatively a sequence loop button, the sequence of nodes that have been input will begin looping playback how it was input, quantized to to the clock pulses. The instrument sequences are vaguely reminiscent of constellations to further reinforce the aesthetic direction of the application.

Users are able to create up to 4 sequences, that are distinguished by color, from buttons on the left hand side of the main application screen (as can be seen in figure 1). Each sequence/color functions as its own instrument/synthesizer with multiple digital audio generation and manipulation parameters available to the user (such as stereo pan, pitch octave, envelope and synthesis waveform).

Additionally, controls enable the user to play the instruments live, without creating looped sequences and to apply digital signal processing (DSP) to the instruments to further manipulate the resulting music. A small tutorial button guides the user through a 20 second demonstration of the application's core functionality.

Users are able to wirelessly connect to nearby devices in order to share and sync the quantization clock, in order to create music together. Following is an in-depth look at the key aesthetic design principles of Pyxis Minor.

Aesthetic Design

In Pyxis Minor, a combination of 3 design principles is crucial in the absence of a widely established musical interaction metaphor (such as piano keys or drum pads). Kell and Wanderley suggest that there needs to be a transparency between musical mapping and the user in order for a musical app to be easy to understand for a beginner. In the absence of a musical mapping metaphor from an existing musical instrument or interface, Pyxis Minor uses familiar elements described by Kell and Wanderley, specifically mapping pitch to x-axis movement (low to high from left to right) and mapping volume on the y-axis (lower volumes equating to the bottom of the screen). [7] The three main principles guiding the user interface construction in Pyxis Minor are minimalism, clarity and stylization.

The visual design was created to be minimal, using pictographs instead of typed labels where possible. This makes the application more accessible to foreign languages without the need for localization. The visual design is minimal in the sense that although the application has many features, they are obscured where possible if they do not add expressivity to the application, or their meaning is unclear. Additionally, where possible the GUI displays only relevant functionality simultaneously, whilst refraining from giving the user too many menus to navigate through.

The design principle of clarity is reflected in the visual design relying upon pictographs that are somewhat universal in their meaning, for example, adding a sequence uses an addition sign, looping a sequence uses a circular, looping arrow and deleting all sequences uses a trash-can pictograph. The two exceptions to this are the waveform selection buttons, as they are visual representations of the literal waveforms, and the effect selection buttons, as there is no standardized pictograph for audio effect types. In these cases, the third design concept is utilized.

Stylization is intended to aid users by making connections between concepts through visual aids, as is the case with the audio effect pictographs. Stylization is also present in the subtle use of animations and transitions between interface screens to aid user understanding of the application mechanics and give overall polish to the application. Together, these three core design principles create a cohesive user experience, where interaction is fun, responsive and intuitive.

Pyxis Minor is an application that attempts to democratize electronic music processes. It does this by providing an opportunity for users of varying musical backgrounds to experience a fun and creative way of making music. Users can structure their own musical experience as they see fit, allowing for a solitary or collaborative experience.

The application is built around a belief in the importance of playfulness. This guides the design of the application from the bottom up, including the technologies used to create the experience as well as key details of the interaction. Structurally, Pyxis Minor includes a foundation built from a video game engine, yet, unlike most video games, does not provide a goal orientation to drive the user experience. While the lack of a clear purpose or goal can sometimes hinder this mode of interaction, it substantially furthers the other key principle by distributing and democratizing control of the experience and the outcome. There are no features for saving sequences or preset sound design, rather, the user is encouraged to explore with the application and avoid permanence by continually changing and manipulating the resulting music and sound design.

The application is based upon the idea that satisfaction of electronic music can be found in the processes, as opposed to only the products. Pyxis Minor can be thought of as a toy, rather than a tool and as such, the resulting social music performance experience of the user(s) can be as important as the sound and music that Pyxis Minor helps them create.

Future Work

Although publicly available for free on the Apple App Store, there is still future work to be done on Pyxis Minor. An implementation of a step sequencer like interface would allow users to modify the temporal aspect of sequences.

An additional layer of user controllable audio effects to allow further musical expressivity. This audio effect layer will go in place of the reverberation effect. By sharing the effect parameters over the network connection another element of co-operative and collaborative play can be introduced.

Early experiments in using accelerometer data to manipulate effect parameters is promising for providing additional ways of controlling the resulting audio. By using conformal mapping techniques applied to accelerometer data input, we can provide new expressive element that makes use of iOS devices specific capabilities. By applying these techniques to sound manipulation and effect parameters Pyxis Minor will add another dimension of expressivity to further user's enjoyment and creativity.

The main architecture of the application can be repurposed in order to create a suite of applications that are all able to be used wirelessly together. This would allow users further freedom by selecting their own preferred method of interaction.

This could be accomplished by rewriting the logic of the system and changing the audio presets, whilst leaving the game engine, audio engine and syncing technology intact.

Additionally, further research into the appropriation of game engines to create playful musical experiences could be pursued by investigating other popular game engines and their integrated audio engines.

References

- [1] Schlitz, Moritz. 1988. "On the Meaning of Life." In *Life and Meaning*, edited by Oswald Hanfling, 60–73. New York: Basil Blackwell.
- [2] Blaine, Tina, and Sidney Fels. 2003. "Contexts of Collaborative Musical Experiences." In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression*, 12934. National University of Singapore. <http://dl.acm.org/citation.cfm?id=1085745>.
- [3] Gresham-Lancaster, Scot. "The Aesthetics and History of the Hub: The Effects of Changing Technology on Network Computer Music."
- [4] Weinberg, Gil, Roberto Aimi, and Kevin Jennings. 2002. "The Beatbug Network: A Rhythmic System for Interdependent Group Collaboration." In *Proceedings of the 2002 Conference on New Interfaces for Musical Expression*, 16. NIME 02. Singapore, Singapore: National University of Singapore. <http://dl.acm.org/citation.cfm?id=1085171.1085197>.
- [5] Weinberg, Gil, and Scott Driscoll. 2005. "Iltur: Connecting Novices and Experts through Collaborative Improvisation." In *Proceedings of the 2005 Conference on New Interfaces for Musical Expression*, 1722. National University of Singapore. <http://dl.acm.org/citation.cfm?id=1085947>.
- [6] Korg Inc. 2014. "KORG WIST — Wireless Sync-Start Technology." <http://www.korguser.net/wist/>.
- [7] Kell, Thor, and Marcelo M. Wanderley. 2013. "A Quantitative Review of Mappings in Musical iOS Applications." In *Proceedings of the Sound and Music Computing Conference 2013*, 47380. <http://tide-pool.ca/papers/Thor%20Kell%20-%20SMC%202013.pdf>.

Bibliography

Blaine, Tina, and Sidney Fels (2003) "Contexts of Collaborative Musical Experiences."

Gresham-Lancaster, Scot (1998) "The Aesthetics and History of the Hub: The Effects of Changing Technology on Network Computer Music."

Kell, Thor, and Marcelo M. Wanderley (2013) "A Quantitative Review of Mappings in Musical iOS Applications."

Korg Inc. (2014) "KORG WIST — Wireless Sync-Start Technology."

Scarth, Greg, and Roger Linn (2013) "Roger Linn On Swing, Groove & The Magic Of The MPCs Timing."

Schlitz, Moritz (1988) "On the Meaning of Life."

Weinberg, Gil, Roberto Aimi, and Kevin Jennings (2002) "The Beatbug Network: A Rhythmic System for Interdependent Group Collaboration."

Weinberg, Gil, and Scott Driscoll (2005) "Iltur: Connecting Novices and Experts through Collaborative Improvisation."

Authors' Biographies

Timothy J. Barraclough is an electronic musician and music technologist currently pursuing a Masters in Engineering at the Victoria University of Wellington. He completed his Bachelor of Arts in Philosophy, Sociology and Sonic Arts composition at the Victoria University of Wellington in 2012 and his Post-Graduate Diploma in Sonic Arts Composition in 2013. His work is focused primarily around creating new ways to consume and interact with artistic works through the use of both existing and emergent technology and the ways that technology can transform, inform and create social experiences.

Dale A. Carnegie has a BSc in Theoretical Physics and Applied Mathematics, an MSc with first class honours in Applied Physics and Electronics and a PhD in Computer Science. He is currently the Professor of Electronic and Computer Systems Engineering at Victoria University of Wellington where he is also Head of the School of Engineering and Computer Science. He heads Victoria University's Mechatronics Group which specialises in Autonomous Mobile Robotics.

Ajay Kapur is currently the director of Music Technology at California Institute of the Arts. He received an interdisciplinary Ph.D. in 2007 from University of Victoria with a focus on intelligent music systems and media technology. Kapur graduated with a BS in Engineering and Computer Science from Princeton University in 2002. A musician at heart, trained on drumset, tabla, sitar and other percussion instruments from around the world, Ajay strives to push the technological barrier in order to explore new sounds, rhythms and melodies.