

Kuatro: A Motion-Based Framework for Interactive Music Installations

David Johnson, Bill Manaris, Yiorgos Vassilandonakis, and Seth Stoudenmier

Computer Science and Music Departments

College of Charleston, South Carolina, USA

{dsjohnsol@g. | manarisb@ | vassilandonakis@ | stoudenmiersh@g.} cofc.edu

ABSTRACT

Kuatro is a development framework for motion-based interactive virtual environments. Using a Model-View-Controller (MVC) architecture and the Open Sound Control (OSC) protocol, the Kuatro offers composers, artists, and interaction designers an environment that makes it easy to develop installations utilizing motion sensors, such as the Kinect, Asus Xtion, Leap Motion, and smartphones. The framework allows tracking multiple users to help designers create collaborative (or competitive) interactive experiences. The main components of the framework are described, as well as the messaging protocol. Through OSC messaging, developers and artists are able to create any number of visual and audio interfaces as well as add any number of motion sensors to the environment. The Kuatro framework is conceived as an extension of the jythonMusic audio-visual programming environment. Herein we present the Kuatro framework in the context of the interactive multimedia art installation, *Time Jitters*. We also discuss a follow-up music composition project, called Pythagorean Tetraktys.

1. INTRODUCTION

We present a novel motion-based system, called Kuatro, which has been developed as an extension of jythonMusic, an environment for computer music education [1]. Kuatro was initially developed in the context of *Time Jitters*, a four-projector interactive installation (see Figure 1), which was designed by Los-Angeles-based visual artist Jody Zellen for the Halsey Institute of Contemporary Art in Charleston, SC, USA. While Zellen provided visual material and the overall concept design, the authors provided interaction design, sound design, and software implementation for this immersive installation. Herein, we report mainly on the sound and interaction design aspects of the installation, as the other aspects have been presented elsewhere (e.g., [2]).

The *Time Jitters* installation provides an example of the

type of installations that may be developed with the Kuatro. *Time Jitters* includes two walls displaying looping video animation, and two walls with interactive elements. The concept is to create an immersive experience for participants, which confronts them with a bombardment of visual and sound elements¹.

This project synthesizes artificial intelligence and human-computer interaction techniques with music and visual art. It utilizes invisible, computer-based intelligent agents, which interact with participants. Each person entering the installation space is tracked by a computer-based agent. The agent presents a unique image and sounds, which change as the person moves through the space.



Figure 1. This photo shows two people moving through the *Time Jitters* installation. Three of the four walls are visible. The left-most and right-most walls are interactive. Also visible is one of the two Kinects used in the installation (top left), two of the four speakers, and one of the four projectors.

As more and more people engage in the space, the interactive walls become a collage of overlapping images of different sizes and types, which are in perpetual motion, as dictated by the number of people in the space. Also sounds break free from the confined or static, highly functional role they usually fulfill in traditional textures, and take on a life of their own².

This creates rich, immersive interactive experiences, which consist of visual and auditory stimuli, resulting from the agents sensing and reacting to human

¹ See a video of *Time Jitters*, <http://goo.gl/TIfpPI>

² Hear audio generated inside *Time Jitters*, <http://goo.gl/eU1rBP>

movements. These experiences are unique every time, depending on the number and movement of participants through the space.

This installation was developed in collaboration with the College of Charleston Computing in the Arts Program [3]. This work spans several of the thematic areas of ICMC, including gestures, motion and music, interactive performance systems, interfaces for sound and music, new interfaces for musical expression, sonic interaction design, and human-computer musical interaction. It was partially funded by the US National Science Foundation and the South Carolina Arts Commission

1.1 Digital Interactive Art

Digital Art can be examined meaningfully by looking into two main key areas: Interface and Interactivity. Interface refers to the visible/tangible/audible components that allow the capture, communication and exchange of data, a juncture between independent systems, a navigational tool that allows communication between them. Interface allows and controls interactivity by designing the role of the participant and the observer. Ideally, interface is both transparent and self-explanatory, though at times it can also be “meaningful” to the user, by attracting her attention to the interface per se as an object of design, rather than looking through it.

Interactivity is a challenging term, as it refers to broad and constantly shifting ideas, and requires a multi-disciplinary approach. As such, it has a decades-long research tradition that challenges both artistic and scientific paradigms. Spiro Kiousis’ attempt at offering a definition states:

“...Interactivity can be defined as the degree to which a communication technology can create a mediated environment in which participants can communicate (one-to-one, one-to-many, and many-to-many), both synchronously and asynchronously, and participate in reciprocal message exchanges (third-order dependency). With regard to human users, it additionally refers to their ability to perceive the experience as a simulation of interpersonal communication and increase their awareness of telepresence.” [4]

This approach incorporates interactivity as both a “media and psychological variable,” and suggests a close and dynamic exchange among structure of technology, communication context and user perception.

It should be obvious that there can be several distinct approaches to designing interactivity. Claudia Giannetti distinguishes media-assisted interactivity in three models: (a) discrete or active systems, in which the user can call content up but has no influence on the transmitted information, (b) reactive systems, where external stimuli, such as user control or altered environmental conditions affect the work’s behavior based on feedback structures,

and (c) interactive systems, in which the user can influence the procedure and appearance of the work, through established temporal, spatial or content-based relations [5]. Giannetti also classifies interactivity in terms of behavior and consciousness as: (a) synaesthetic, which involves various materials and elements, such as image and sound, (b) synergetic, which takes place between different states of energy, such as environmental works, and (c) communicative/kinetic, between humans or humans and objects [6, p. 8].

Clearly, the *Timer Jitters* installation falls under Giannetti’s (c) categories in terms of both model and behavior. However, the *Kuatro* system is much more general than that, as it can be used to design different kinds of interactive experiences.

1.2 Interactivity in Music

Performing music is by default an interactive activity. Musicians rely on their musical training and performance tradition to interpret a set of instructions by a composer codified into musical notation, coordinate their actions with those around them, and constantly adjust their timing, volume, timbre and articulation in an interactive manner, whether they follow a conductor, react to each other in a chamber music setting, or engage in free improvisation unaware of upcoming musical events contributed by their band mates. Successful musical performances rely upon their ability to remain constantly engaged in the performance, and anticipate and react meaningfully to the musical gestures of others within a well-defined contextual environment.

Since its early days computer music has offered fresh approaches to timbre and technical possibilities. Its most important contribution however has been the examining and development of new ways of interactivity between composer and performer, performer and computer generated sound, sound and physical space, performer and physical space, and generated sound and audience.

In recent years, as computers have become able to handle large data efficiently, composers have been experimenting with expanding interactivity design to include physical space, human gesture and multimedia. Such interactions with the physical and visual domains have allowed computer musicians to incorporate gestures of a physical nature, which have forged new dynamic relationships between the digital environment and the external world and allowed novel structural and formal possibilities. Some composers have captured and imported such concepts into instrumental compositions via computer assisted composition software. For example, Composer Philippe Leroux used *Open Music* to transform his own handwriting captured on a Wacom tablet into pitch/rhythm patterns in his piece *Apocalypse* (2005). The handwriting was also transformed into spatialization patterns for diffusing the electronics through the surround sound speaker system [7]. Gesture capture sensors have made interactive audio-visual installation projects possible. *Virtual Pottery* uses hand gesture to craft a virtual audiovisual installation combining physical metaphors of 3D pottery objects and sound shapes [8]. *SoundLabyrinth*, a multichannel

interactive audiovisual artwork explores the relationship between physical gesture and sound. Using the Kinect's skeleton tracking algorithm, it maps subject locations to XYZ coordinates, and generates sounds fed to an elaborate speaker system in a dome-spaced environment [9].

Another example is Tom Mays' *Acousmeaucorps* interactive sound installation, which generates spatialized sound by tracking movement and position data through a video camera connected to a computer running Max/MSP/SoftVNS. The human body becomes a performance instrument, generating and triggering sounds and building musical sequences through walking, running, moving arms, and even flexing fingers. This installation aims to encourage visitors to become creators and performers of their own music and dance, by moving through the space and experiencing the generated sounds [10].

Interactive virtual environments may also be used to analyze, represent, and interact with large data sets, as in the *AlloSphere*. The *AlloSphere* is a 3-story spherical structure designed as a fully immersive multimedia virtual environment for scientific studies and artistic performances. It was conceived as an instrument for interpreting big data visually and aurally. *AlloSphere* is based on the Device Server, a development framework for mapping distributed interactive devices to visual and auditory output using an OSC communication protocol [11].

2. INTERACTION DESIGN

As mentioned earlier, *Time Jitters* was a multimedia art exhibit for the Halsey Institute of Contemporary Art. The first iteration of the Kuarto was developed to support the environment designed for this exhibit. The environment consisted of a 20' x 14' (6.1m x 4.3m) room with four projectors displaying content on each of the four walls of the room (see Figures 1 and 2).

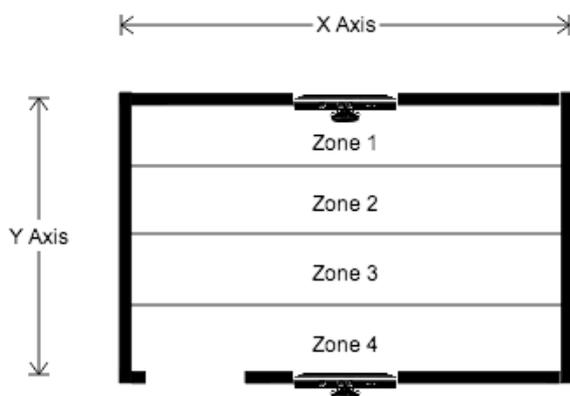


Figure 2. *Time Jitters* space. The interactive walls are denoted with Kinect images.

2.1 Interaction Language

The overhead view of the space was mapped to an XY plane and divided into four horizontal zones. Using the Kuarto framework, users' locations in the *Time Jitters* space were tracked using two Kinects. Each of the Kinect's output data was processed by separate Kuarto Client applications running on different workstations. The clients sent sensor data to the Kuarto Server for coordination within the Virtual World. After data coordination, the Kuarto Server broadcast the users' virtual world locations to all views registered with the server.

In developing the *Time Jitters* exhibition we took advantage of the framework to create two views, an audio view and a visual view. The independent development of the views afforded the visual artist and the composer freedom to express their artistic visions. For each view we mapped user events to specific actions. These are further explained in the following sections.

2.1.1 User Events

The *Time Jitters* exhibit supported three user events:

- **User Enters Space** - When a user enters the space, the Kuarto Server recognizes the user as a new user, adds them to the virtual world coordinate space, and sends a *newUser* message to all registered views (see section 4 for more information about the messaging protocol).
- **Users Moves** - As users move inside the space, the Kuarto Server updates their coordinates in the virtual world coordinate space, and sends *userCoordinates* messages for each user to all registered views.
- **User Exits Space** - When a user exits the space, the Kuarto Server removes them from the virtual world coordinate space, and sends a *lostUser* message to each of the registered views.

3. SOUND DESIGN

With the predefined user events in mind, we designed modular soundscapes that worked well when there was one user in the room as well as when there were multiple users in the room.

Soundscapes were built by manipulating and layering pre-recorded sounds emanating from crystal glasses. These glasses were either

- “finger-bowed”, or
- struck with various objects within the resonant space of a grand piano.

The sound environment was meant to match the physicality of the enclosed visual space of the installation, with the perceptible piano resonance present

in all the recorded sounds. At the same time, the sounds were specifically designed to contradict the raw quality of the news images by introducing a sense of familiarity and calmness³.

Recorded sounds were heavily processed with standard techniques, such as pitch shifting, time stretching, compression, equalization and digital effects to produce a wide palette of timbres that were organically related yet diverse and interesting.

3.1 The Glaser - A Sound Exploration Instrument

In order to facilitate exploring various sounds and quickly manipulating their attributes (e.g., frequency, volume, and panning), we designed a specific instrument, called the “Glaser.”⁴ The “Glaser” consists of three displays with 8 sliders each (see Figure 3). These displays work in parallel, i.e., the first slider (across all displays) controls the first assigned sound, whereas the second slider (across all displays) controls the second sound, and so on.

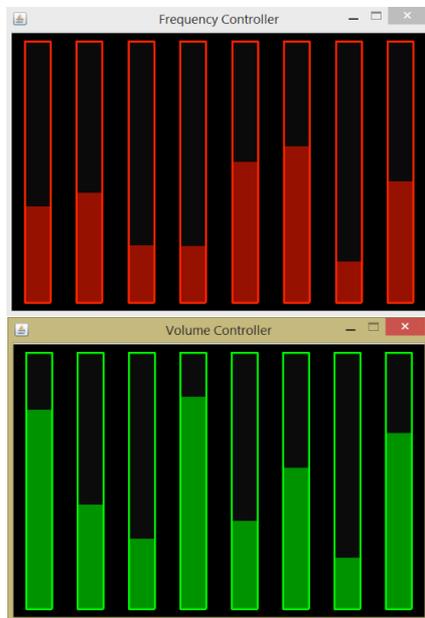


Figure 3. Two of the three Glaser displays controlling frequency and volume (panning is not shown) for 8 different sounds simultaneously.

“The Glaser” allows the user to control volume, frequency, and panning of eight audio samples simultaneously. Each display’s faders are oriented to match their natural orientation, i.e., frequency and volume are vertical (low to high), and panning is horizontal (left to right). As each fader is adjusted, the corresponding audio is altered accordingly and that value is displayed in a textual output window (not shown here). It is important to note that this does not alter the actual audio file, but only what is heard dynamically. The

outputted values may then be used in the code that drives the installation.

The initial motivation behind “The Glaser” was to provide an environment to aid in the creation of the various sound states in the Time Jitter project. However, it is quite useful for other related projects.

3.2 Sound Assembly

Drawing from this library of processed sounds, textures were built that aimed to be frequency-band specific and harmonically coherent, approximating overtone series spacing, bell harmonies, or even tonal-referential chords.

A “zero-state” texture was created to play continuously when the installation space was empty of users with the intent to attract the ear of passers-by and entice them into entering the space. This was intentionally rich and wide in frequency range, comprising eight layers of loops, each set to a different and quite long loop cycle, so that it gave the illusion of being non-repetitive, somehow stable and catchy. The same sound material used in the “zero-state” texture became the building blocks for the user-based sounds (see Figure 4).

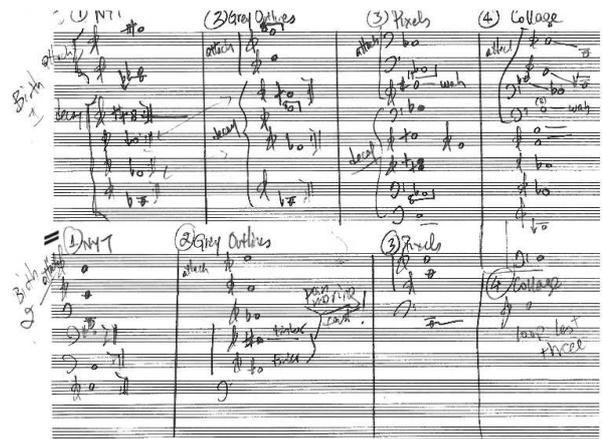


Figure 4. Sound design using sound building blocks in conjunction with user events and Kuatro space zones (see Figure 2).

Each user event was tied to a musical attribute for manipulation. Sound events were built with several layers of the sounds mentioned above, organized into two categories: a) “entrance” sounds and b) “in-the-zone” sounds. “Entrance” sounds were conceived with a strong, identifiable attack component that would occur once, triggered by the entrance of a user into one of the predefined zones in the installation. They were timbrally rich and mostly percussive, so that the user could quickly associate them with the corresponding shifts in the visual elements, as well as the specific spot they occupied within the installation space. Before the “entrance” sounds died away, “in-the-zone” sounds were introduced in a crossfade manner. These were multilayered textures of sinetone-like sounds, tuned in ratios close to the overtone series, occupying specific frequency regions, and planned to complement each other in case of multiple

³ Hear an example, <http://goo.gl/eU1rBP>

⁴ “The Glaser” is named after Matthew Glaser, a freshman computer-music class student, who designed an early version of the instrument.

users. They were looped in irregular time intervals so that the resulting timbre was constantly shifting with the intention that if a user stayed within a specific zone, they could choose to focus on the shifting timbre and experience the internal motion of the sound.

- **User Enters Space** - When the user entered the space, the zone's "entrance" sound was triggered, cutting off the "zero-state" soundscape, and quickly crossfading into the looped "in-the-zone" texture.
- **User Moves** - As the user moved within the space, once the "entrance" sound had faded, the "in-the-zone" sounds come to the foreground, which "follow" the user by panning between the speakers, as the user moves laterally within a zone.
- **User Exits Space** - Upon exiting the space, the "in-the-zone" sounds fade out and the "zero-state" soundscape fades back in.

A lot of care was taken to give each additional user that may enter the space a different sound identity, so that they could quickly associate their movements in the space with their own assigned sounds, and at the same time be aware or even surprised of how their sound worked with or against the sounds of other users, therefore creating motivation for users to move towards or away from each other. Each time an additional user entered, the overall sound of the installation space would appear to be radically altered, yet within minutes, a new blend of stability was established. The users were thus motivated towards both making exaggerated moves and standing still.

3.3 Visual Interaction Design

For completeness, we provide a quick overview of the interaction design we provided to the visual artist. The artist, Jody Zellen, was commissioned by the Halsey Institute, as part of her *Above the Fold* series. Using images taken from the front page of the New York Times over a year period, Zellen selected a number of images and digitally processed them until they became her own works. The *Time Jitters* exhibit was designed to show users the transitions that Zellen went through as she worked through each piece.

Initially, the artist came to us with the idea to use 5 on/off sensor plates on the floor. We explained to her the capabilities of the Kuartro system, and provided to her the above interaction design.⁵

4. THE KUATRO FRAMEWORK

Integrating motion sensors into interactive spaces is a non-trivial task that often requires a deep understanding of the sensor technology being used, thus making entry

into the field of interactive music and art difficult for composers and artists. The Kuartro has been designed to reduce the technological barriers of developing sensor-based multimedia interactive art installations so that composers, artists and interaction designers can focus on expressing their visions.

The Kuartro was designed using the Model-View-Controller (MVC) architectural pattern. Using this pattern simplifies the development of interactive systems by allowing for independent development of the various Kuartro components. This allows composers, artists and designers to focus on the components that are important to their work. It also facilitates use of various controllers, such as motion controllers like the Kinect or Asus Xtion or gestural controllers like the Leap Motion or a smartphone (e.g., iPhone). To help support heterogeneous controllers and communication among various Kuartro components, we provide a messaging API using Open Sound Control (OSC) [12].

4.1 Architecture

Using the MVC pattern the Kuartro framework defines three types of components, (a) Kuartro Clients, (b) The Kuartro Server and (c) Kuartro Views. Figure 5 shows where each component fits into the MVC architectural model. Communication between the components is handled via OSC messages (see next section).

The Kuartro Clients receive user input from the motion sensor devices, and send it to the Kuartro Server. As part of the framework, we have developed a client for the Kinect and are working on an additional client for the Asus Xtion controller.

The Kuartro Server receives data from the various clients and normalizes it, coordinates data from overlapping sensors (performing error correction through averaging), and uses the data to update a virtual model of the interaction space being tracked. As the virtual model is being updated, the server sends updated information to the Kuartro Views.

The Kuartro Views receive information about updates to the virtual model, and manipulate / update the user interfaces defined by artists and composers.

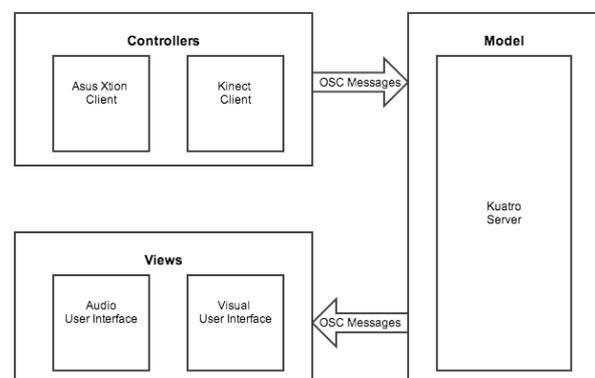


Figure 5. The Kuartro MVC Architecture.

⁵ See a video demo, <http://goo.gl/TIfpPI>

Using the view components, artists and composers can create their own interfaces to represent the data in various formats. For example, an artist may create a Visual View which displays and modifies their artwork based on user location in the virtual world; or a composer may create an Audio View which plays and modifies sounds they have designed based on user location. The Kuarto framework facilitates creating views with specific artistic objectives, while keeping them independent from the rest of the system.

The Kuarto was developed in Jython using custom music and GUI libraries [1]. By using OSC in the framework's design, additional devices and views can be developed using any language as long each component implements the Kuarto's messaging contract.

4.2 OSC Messaging API

To support distributed heterogeneous components we have designed a simple OSC protocol for the Kuarto framework. Open Sound Control (OSC) was designed at CNMAT as a transport protocol to support interactive music. Using the concept of address spaces to organize messages and parameters affords developers the flexibility in design rather than having to adhere to strict parameters that may not meet their needs. Since its development in 1997, OSC has been used in a variety of interactive instruments and exhibits [13]. OSC was chosen as the messaging framework because the Kuarto is designed for interactive spaces with heterogeneous components distributed over a network.

The Kuarto OSC protocol consists of messages that are sent from the client and received by the server at a specific OSC address. The following section provides an overview of the Kuarto OSC protocol.

4.2.1 Kuarto Client-to-Server API

Kuarto Clients may send the following OSC messages to the Kuarto Server:

- **Register Controller** - registers a controller device.
- **New User** - updates the virtual world with location information about a new user, i.e., a user that has just entered the controller's viewing space. This includes a unique user ID, and location coordinates.
- **Lost User** - indicates that a particular user has exited the controller's viewing space.
- **User Coordinates** - provides location information about a particular user as (s)he moves within the controller's viewing space.

4.2.2 Kuarto View-to-Server API

The Kuarto View-to-Server API allows views to register callback functions to be called by the Kuarto server. Using callback functions allows the views to encapsulate and hide from the server what they do with the data. The

server just calls the view functions when needed, and the views directly update themselves. Kuarto Views use the following OSC messages:

- **Register View** - this message registers a visual or sound view with the server providing connection information to the view's OSC port.
- **New User** - this registers a callback function to be called when a new user is added to the virtual world. When called, this function expects the new user ID, and the XYZ coordinates.
- **Lost User** - this registers a callback function to be called when a user is removed from the virtual world. When called, this function expects the user ID.
- **User Coordinates** - this registers a callback function to be called when a user's location has been updated in the virtual world. When called, this function expects the user ID, and the new XYZ coordinates.

4.3 Adding Kuarto Components

Using the OSC API described in the previous section adding new motion controllers and views is as simple as registering new components with the Kuarto Server. In cases where the viewing areas of devices overlap, there is a calibration step that must occur to ensure the Kuarto Server recognizes the sets of data from the multiple devices as being a single user. After calibration has occurred, as a user enters the Kuarto environment the Kuarto Server handles the coordination of data from the separate devices to into a single set of data for the individual user.

5. PYTHAGOREAN TETRAKTYS

This section describes an on-going music composition project involving the Kuarto framework. The title of this project is *Pythagorean Tetraktys*. It has a strong component of interaction between physical space, perception of sound field and movement, and human gesture. It is also a study in and an exposition on how musical form can be constructed from primitive elements following Pythagorean principles of harmony and sound synthesis.

This composition project is based on and informed by research results stemming a decade-long interdisciplinary project in exploring applications of Zipf's law and fractal geometry in the analysis, composition, and performance of musical works, e.g., [14, 15, 16, 17, 18].

The Tetraktys was very important to the Pythagoreans, as it symbolized the deeper mysteries of the cosmos and its creation. Not much is known about the Pythagoreans (they were sworn to an oath of secrecy) and the first writings about them come from hundreds of years later. However, various writings about the Tetraktys (and the Pythagoreans) have survived, indicating the significance of it, e.g., [19, 20, 21].

Due to the limited space, herein we provide an overview of the composition space and approach of the piece.

5.1.1 Requirements

Pythagorean Tetraktys uses a single human performer with a musical instrument (e.g., voice, guitar, violin, etc.). The only requirement is that the instrument is portable. A single computer running the Kuarto system, with a single Kinect and a projector, is utilized. A wireless microphone is used to capture the audio generated by the instrument and transmitted to the computer's audio subsystem.

Visually, the composition is performed on the Pythagorean tetraktys shape (see Figure 6).

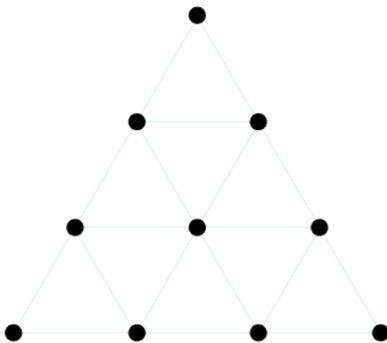


Figure 6. The tetraktys (Greek: τετρακτύς) is a triangular figure arranged in four rows – one, two, three and four points, adding up to ten. It was an important mystical symbol to the Pythagoreans.

5.1.2 Overview

The piece unfolds by having the performer moving through a virtual space created by the Kuarto and projected by the computer to a screen. The performer's movements are captured by the Kinect, similarly to the *Time Jitters* installation. As the performer steps on each of the points defining the Tetraktys, he performs short fragments of linear material, of which the computer initiates a recording. When the performer steps off a point, the computer loops and plays back the audio recorded. The performer can control the looping functionality (start recording, start looping, stop looping, delete recording) based on his/her positioning relative to a point.

The points are numbered 1-10 (starting from the top, and moving left to right). The first point (top, corresponding to 1) reproduces recorded sound as is. The second point alters the looped audio by increasing its frequency by 2, and lowering its volume by 1/2. The third point increases the frequency by 3 and volume by 1/3, and so on. In the end, the music recorded and looped by the performer creates a texture modeled after proportions similar to the harmonic (overtone) series (i.e., 1/1, 1/2, 1/3, 1/4, 1/5, and so on) producing a complex yet harmonious combination of timbres and rhythms.

Additionally, and most importantly, sounds are distributed in space (using a combination of panning and depth/reverb) in spatialization patterns that correspond to the patterns of the Tetraktys points, recreating in physical space perceivable structures that are also derived from proportions similar to those that generate the sound material. Based on the aforementioned audio-visual structure, the narrative of the piece unfolds in space and time, creating pathways in the space and musical texture that enhance the perception of the Tetraktys proportions, as well as explore inner relationships and sonorities within this space. Finally, the audience experiences the sounds through audio streaming software (e.g., Airfoil, Airfoil Speakers app) on their smartphones connected to a local WiFi network, and through individual stereo headphones.

6. CONCLUSION

We presented an overview of the Kuarto framework for developing motion sensor based virtual environments. By removing the complexity of dealing with multiple types of sensors, the Kuarto enables artists and composers to quickly and easily create any number of interactive interfaces to realize their artistic visions. The Kuarto also makes it simple for developers to introduce new types of gesture and motion sensors into the system to extend the levels of interaction available.

To further remove complexity from the implementation of interactive installations, we plan to package Kuarto clients as plug and play hardware devices. For example, we plan to package a motion sensor such as an Asus Xtion Pro with a small microcontroller such as the Raspberry Pi. Users would then simply need to configure the device with the IP address of the server.

The Kuarto has already been used in developing two interactive installations so far. The first is the *Time Jitters* visual art installation described in the introduction section. The second is the *Pythagorean Tetraktys*, which captures a performer's location within a triangular space to generate and spatialize sound. We hope to use this framework as the basis for many new interactive sound installations and instruments.

Acknowledgments

This work has been partially funded by the US National Science Foundation (grant DUE-1044861). The following individuals contributed significantly to the *Time Jitters* installation, in addition to the visual artist Jody Zellen - these are Mark Sloan, Andrew Steever, Lizz Biswell, and Karen Ann Myers. Also Dana Hughes and Kenneth Hanson have contributed to the development of the jythonMusic libraries upon which the Kuarto framework is based. Finally, Matthew Glaser provided the inspiration for the development of the Glaser instrument.

7. REFERENCES

- [1] B. Manaris, and A. Brown, *Making Music with Computers: Creative Programming in Python*,

- Chapman & Hall/CRC Textbooks in Computing, 2014.
- [2] J. Drucker, "Pause Effect," <http://goo.gl/kLwgbY>
- [3] College of Charleston, Computing in the Arts Program, <http://www.cs.cofc.edu/~manaris/cita/>
- [4] S. Kioussis. *Interactivity: a concept explication*, New Media Society, Vol. 4, No. 3, pp. 355-383, 2002.
- [5] C. Giannetti, "Media Art Net-Aesthetics of the Digital," <http://goo.gl/WZxknD>
- [6] F. Blum, *Digital Interactive Installations*, VDM Verlag Dr. Mueller e.K., 2007.
- [7] Y. Vassilandonakis, "An Interview with Phillippe Leroux," *Computer Music Journal*, Vol. 32, No. 3, pp. 11-24, 2008.
- [8] Y. Han and B. Han, "Virtual Pottery: An Interactive Audio-Visual Installation," in *Proc. of the International Conference on New Interfaces for Musical Expression (NIME)*, Ann Arbor, 2012.
- [9] M. Pedersen, "Sound Labyrinth: Exploration Of The Embodied Sublime Through An Immersive Audio/Visual Installation," in *Proc. of the 2013 International Computer Music Conference*, Perth, pp. 169-176, 2013.
- [10] T. Mays, Acousmeaucorps, <http://goo.gl/upqld1>
- [11] C. Roberts, J. Kuchera-Morin, L. Putnam, and G. Wakefield, "Dynamic Interactivity Inside the Allo-Sphere," in *Proc. of the International Conference on New Interfaces for Musical Expression (NIME)*, Sydney, 2010.
- [12] M. Wright and A. Freed, "Open Sound Control: A New Protocol for Communicating with Sound Synthesizers," *Proc. of the 1997 International Computer Music Conference*, in Thessaloniki, Hellas (Greece), 1997, pp. 101-104.
- [13] M. Wright, "Open Sound Control: an enabling technology for musical networking," *Organised Sound*, vol. 10, no. 3, pp. 193-200, Cambridge University Press, 2005.
- [14] B. Manaris, J. Romero, P. Machado, D. Krehbiel, T. Hirzel, W. Pharr, and R.B. Davis, "Zipf's Law, Music Classification and Aesthetics," *Computer Music Journal*, vol. 29, no. 1, pp. 55-69, 2005.
- [15] B. Manaris, P. Roos, P. Machado, D. Krehbiel, L. Pellicoro, and J. Romero, "A Corpus-Based Hybrid Approach to Music Analysis and Composition," in *Proc. 22nd Conference on Artificial Intelligence (AAAI-07)*, in Vancouver, BC, 2007, pp. 839-845.
- [16] B. Manaris, D. Hughes, and Y. Vassilandonakis, "Monterey Mirror: Combining Markov Models, Genetic Algorithms, and Power Laws," in *Proc. 1st Workshop in Evolutionary Music, 2011 IEEE Congress on Evolutionary Computation (CEC 2011)*, New Orleans, USA, 2011, pp. 33-40.
- [17] B. Manaris, P. Roos, D. Krehbiel, T. Zalonis, T., and J.R. Armstrong, "Zipf's Law, Power Laws and Music Aesthetics," in Li, T., Ogihara, M., Tzanetakis, G. (eds.), *Music Data Mining*, pp. 169-216. Chapman & Hall/CRC Textbooks in Computing, 2011.
- [18] B. Manaris, D. Johnson, and Y. Vassilandonakis, "Harmonic Navigator: A Gesture-Driven, Corpus-Based Approach to Music Analysis, Composition, and Performance", in *Proc. 2nd International Workshop on Musical Metacreation (MUME 2013), 9th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Boston, MA, 2013, pp. 67-74.
- [19] J. Godwin, *The Harmony of the Spheres - A Sourcebook of the Pythagorean Tradition in Music*, Inner Traditions International, Ltd., 1993.
- [20] J. James, *The Music of the Spheres - Music, Science, and the Natural Order of the Universe*, Springer Verlag, 1993.
- [21] E. G. McClain, *The Pythagorean Plato - Prelude to the Song Itself*, Nicolas-Hays, Inc., 1978.