

SEVEN ATTEMPTS AT MAGIC: A DIGITAL PORTFOLIO DISSERTATION OF  
SEVEN INTERACTIVE, ELECTROACOUSTIC, COMPOSITIONS FOR  
DATA-DRIVEN INSTRUMENTS

by

STEVEN JOSLIN

A DISSERTATION

Presented to the School of Music and Dance  
and the Graduate School of the University of Oregon  
in partial fulfillment of the requirements  
for the degree of  
Doctor of Musical Arts in Music Performance: Data-driven Instruments  
June 2018

# DISSERTATION APPROVAL

Seven Joslin

Title: Seven Attempts at Magic: A Digital Portfolio Dissertation of Seven Interactive, Electroacoustic, Compositions for Data-driven Instruments.

This dissertation has been accepted and approved in partial fulfillment of the requirements for the Doctor of Musical Arts in Performance of Data-driven instruments degree in the School of Music and Dance by:

Dr. Jeffrey Stolet	Chair
Akiko Hatakeyama	Member
Steve Owen	Member
John J. Park	Member

and

Sara Hodges	Dean of the Graduate School
-------------	-----------------------------

Original approval signatures are on file with the University of Oregon Graduate School. Degree awarded June 2018.

© 2018 Steven Joslin

This work is licensed under a Creative Commons  
**Attribution-NonCommercial-NoDerivs (United States) License.**



# ABSTRACT

Steven Joslin

Doctor of Musical Arts in Performance of Data-driven Instruments

School of Music and Dance

June 2018

Title: Seven Attempts at Magic: A Digital Portfolio Dissertation of Seven Interactive, Electroacoustic, Compositions for Data-driven Instruments.

This Digital Portfolio Dissertation presents seven compositions including text documents that explain the synthesis techniques, data mapping and routing, visual elements, the software used, all software needed to reproduce these works, and a video recording of all seven compositions. The unifying thread in my seven works is magic. The sense of magic in a live performance is the connection between artist and audience that lies beyond the immediate understanding of any work. I use this insight to create a new world inspired by sound and visuals. I perform each of these works by combining my understanding of data-driven instruments and my experience as a classically trained musician. The combination of sound design, visual composition, and a sense of magic allows me to realize these seven works. My goal is to contribute to the extensive library of electroacoustic works through my performance of my music with data-driven instruments.

## **SUPPLEMENTAL FILES**

VIDEO: CLARA VOCE

VIDEO: OURIANA'S MEDITATION

VIDEO: CHANNELING RACHMANINOFF

VIDEO: CURVES & LINES

VIDEO: EUPHONY

VIDEO: INSISTENCE OF SAMENESS (IS)

VIDEO: SUB-ROSA

SOFTWARE: CLARA VOCE

SOFTWARE: OURIANA'S MEDITATION

SOFTWARE: CHANNELING RACHMANINOFF

SOFTWARE: CURVES & LINES

SOFTWARE: EUPHONY

SOFTWARE: INSISTENCE OF SAMENESS (IS)

SOFTWARE: SUB-ROSA

# CURRICULUM VITAE

AUTHOR: Steven Joslin

## GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene  
University of Michigan, Ann Arbor

## DEGREES AWARDED:

Graduate Certificate, Nonprofit Management, 2017, University of Oregon  
Master of Arts, Media Arts, 2013, University of Michigan  
Bachelor of Fine Arts, Performing Arts Technology, 2011, University of Michigan

## AREAS OF SPECIAL INTEREST:

Nonprofit Management and Program Development  
Curriculum Development for STEAM Education  
Developing Data-driven Interfaces

## PROFESSIONAL EXPERIENCE:

Graduate Teaching Fellow, Music Technology, University of Oregon, 2015-18  
Lead Instructor, SPARK Summer STEAM camps, Lane Community College,  
2017-18  
Graduate Teaching Fellow, Digital Arts, University of Oregon, 2014-15  
Lead Instructor, Music & Engineering Summer Camp, University of Michigan,  
2014  
Adjunct Professor of Music, Concordia University, Ann Arbor, 2014  
Instructor, MPulse Performing Arts Technology Summer, University of Michigan,  
2011

## GRANTS, AWARDS, AND HONORS:

Graduate Teaching Fellowship, Music, University of Oregon, 2015-18  
Artist in Residence, Crater Lake National Park, 2015  
Graduate Teaching Fellowship, Digital Arts, University of Oregon, 2014-15  
Travel Grants, School of Music and Dance, University of Oregon, 2015-2017  
Exhibit Grant, University of Michigan, 2012

## CONFERENCE PRESENTATIONS:

SEAMUS 2016, 2017, 2018  
Seattle Maker Faire 2015, 2016, 2017  
Electronic Music Midwest 2016  
SubUrban Projections 2015, 2016

## ACKNOWLEDGEMENTS

This dissertation represents four years at the University of Oregon, for which I am grateful to have attended.

I would like to thank my dissertation committee: Jeffrey Stolet (chair), Akiko Hatekayama (member), Steve Owen (member), and John Park (member) for their support and guidance during this work and over the last four years. I would also like to thank Dr. Chet Udell for helping me to improve my Arduino skills by leaps and bounds. All of these professors have helped to shape my work, my skills, and my character in some way.

A special thanks to my brother, Dr. Art Joslin for keeping me sane by reminding that this endeavor was not meant to be easy.

A very special thanks to my partner, Theya Harvey for her endless understanding of my need for solitary confinement in order to work, **and** for her excellent proof-reading skills.

None of this would have been possible without the letter... H.

## **DEDICATION**

This work is dedicated in loving memory to three people. First, my mom, who endlessly supported my passion for music. To my dad, who taught me the value of hard work. And to my sister in-law, Diane, my first music teacher, who helped to ignite the spark of music in my life.

# TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION.....	1
II. CONCEPTUAL AND THEORETICAL FRAMEWORKS.....	11
III. DISCUSSIONS OF INDIVIDUAL PORTFOLIO COMPOSITIONS.....	26
III.1 CLARA VOCE .....	26
III.2 OURIANA’S MEDITATION.....	38
III.3 CHANNELING RACHMANINOFF .....	53
III.4 CURVES & LINES.....	67
III.5 EUPHONY .....	80
III.6 THE INSISTENCE OF SAMENESS (IS) .....	93
III.7 SUB-ROSA.....	104
SUMMARY .....	117
APPENDICES .....	119
REFERENCES CITED.....	140
SUPPLEMENTAL FILES	
VIDEO: CLARA VOCE	
VIDEO: OURIANA’S MEDITATION	
VIDEO: CHANNELING RACHMANINOFF	
VIDEO: CURVES & LINES	
VIDEO: EUPHONY	
VIDEO: INSISTENCE OF SAMENESS (IS)	
VIDEO: SUB-ROSA	

SOFTWARE: CLARA VOCE

SOFTWARE: OURIANA'S MEDITATION

SOFTWARE: CHANNELING RACHMANINOFF

SOFTWARE: CURVES & LINES

SOFTWARE: EUPHONY

SOFTWARE: INSISTENCE OF SAMENESS (IS)

SOFTWARE: SUB-ROSA

## LIST OF FIGURES

Figure	Page
1. Size comparison of FSR sensors .....	17
2. Diagram of data-driven instrument for III.1 <i>Clara Voce</i> .....	28
3. Screenshots of visual output for each section .....	33
4. Diagram of data-driven instrument for III.2 <i>Ouriana's Meditation</i> .....	40
5. Two small holes cut into acrylic for FSR and USB cables to pass .....	41
6. Raw and smoothed data streams from interface .....	44
7. Diagram of data-driven instrument for III.3 <i>Channeling Rachmaninoff</i> ...	54
8. Photos of finished Lux Codex .....	55
9. Working prototype of Lux Codex interface .....	56
10. Photo of early prototype of the cylindrical tubes .....	65
11. Diagram of data-driven instrument for III.4 <i>Curves &amp; Lines</i> .....	69
12. Performance setup for III.4 <i>Curves &amp; Lines</i> .....	76
13. Diagram of data-driven instrument for III.5 <i>Euphony</i> .....	81
14. Mapping the four corners of a Wacom tablet to align with a quadraphonic speaker system .....	91
15. Diagram of data-driven instrument for III.6 <i>Insistence of Sameness (IS)</i> ...	93
16. Examples of visual instructions from the score .....	102
17. Diagram of data-driven instrument for III.7 <i>Sub-Rosa</i> .....	105
18. Photos of Gem parts .....	107
19. Photos of Geode parts .....	108
20. Photos of Cone parts .....	109

## LIST OF TABLES

Table	Page
1. Equations used for frequency values in both solo sections of <i>Ouriana's Meditation</i> .....	50
2. Newly formed text used in III.5 <i>Euphony</i> .....	87

# CHAPTER I

## I. INTRODUCTION

This Digital Portfolio Dissertation Proposal is made up of seven different compositions for data-driven instruments (DDI), this document that explains each of the compositions in some detail, and a video recording of each work. Examination of these elements provides an in-depth view of the interfaces, composition techniques, sound design, software, performance practices, and visual output in the works in this dissertation. The videos included in this dissertation are of slightly varying quality. This inconsistency in quality exists because these videos were created over the course of three years. Three years is a long span of time in terms of technological capabilities.

Each of the seven compositions is represented in this dissertation by both writing and video. This video documentation illustrates the sound world and the performance techniques used to create that sound world. In an effort to be as clear as possible in my writing, I have chosen to use specific terms to have specific meanings. For instance, the word *sample* has several meanings depending on the context. In spoken conversations, magic is easier to use this word and describe the context. However, I have chosen to use *sample* when I am discussing the smallest measurement taken by an analog to digital converter (ADC). When I am writing about playing a short sample of a larger audio file, I use the term *audio file*. I intentionally use specific language in this dissertation for clarity.

Each of the performances in this dissertation features a unique data-driven instrument. A data-driven instrument is made up of three basic modules: an interface module, a software mapping module, and a sound-production module made from

hardware or software. Each data-driven instrument in this dissertation follows this same modular paradigm, but with very unique outcomes. Data-driven instruments are explained in detail in the next chapter.

The larger theme in each of the referenced works is magic. The online dictionary, defines magic as, “an extraordinary power or influence seemingly from a supernatural source.”<sup>1</sup> Each of the works in this dissertation plays with this theme. The power of magic cannot be understated. For many of us, magic is the distance between our abilities and the limits of our imagination. Magic is the unknown, sometimes the unknowable. Magic is in the witnessing of events that exist just beyond our own abilities that inspire us most. The theme of magic is present in all seven compositions in some appreciable way. In my experience, watching skilled performers interact with their instruments at the highest level has been the most fascinating, inspiring, and magical events for me to witness. The culmination of these experiences inspires me in all of my creative work.

I very intentionally defined what magic means to me concerning live performance. However, my intent here is not to divert the focus of my research to magic. Magic is more of a muse to me. Magic is only one type of inspiration that I find useful when composing and performing. Magic helps to point to the direction for me to go, but magic is never the end, rather the means. I have attempted to point out the connection to magic in each of my seven works. However, an audience must find their own connections between magic and my performances. Magic must be observed and defined by an audience — not the composer.

---

<sup>1</sup> Merriam Webster. “Magic | Definition of Magic by Merriam-Webster.” Accessed May 1, 2017. [https://www.merriam-webster.com/dictionary/magic?utm\\_campaign=sd&utm\\_medium=serp&utm\\_source=jsonld](https://www.merriam-webster.com/dictionary/magic?utm_campaign=sd&utm_medium=serp&utm_source=jsonld).

In addition to my work in this dissertation, I am also a classically trained percussionist. One aspect of drumming I enjoy is that I can arrange or organize the pieces of my setup (interface) in almost any way I can imagine. Composing and performing with new interfaces that are of my own creation is very fulfilling in a similar way. I can build anything I can dream, with some exceptions. I enjoy building interfaces that do not immediately satisfy expectations from an audience perspective. Interrupted expectations help to create interest as I *teach* an audience about the interface as I perform. A grand piano, a flute, a saxophone, and a clarinet all have some fixed number of keys. Every audience member has *some* expectation for the performative actions employed with these instruments. However, interfaces for data-driven instruments can be made up of any type or combination of keys (buttons) or other sensors. A data-driven instrument can also change its own functionality at any point during a performance. Controls that behave one way in the beginning of a work can behave another way at the end. This functionality is unique to data-driven instruments.

## **SEVEN COMPOSITIONS**

*Clara Voce* (Chapter III.1) allows me to take full advantage of my height. Performing this work requires me to control aspects of the sound and visuals by waving my arms in every possible direction. These movements are interpreted into a data stream by the Gametrak controller. This data is then distributed to multiple software environments to create both the sonic and visual domains. Though my performative actions are exaggerated, they contribute to the overall narrative of this composition. All

of my performative actions are translated to the sound design and visual output. For me, the magical moments happen in every section where I appear to *pull* the audio from the controller. This act of triggering the onset of sound by pulling a string has implications to many stories, metaphors, and lore. However, the resultant sound output of these actions provides a closer view into the hidden sounds in speech. Because the audio files are spoken text and these files are playing from a granular synthesizer, I create new harmonies and dissonances that are in the original audio, but overlooked (or overheard) by humans in real time.

In *Ouriana's Meditation* (Chapter III.2), I use a small, frosted, acrylic box. There are not apparent controls on this interface. The controls are, however, embedded inside the box. This embodiment conceals the technology in a way that allows the audience to inject their sense of a magical narrative. The physical movements in this work include turning, angling, and shaking the box. These movements generate the data that creates the sound output and are visually symmetrical to the physical input.

The acrylic box in *Ouriana's Meditation* is meant to resemble a MusicBox. This association begins at the onset of sound when I open the lid to trigger the sound of a tiny hand-cranked MusicBox generator. Closing the lid also triggers a sound. Throughout the performance I further imply the MusicBox narrative through my performative actions by aligning my shaking actions with the onset of bell sounds. These actions imply that the box could have bells inside that are sounding from my physical actions. The onset of sound is actually triggered by me pressing a small button, but pressing a small button is

far less interesting than my choreographed movements because my movements are intended to support the underlying narrative of the muse.

In the middle section of *Ouriana's Meditation* the MusicBox is used to control the mix of four sounds. In this section, my movements are measured in deviations in the change of angle of the interface. When I hold the interface level to the ground, all four sounds are heard. As I angle the interface in any direction, the result is evident in the mix of the four audio files in real time. This change in function of my interface is another example of the flexibility afforded by using a data-driven instrument.

*Channeling Rachmaninoff* (Chapter III.3) is a unique composition in this portfolio as it employs light to re-interpret a physical medium — piano rolls. And while the primary data-producing input from me is turning a take-up roll to move the paper over a row of multi-colored lights (light bar), the sonic result is nothing like the source for this composition, *Prelude #1 in C# minor, Op. 3, No. 2* by Sergei Rachmaninoff. The interface for this work is a custom made DDI, the Lux Codex, that uses computer vision and multi-colored light to re-interpret piano rolls for electroacoustic performance. I designed the Lux Codex and accompanying software for this dissertation.

In action, the Lux Codex behaves more like a player piano with manual control. In a player piano, a paper roll is moved over a drum that senses the small holes in the paper and translates that data to pitch and duration. In the Lux Codex, the light bar replaces the moving drum. The paper piano roll moves over the multi-colored light, and the variations in brightness of each color are measured to create data input to control sound and visuals. The various elements of light detection call upon the theme of magic

in this composition as the variations in light are caused by the particular paper roll used in a given performance. The idea that computer vision can *see* through paper and record the light variations as data that is used to create music is nothing short of magic.

*Curves & Lines* (Chapter III.4) is a departure from the other works as it does not use a laptop or desktop computer. However, it does require an iPhone, iPad, and some specialty hardware. I created several drawings of curves, lines, dots, and shapes that are interpreted by software on the iPhone, Phonopaper, using computer vision to assign frequency, duration, and amplitude to the drawings. In all cases, the Y-axis of the drawings are mapped to pitch, while the X-axis of the drawings are mapped to duration. This mobile application allows me to capture a line drawing as an audio waveform in real time. This process is the primary sound generator in this work. The resultant audio is sent to an iPad for manipulation and routing. Deadalus, an application on the iPad, controls timbre and spatiality for the input sound from the iPhone. The visual output for this work is the video output of the iPhone. The magic of this work lies in the transformation of what is seen, into what is heard. Since the audio is generated by line drawings, it is difficult to control all aspects of sound. This process results in sound output from the iPhone that is sometimes equal parts harmonic, rhythmic, and noise. Because I gave up control over the sound at the first stage, I used subtractive synthesis techniques to sculpt the final sound. Each of the line drawings converted to audio are filtered and recorded in an application on the iPad, LoopyHD. LoopyHD is a twelve track looping software that allows me to record, play, reverse, pan, and mix each of the twelve tracks in real time.

Similar to *Clara Voce*, *Euphony* (Chapter III.5) contains a rich sound world derived primarily from audio files of speech. The text spoken in this work comes from several sources. While the majority of the text is carefully extracted from *Winterreise* by Franz Schubert, there are also selections from Sazaria Quarles and Margaret Wise Brown.<sup>2 3</sup> Syntactically, the thread in all of these texts connects the narrative of the Wanderer in *Winterreise* to the sound of birds and a warm summer night. This work is an example of creative recomposition. I used existing material to realize a new outcome. My composition techniques included the mixing of small pieces of text to find the combinations that was most pleasing to me. I use text painting in this performance to further emphasize specific words and sounds. I am able to change the pitch of a sound without changing its timeindex. This functionality allows me to emphasize certain words or phrases by modulating the sound through a series of filters and effects, like delay and reverb. This composition features live resynthesis of spoken text. There is magic in the performance of this text. The combination of interface and sound design are well-suited to one another. The sensor resolution of the Wacom tablet allows for an imaginative performance of the sound material. My performative actions are physically small in relation to other works in this dissertation. And because all of the sound production is a result of these smaller actions, I am able to trigger multiple sounds and provide multiple forms of ongoing control easily with the pen and tablet, simultaneously.

---

<sup>2</sup> Sazaria Quarles. "Haiku About Summer Nights, Summer Midnight, Summer Poem." Last modified 2015. Accessed November 1, 2017. <https://www.familyfriendpoems.com/poem/summer-midnight>.

<sup>3</sup> Margaret Wise-Brown. (1997). *Pussy Willow*. Random House Children's Books. Retrieved from [https://books.google.com/books/about/Pussy\\_Willow.html?id=eoMz96LCR4EC](https://books.google.com/books/about/Pussy_Willow.html?id=eoMz96LCR4EC)

*The Insistence of Sameness (IS)* (Chapter III.6) is my interpretation of one small aspect of my experience with knowing someone who lives with autism spectrum disorder (ASD) and how it affects the mind.<sup>4</sup> While I openly admit to being inspired by a particular person, this work is meant to present a series of movements and sounds that will inspire an audience to consider the larger narrative and raise awareness for ASD. For me, the magic in this composition is the translation of movement to sound. My physical movements are a theatrical attempt at reproducing body movements sometimes described as *stims*, short for stimulant. “Stimming is a repetitive body movement that self-stimulates one or more senses in a regulated manner. Stimming is known in psychiatry as a “stereotypy,” a continuous movement.”<sup>5</sup> I attempt to combine these movements with sound design that represent the struggle of focus. For this, there are three main types of sounds: rich, chordal pads, sharp, sparkly noise permutations, and transitional sounds.

I use various combinations of sounds to create a sonic world that is clearly fractured by noise. This sonic world is a direct inference to a small selection of my experience with ASD. I do not intend to imply that noise is bad or that people who live with ASD have less value than others. Rather, I use the sonic relationship between soothing sounds and noise sounds to illustrate the duality of silence and noise that exists in us all. My performative actions and their resultant sounds are meant to stimulate positive discourse around ASD. Unfortunately, stereotypes about ASD persist and it is my

---

<sup>4</sup> Autism Speaks. “Neurobiological Mechanisms of Insistence on Sameness in Autism | Autism Speaks.” Autism Speaks Website. Accessed May 1, 2017. <https://www.autismspeaks.org/science/grants/neurobiological-mechanisms-insistence-sameness-autism>.

<sup>5</sup> Autism Wiki. “Stimming | Autism Wiki | FANDOM Powered by Wikia.” Autism Wiki. Accessed April 1, 2017. <http://autism.wikia.com/wiki/Stimming>.

intention to help dispel the stigma and bias that surrounds ASD. I am not on the autism spectrum, and in no way claim to understand ASD better than anyone else. Instead, I am hoping to raise awareness through my performance of this work.

*Sub-Rosa* (Chapter III.7) connects the mystical powers of crystals, gems, and geodes to the Fibonacci sequence. The initial section of the sequence is used as frequency data in a synthesis algorithm in Kyma. The movement of each controller is mapped to different aspects of sound. Each of the smaller interfaces in this composition have very limited, yet specific capabilities. Together, these interfaces provide both discrete and interdependent data streams. The various combinations of the interfaces provide parametric control over the sound during performance, while providing a visual context through their physical forms and illumination.

The magic element in this work surrounds the connection between the movement of the Geodesic Electronic Movement Sensors (GEMS) and the sounds they produce. Each controller provides a single element of control, but together they form a more complete instrument. These interfaces are controlled on a thin vinyl mat, similar to placemat. This mat has a design printed on it for each interface and also frames the interfaces nicely during performance. The shapes printed on the mat are analogous to the physical movements of each interface. The designs on the mat combined with glowing interfaces, give the appearance that I am playing some sort of game. In fact, that is not far from the truth. The mat informs my interactions with each interface during performance. And because I simultaneously broadcast a close-up view of the mat during performance, the mat also helps to provide a context for my movements to an audience.

*Sub-Rosa* is another example of a data-driven instrument changing its functionality during performance. Each interface functions slightly differently in every section of this work. This changing of function is an advantage to performing with data-driven instruments.

## CHAPTER II

### II. CONCEPTUAL AND THEORETICAL FRAMEWORKS

To perform with an acoustic instrument, a performer exerts some amount of physical energy to generate sound. This physical energy is generated from moving air through a trumpet or bowing a violin string, or even by hitting a drum. In all of these instruments, there is some type of interface that a performer must interact with in order to generate sound. Once the performer fills the trumpet with air he presses different keys to alter the airflow. Different combinations of keys change the routing of airflow. This change results in lower and higher pitches from the instrument. A data-driven instrument works within the same paradigm as an acoustic instrument — but data replaces energy and software is used for most other functions.

Interfaces are typically physical objects, embedded with sensors that allow a performer to generate data through interaction. However, the interface is not a complete instrument. The larger classification is data-driven instrument (DDI). A DDI is a combination of three primary parts: interface, a software mapping layer, and a sound (or visual) producing algorithm. The interface can be any type of device that generates and outputs a data stream (usually numbers) as a result of performer interactions. The software mapping layer is essential as it is the translation from the data generated from sensors to the data needed for sound production. The mapping layer applies simple math functions to incoming data streams to translate to a useable range and type of data

required by sound or visual production. Finally, the sound (or visual) producing algorithm in a DDI can be software, hardware, or a combination of both.

A data-driven instrument typically includes a nontraditional interface that is equipped with sensors and often embodied in other objects. These interfaces allow a performer to physically interact with its sensors in various ways. The manipulation of sensors replaces the physical energy exerted into a traditional instrument. In a data-driven instrument, data generation becomes the surrogate for physical exertion.

Another aspect of a data-driven instrument is a software mapping layer. This software layer exists after the interface and before the sound and visual producing software. Data mapping is a general term that is used to describe various processes for altering data. Data mapping is an essential aspect of a data-driven instrument because data mapping provides a means to translate raw data from the interface to a range and format that is suitable for its destination(s).

## **DATA MAPPING**

Data mapping is where the magic happens in every one of my performances. Data mapping allows me to match performative actions to resultant sound and visual outcomes. Data mapping allows me to generate loud sounds from small actions. Data mapping also allows me to decide *how* I want to control a sound. For instance, In chapter III.1 *Clara Voce*, I map my right arm movements with the controller to the drawing location of a particular section of the visual dimension. I chose to use a movement where I begin with my right arm close to my body and move it away from my body for this

aspect of control. This movement results in the drawing location to move from the outside edges to the center. I could have easily mapped this control in the inverse (movement away from my body to closer to my body). However, at that particular moment in this performance I am also performing movements with my left arm. Data mapping provided a means to reverse this control to better fit with my other performative actions.

Data mapping is necessary to provide each sound producing (and visual producing) algorithm with the appropriate range of data (numbers) to efficiently produce the desired outcome. There are many data mapping strategies employed in the compositions in this dissertation. For Instance, in *Channeling Rachmaninoff*, the primary data is generated from RGB color values, which are measured as an integer between 0-255 (256 total values). This method of data mapping is called scaling. Scaling is the process of interpolating an input range to new output range. For example, incoming data from Wiimotes (via OSCulator) have a native range of 0-1. However, controlling devices in Ableton Live requires the data to be in a range of 0-127. Range scaling can easily achieve this goal by applying simple math to generate a useable output value. Range scaling is very common and is used throughout every work in this dissertation.

Range splitting is the process of extracting a portion of the incoming data stream. This technique is sometimes necessary depending on the physical affordances of an interface. In *Clara Voce*, I use range splitting to extract a portion of the Gametrak's Z-axis to control MIDI modules in Ableton Live. The full range of the Gametrak's Z-axis is 0-4095 (4096 total values). In this example, I use range splitting to extract (or split) the

full range between 100-500. This range of 400 values is scaled to 0-127 for use in MIDI modules. Range splitting is needed in order to facilitate my desired physical position during performance.

Compression and expansion of data are common methods of data mapping. Compression involves taking a large data range and reducing it to a smaller range for use in a sound or visual producing algorithm. Expansion of data is taking a smaller range of data, usually numbers, and generating a larger output range of values. Both of these techniques were used in every composition. The use of these two techniques is pervasive in almost every instance where changing the data stream is necessary. The previous example of *Clara Voce* is an example of both range splitting and data expansion. Often, more than one approach to modifying data is required in order to produce the desired output.

Data routing is the process of taking an input data stream and sending it to a discreet location (or locations). Data routing is different than data mapping in that routing is only concerned with forwarding the data to a particular location, routing does not change or alter the data stream. Data routing is essential to every DDI. One of the advantages to using a DDI is that the instrument can have different capabilities at any time during a performance. A DDI is mutable and can change based on performer or compositional needs.

Data routing is essential in every work in this dissertation. In *Insistence of Sameness (IS)*, I generate multiple, unique sounds using the same button on the Wiimote. Data routing is the process that makes this functionality possible. In this case, the button

sends the same ON/OFF data, but the destination of its data is changed depending on the current section of the composition. This method of handling data from interfaces exists in all of the compositions in this dissertation.

## **INTERFACES**

An important aspect to every DDI is modularity. Each of my interfaces is a combination of smaller modules (sensors, buttons, etc.). Modularity is important as it allows me to build in smaller sections, as well as to experiment with module selection and order. Some modules include objects with sensors that produce data. Other modules are sound-producing modules that receive data and use it to generate sound or visuals. One of the works in this dissertation uses only mobile devices as the method of input, mapping, and sound production, while the remaining six works all use a computer in some way.

While the interfaces presented in this dissertation are all very different from one another, they all have several things in common. My interfaces are all physical objects that need human/performer interaction in order to generate data. These interfaces contain at least one sensor that captures the physical interactions of a performer and outputs analogous data. Each of the interfaces have differing methods of feedback, observable by an audience and the performer. Most importantly, each of the interfaces listed here are mutable. They can change over time — and they often do. Their functionality can adapt to any purpose at any time. The interfaces in this portfolio offer varied ways for me to generate data that can be transformed into sonic and visual output.

Physical sensors include buttons, accelerometers, computer vision, conductive pens, mobile devices and applications, and video game controllers. These physical sensors offer many possibilities for interaction. A button can be a toggle — has an on and off position — or a momentary — only active while being held down. A Force-Sensitive Resistor (FSR) is a sensor that measures the amount of force (pressure) exerted onto its surface. In *Ourliana's Meditation* (Chapter III.2), I use a smaller version of an FSR sensor as a button. I refer to this sensor in my writing as a micro-FSR. The micro-FSR functions in the same way as a typical FSR, although it is much smaller (see Figure 1 for a comparison in size between a normal FSR and the micro-FSR used in this work). An accelerometer is a familiar sensor to many people as it is inside every smartphone. This sensor measures the change in angle of tilt in several directions: X, Y, and Z. In addition to accelerometers, 9 degrees of freedom (9DoF) sensors allow the measurement of change in the angle similar to the accelerometer, but also measures orientation and direction of travel through the use of a compass. The 9DoF sensor can measure a change of movement on nine axes. I use computer vision in two different works in this dissertation. In *Channeling Rachmaninoff* (Chapter III.3) I use a small webcam to measure the change of colored lights through a piano roll. In *Curves & Lines* (chapter III.4) I use the camera from an iPhone to capture line drawings that are translated into sound. Conductive pen tablets allow a performer to freely draw in a predefined space to generate multiple data streams, simultaneously. Video game controllers have their own social baggage attached. While many people who play video games immediately understand

these controllers, non-gamers do not have the same understanding or expectations that gamers have regarding functionality or purpose.



**Figure 1.** Size comparison of FSR sensors.

Sensor Products Inc. “FSR Sensors Size Comparison.”

Accessed March 19, 2017. <http://www.sensorprod.com/campaign/fsr-sensors/index.php?mcode=gc-force>.

## COMMUNICATIONS

Data communications is defined by Random House as “electronic transmission of information that has been encoded for storage and processing by computers.”<sup>6</sup> Sending and receiving data is essential to every work in this dissertation. Since every data-driven instrument is a series of modules, it is important that these modules communicate with one another. I use several modes of communication in these compositions. MIDI is a technical standard that describes a protocol for digital instruments and computers to communicate with other digital instruments and computers. I use MIDI data in *Channeling Rachmaninoff* (Chapter III.3) to change the visual output for each section of the composition. I also use MIDI in *Curves & Lines* (Chapter III.4) from the foot controller to all applications running on the iPad to control various functions. *Insistence of Sameness (IS)* (Chapter III.6) relies heavily on MIDI communication to both generate

---

<sup>6</sup> Dictionary.com. “Data Communication | Define Data Communication at Dictionary.com.” Accessed March 19, 2018. <http://www.dictionary.com/browse/data-communication>.

and control sound. *Sub-Rosa* (Chapter III.7) also relies on MIDI communication between my computer and my sound generating hardware.

Open Sound Control (OSC) is another form of communication used in my compositions. OSC is based on the UDP protocol though does not feature a callback function. No callback function means that data is sent, but no confirmation of that data being received will be sent back to the sender. This one-way communication is well suited for musical performance situations because it should always be obvious that data is or is not being transmitted. OSC uses a unique addressing system. A typical OSC message from a fader mapped to a real-time control could look like this: /mixer/fader1/XXX. This message is sent to a receiver on the same network, that has an address of fader1 inside of an object titled mixer. The ‘XXX’ refers to the actual data, or numbers, being transmitted.

In serial communication, data is transferred one bit at a time, one after another.<sup>7</sup> Serial data can be any combination of letters or numbers. However, because serial communications is 8-bit resolution, the highest number that a serial packet can represent is 255. Sending larger values over serial is possible if that value can be taken apart and reassembled on the other side of the serial bus. This method was not needed in my work. I rely on serial data in two of my compositions — *Ouriana’s Meditation* (Chapter III.2), the interface is communicating over a USB cable, using the serial protocol. However, in *Sub-Rosa* (Chapter III.7), the interfaces are sending serial data wirelessly to my

---

<sup>7</sup> Jimbo. “Serial {Communication} - Learn.sparkfun.com,” n.d. Accessed March 19, 2018. <https://learn.sparkfun.com/tutorials/serial-communication>.

computer. In both works, raw sensor data is scaled to a smaller representation before transmission.

I use the Human Interface Device (HID) specification in *Clara Voce* (chapter III. 1) for communication between the interface and my computer. HID is another specification for mitigating the transfer of data over USB. From the specification from [usb.org](http://usb.org), “The HID class consists primarily of devices that are used by humans to control the operation of computer systems. Typical examples of HID class devices include: ... standard mouse devices, trackballs, joysticks, games or simulation devices—for example: data gloves, throttles, steering wheels, and rudder pedals.”<sup>8</sup> HID is unique in that it allows a user to transmit higher values over USB compared to serial because HID utilizes a 12-bit resolution that can represent values up to 4096.

## **HARDWARE**

I use various types of hardware in this dissertation from well-known devices like an Apple MacBook Pro, iPhone, and iPad. I also use several microcomputers. Microcomputers are very small in physical size, but have more than enough computing power to process the input from various sensors and output as data for use in any DDI modules. One of the microcomputers I use often is the Adafruit Trinket. The Trinket is a small microcomputer that offers analog and digital input/output. The Trinket is the heart of each of the GEMs interfaces in chapter III.7. The Adafruit Trinket is also the computer inside the MusicBox in *Ourliana's Meditation*. The Adafruit Trinket is versatile in that it

---

<sup>8</sup> USB.org. “Device Class Definition for Human Interface Devices (HID) Universal Serial Bus (USB)” (2001). Accessed March 19, 2018. [http://www.usb.org/developers/hidpage/HID1\\_11.pdf](http://www.usb.org/developers/hidpage/HID1_11.pdf).

can connect as a Serial device over a USB cable or wirelessly, with additional hardware. In every case, the Adafruit Trinket processes the input of various sensors, scales the input sensor range to the desired output range, and communicates that data to software on my laptop.

I use video game controllers in two compositions, *Clara Voce* and *Insistence of Sameness(IS)*. In *Clara Voce*, I use a Gametrak controller that was intended for a golf themed video game. *Insistence Sameness (IS)* features two Nintendo Wii Remotes (Wiimotes). These controllers were the primary controllers for the Nintendo Wii gaming console. The Wiimotes connect to my laptop via bluetooth and a helper software, OSCulator, manages the connection and the transfer of data from the Wiimote to the computer. OSCulator interprets the raw data from the Wiimotes and scales every data stream to a range between 0-1, as a decimal. OSCulator communicates with other software on my laptop via OSC. OSC is a standard for sending and receiving data over a network connection. In all cases where OSC is used to send data, Max/MSP is involved in some way. Max/MSP provides easily accessible tools for sending and receiving OSC messages.

## SOFTWARE

I use several software applications in this dissertation, including Ableton Live, Kyma, Max/MSP/Jitter, OSCulator, and Processing.<sup>9 10 11 12</sup> The audio software used is: Ableton Live, Max/MSP, Kyma, Phonopaper (iOS), Dedalus (iOS), and LoopyHD (iOS). I also use software to generate visual content during performance. The visual software used is: Jitter and Processing. All software operates on my laptop except the titles indicated as iOS software. These applications operate on an iPhone or an iPad.

Software applications that are not used during a performance, but were used in the creation of essential aspects of this dissertation include: Adobe Illustrator, Adobe Photoshop, Apple Motion, Apple Final Cut Pro, Apple Logic Pro X, iZotope's Rx6 Advanced Waveform Editor, and Arduino IDE. Each of these applications was used in some way to either design an aspect of this dissertation or to create content for real-time performance.

Max/MSP is a visual programming language where objects are connected together in various ways to create a desired output. Often, that desired outcome is audio. However, Max/MSP is very good at manipulating data in seemingly endless ways. Max/MSP connects to the GEMs in the final composition (Chapter III.7) Max/MSP is also the primary software mapping layer for five of the seven compositions in this dissertation.

---

<sup>9</sup> "Music Production with Live and Push | Ableton." Last modified 2018. Accessed March 19, 2018. <https://www.ableton.com/>.

<sup>10</sup> Cycling74. "Cycling '74." Last modified 2018. Accessed March 19, 2018. <https://cycling74.com/>.

<sup>11</sup> OSCulator.net "Home | OSCulator." Last modified 2018. Accessed March 19, 2018. <https://osculator.net/>.

<sup>12</sup> "Processing.org." Processing.org. Accessed June 03, 2017. <https://processing.org/>.

The compositions that do not use Max/MSP are *Curves & Lines* and *Euphony*. *Curves & Lines* is performed without the use of my laptop computer. In *Euphony*, the Wacom tablet connects directly to Kyma and no intermediary software is needed. Kyma offers many ways to map, scale, and route data from the Wacom tablet.

Kyma is the primary sound synthesis engine used in this dissertation. Kyma is used in four of the seven compositions. Kyma, like Max/MSP, is a visual programming language for sound design. In Kyma, like Max/MSP, objects are connected with wires. However, the resemblance between Max/MSP and Kyma stops there. Kyma is hardware and software dedicated to sound production. Kyma also provides a language for mapping, routing, scaling and shaping data in real time. In Kyma, Capytalk is the language for this purpose. A combination of connecting objects and properly formatted Capytalk expressions make Kyma a very powerful sound design language.

Ableton Live is a Digital Audio Workstation (DAW) software that provides a powerful, yet unique set of tools for sound design. In addition to typical features of a DAW, Ableton Live exposes every control for every effect, filter, or EQ to MIDI control. Ableton Live has a unique capability that allows a Max/MSP file to load as an effect in an audio channel. This level of functionality allows a composer to create and use custom made instruments and effects at a fairly low computational cost.

Adobe Illustrator, Adobe Photoshop, Apple Motion, and Apple Final Cut Pro were used to generate either design documents for the physical construction of an interface or performance surface, or visual assets used during performance. Apple Logic Pro X was used to create the explosion sound at the start of *Ouriانا's Meditation*. iZotope's Rx6

Advanced Waveform Editor is my primary waveform editor and I use it to prepare audio files for performance. The Arduino IDE is an environment that allows developers to write code and upload to an Arduino. This software is used during the building of an interface.

In addition to the software titles listed above, there are also a few mobile applications that operate on Apple mobile devices. Phonopaper captures photos of drawings to create audio. Phonopaper works by associating high contrast features in the drawings to pitch and duration. Phonopaper allows control over the playback rate, amplitude, and playback position. Dedalus is an iOS application that operates on an iPad. Dedalus is primarily an audio effects application that adds delay, overdrive, distortion, and frequency shifting. Dedalus also provides comprehensive mapping and routing features. Incoming MIDI data from an external foot switch (SoftStep 2) is routed to Dedalus to control spatialization and amplitude. The SoftStep 2 also sends MIDI data to LoopyHD for control over basic functions. The connection between the SoftStep 2 and all iPad applications is the iTrackDock (iTrack) iOS hardware interface. This interface provides audio input and output to the iPad as well as USB for MIDI controllers like the SoftStep 2.

## **SOUND SYNTHESIS**

The compositions in this portfolio share sound synthesis techniques. There are examples of additive and subtractive synthesis in several of these works. There are many examples of granular synthesis and sample manipulation. Additive synthesis is combining multiple waveforms with varying frequencies, amplitudes, and phases. Subtractive

synthesis is achieved through the use of filters to remove unwanted characteristics from a sound source. Both of these techniques can be found in *Curves and Lines*, as well as *Insistence of Sameness*.

Granular synthesis is a method of breaking down an input signal into smaller pieces called grains. These grains are typically a length between 1-50 ms, where each grain can be manipulated uniquely from one another. Granular synthesis can be found in *Clara Voce* as the primary sound-producing algorithm. In *Euphony*, you can hear granular synthesis in the recorded, manipulated sounds layered under the main voice. Granular synthesis plays a major role during the middle section of *Ouriana's Meditation* in the multiple roti-tom sounds, sirens, and low sub bass sounds as are all being crossfaded while being granulated with varied parameters.

Audio file manipulation is present in every composition. Sample manipulation includes recording audio to a digital medium and using various software techniques to alter the original content. Most notably, *Insistence of Sameness* uses multiple sample manipulation techniques including granulation, audio file playback, audio file stretching, and multisample playback. Multisample playback is a technique where a sound producing mechanism cycles through a folder of audio files in some order. This technique can be heard in *Insistence of Sameness (IS)* and *Sub-Rosa*.

Kyma offers multiple sound synthesis opportunities that are simply not available in other software.<sup>13</sup> First is the Time Alignment Utility (TAU) and resynthesis tool where “temporally coordinated amplitude, frequency, formant, and bandwidth envelopes can be

---

<sup>13</sup> Kyma. Accessed June 03, 2017. <http://kyma.symbolicsound.com/>.

mixed and morphed between and among to yield stunning timbral nuance.”<sup>14</sup> Resynthesis is the process of breaking down a complex waveform into its individual components — sine waves, each reproduced by a series of oscillators to generate sound with greater control. The TAU can be heard throughout *Euphony* as it is the primary synthesis technique for the primary sound generation. The TAU allows me to independently manipulate frequency and pitch of an audio file during performance.

The SampleCloud prototype in Kyma granulates an audio file, but its particular combination of parameters provides a unique approach to granular synthesis. The “SampleCloud outputs a granulated sound cloud based on a particular audio file while providing controls to determine the grain shape, grain duration, cloud density, spatialization, amplitude, and frequency.”<sup>15</sup> Examples of the SampleCloud object can be heard in *Channeling Rachmaninoff* and *Euphony*.

I also use the Kyma Replicator prototype. This prototype plays multiple copies of an audio file, where each copy can be modulated independently in terms of pitch, spatialization, amplitude, and timbre in real time. The Replicator can be heard in the many, jingly bell sounds in the early minutes of *Ouriana’s Meditation*.

---

<sup>14</sup> Jeffrey Stolet, *Kyma and the SumOfSines Disco Club*. Edited by Adam Shanley. Morrisville: Lulu Press, 2012.

<sup>15</sup> Stolet. Ibid.

# CHAPTER III

## III. DISCUSSIONS OF INDIVIDUAL PORTFOLIO COMPOSITIONS

### III.1 CLARA VOCE

A multichannel, real-time, interactive composition for Gametrak, iPad Mini 4, Max/MSP, Ableton Live, and Processing.

#### INTRODUCTION

*Clara Voce* is Latin for “secret voice.” *Clara Voce* is a fitting title for this work as the primary audio content is derived from recordings of spoken text. The text is a collection of statements, spoken and whispered in French, regarding the colors green and blue (Appendix A). I chose French because I love the sensuous sound of the language. This work begins with a performance of a set of phrases and evolves into multiple voice sounds interacting with one another in a conversation. Using performative actions described later in this chapter, I am able to perform these conversations in a musical way. In one case, I find new words that are not part of the original text. At 2:14 in the accompanying video of this work you can hear the voice speaking the phrase, “On and on and on...” This new phrase was created from the same audio file of French text by manipulating the timeindex of the audio file. Finding this feature of reorganizing texts is something I consider a happy accident. I could never have intended this outcome, but I reproduce it musically for my benefit.

My connection to magic in this work is in the data mapping. My ability to map larger, exaggerated movements to sound and visual parameters allows me to realize this composition.

The live performance of this work employs an enormous performative space as it relies on the width of my reach. Standing at six feet six inches tall, I can reach 100 inches vertically, and have a total reach of 80 inches. The Gametrak contains two simple joysticks that move around a 180° sphere, these joysticks have strings attached that allow a performer to pull them from the interface. Because the Gametrak (interface) has a Z-axis (vertical) range of 10 feet, I can leverage my height to make the delivery of this performance more dramatic.

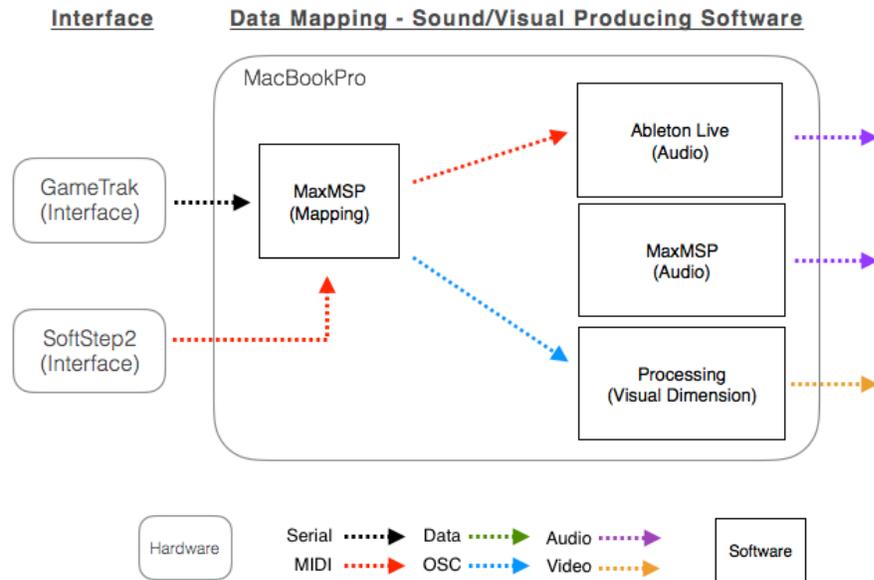
Primarily, I use granular synthesis in this work because this composition requires control over aspects like playback rate, pitch — separated from time, and calculated randomness. I use the Max for Live device Granulator II in Ableton Live.<sup>16</sup> This device offers the exact toolset I need. I use the Gametrak to control the audio manipulations and trigger different pitches. The visual context in this work is created in Processing. The visual elements used in this work are primarily lines and dots of varying sizes, colors, and transparencies. These elements are presented in real time and controlled by my manipulation of the Gametrak controller. The visual portion of this work, like the audio portion of this work, is performed in real time.

---

<sup>16</sup> Robert (Monolake) Henke. “Granulator II | Ableton.” Accessed March 19, 2018. <https://www.ableton.com/en/packs/granulator-ii/>.

## INTERFACE, HARDWARE AND SOFTWARE

The data-driven instrument for this composition consists of a Gametrak video game controller, a SoftStep 2 foot controller, Max/MSP, Ableton Live, and Processing. All software is operating on an Apple MacBook Pro (Figure 2).



**Figure 2.** Diagram of data-driven instrument for III.1 *Clara Voce*.

The physical interface for this composition is a Gametrak video game controller created by In2Games. The Gametrak controller consists of two joysticks that are each three-dimensional (3D) joysticks. Each joystick generates X, Y, and Z data (when the string is pulled out of the interface), so each joystick control works like three faders that are interdependent of one another. Each joystick can be manipulated on its X, Y, and Z axes. Each axis outputs an integer data that ranges from 0 - 4095. The uniqueness of the Gametrak lies in the three-dimensional joystick. The handle on both 3D faders is connected to a string-pot that pulls out of the box. A string-pot is simply a potentiometer

that is turned by pulling a long, spring-wound spool of string. This pulling of the string-pot turns the potentiometer which outputs data. According to In2Games, the mechanisms can determine position, “to an accuracy of 1 millimeter anywhere within a 3 meter cube around the unit, with no processor overhead or time delay.”<sup>17</sup> This level of responsiveness and precision was essential for the controller’s original purpose — to control a virtual golf swing! This level of precision and responsiveness is also well suited for use as a real-time interface in a musical performance.

I use my height as an advantage in this work by utilizing as much of the Gametrak’s three-meter cube as possible. This translation of my larger movements, I believe, are more compelling than much smaller movements using the same interface. Larger performative actions are not only easier for an audience to perceive, they also add to the level of drama and difficulty for the performer.

In addition to the Gametrak, I also use a SoftStep 2 MIDI foot controller.<sup>18</sup> This foot controller is connected to my laptop via USB and its input is received in Max/MSP for mapping and routing to Ableton Live, Processing, and within Max/MSP. The SoftStep 2 houses 10 small foot switches that send data when pressed. In this work, I focus primarily on shaping ongoing sound. As such, all voice sounds are triggered by an algorithm in Max/MSP that monitors the incoming Z-axis values and triggers MIDI pitches based on a predefined subdivision. The primary means of subdividing this data

---

<sup>17</sup>Adrian Freed, et al. *Musical Applications and Design Techniques for the Gametrak Tethered Spatial Position Controller*. Technical paper. Proceedings of the SMC 2009 - 6th Sound and Music Computing Conference, 23-25 July 2009, Porto - Portugal.

<sup>18</sup> Keith McMillen. “SoftStep 2 USB MIDI Foot Switch & Expression Pedal Controller | Keith McMillen Instruments.” Last modified 2018. Accessed March 19, 2018. <https://www.keithmcmillen.com/products/softstep/>.

stream is to divide the incoming number by 15, and monitor the remainder (modulo).

This method of subdividing a data stream provides the necessary outcome with very little CPU overhead.

The software for this composition includes Max/MSP, Ableton Live, and Processing. Max/MSP is the first point of contact for the Gametrak. Max/MSP receives the data from the Gametrak using the Human Interface Device (HID) protocol, outputs multiple mappings of the incoming data depending on the destination, and routes all data to either Ableton Live as MIDI controller data (0-127), or Processing as floating-point values (0-1). Ableton receives the MIDI controller data and routes it to the specified location (e.g., amplitude, pitch, spatialization, etc.). Max/MSP sends the floating-point values to Processing via the OSC protocol. Max/MSP is also responsible for receiving input from the SoftStep 2 controller and routing to the appropriate destinations.

Not all of my performative actions utilize every inch of my physical reach. Some are constrained to much smaller areas depending on the requirements of the composition. For example, I both trigger the onset of the sound and provide ongoing control with the Z axis of the left joystick of the Gametrak. As I pull the string, the sound progresses through its timeindex. However, the onset is only triggered after the Z-axis is pulled out of the controller several inches. The full range of the sound is then mapped to a physical space of roughly one meter. So pulling the string several inches triggers the sound and continuing to pull up to one meter will generate ongoing control. This result was achieved by extracting a smaller range of values from the raw data and mapping that range to an appropriate (smaller) number based on its destination. This entire control and

mapping mechanism is designed to split the raw data stream from 100-500. The Z-axis outputs 0 when at rest, so 100 is several inches above the resting point. 500 lands just above my waist. This mapping strategy allows me to assign another audio file and controller data streams to the range above 500 (500-4095). I used this method to generate multiple, unique data streams from a single axis of the interface. This mapping is performed in Max/MSP in real time.

## **SOUND AND SYNTHESIS**

The synthesis processes used in this work include audio file playback and manipulation, granular synthesis, and a custom software instrument I created. The primary software used for sound generation in this work is Ableton Live.

Audio file playback and manipulation can be heard every time I breach a threshold while reaching high over my head. The resultant sound from this action is the sound of a bell. An audio delay effect is applied to this sound to create a repeating rhythm that fades in amplitude over time.

Ableton also serves as a host application for Absynth 5, by Native Instruments.<sup>19</sup> Absynth is a plugin synthesizer that provides various options for additive and subtractive synthesis within a host application (Ableton). In this work, I use a custom synthesizer sound I created for this composition. This sound is a combination of a pad and string texture with a Low Frequency Oscillator (LFO) modulating frequency values in real time. This sound can be heard during the final minutes of this work.

---

<sup>19</sup> “Komplete : Synths : Absynth 5 | Products.” Last modified 2018. Accessed March 19, 2018. <https://www.native-instruments.com/en/products/komplete/synths/absynth-5/>.

The first solo section (B) that features an audio file of spoken text is generated from the Phase-Vocoder-Sampler (PVS) in Max/MSP. This software is embedded inside Max/MSP and functions in a similar way as a granular synthesizer. An audio file of spoken text is loaded into the PVS and controlled by the Gametrak data generated from my actions.

Granular synthesis is the primary form of synthesis in this work. Every audio file of voices in this work comes from the Max for Live device, Granulator II. A Max for Live device is a Max patch that operates inside of Ableton Live. Max4live devices work on audio and MIDI tracks. This device allows seemingly endless control over an audio file. The controls used in this work are Grain — size of the window; Pitch — frequency of audio output; Position — the current location of playback within the audio file; and Spray — an amount of randomness applied to the Position control. Each of these controls are modulated in real time as a result of my performance with the Gametrak.

## **VISUAL DIMENSION**

Gametrak data generated by my movements is transmitted from Max/MSP to Processing to control the visual output. The visuals for this performance are simple shapes that generate two-dimensional (2D) line drawings (Figure 3). The visual sequences in this work include points, lines, and curves that are controlled directly from the position of each 3D joystick on the Gametrak. This mapping of my physical movements to a 2D line drawing works well because my performative actions are very big in this composition. I considered mapping my movements to objects in a 3D space

inside of Processing. However, I chose to map the X-axis data and Y-axis data to the X and Y axes of a projection screen. The third dimension in Processing is represented by perceived size of objects from the viewers perspective. After careful consideration, I decided to map the Gametrak Z-axis to the overall brightness of the elements in the processing sketch. This mapping approach produced a better result than translating interface data to a literal 3D environment.



**Figure 3.** Screenshots of visual output for each section.

Because of my vertical height and horizontal reach, my movements and their musical (and visual) outcomes are easily observable. I am not trying to mimic my physical movements in the drawings, nor am I concerned with a direct correlation between the visuals and the sound. The connection is metaphorical. However, I am monitoring the visual output during performance.

Processing is used to generate the visual content for this work. Processing receives incoming data from Max/MSP via the OSC protocol. This composition can be divided into larger sections that each have a unique visual design. Dividing into required careful arrangements of the code in Processing. Essentially, I needed to turn off (or on) blocks of code. I achieved this through a series of smaller functions and boolean variables to determine which block should be active at any time. I change the state of these boolean

values that control the blocks of code with the SoftStep 2. Because these changes also happen as the musical sections change, I also use the SoftStep 2 to perform all switching and routing actions for the audio and visuals at the same time.

Processing runs on a loop, and executes all of the code in its loop as many times per second as possible. The actual speed of the loop depends on the requirements of the code as well as the CPU performance. I intentionally set Processing to run its loop 24 times per second. I selected this number because it is the frame-rate of film and I like the look of film. Processing's main loop is called `draw()` and it is the primary loop.<sup>20</sup> The code inside of the main loop is responsible for everything that Processing displays.

The `draw()` function in Processing is additive in the sense that Processing does not clear the screen after every frame unless I explicitly tell it to do so. This allows for almost endless variations in visual output. Clearing the screen erases everything currently displayed before redrawing the next frame. Since every iteration of the main loop is a single frame that repeats 24 times per second, clearing the screen at the start of the loop removes the traces of previous frame before redrawing.

In this work, I am always in control of clearing the screen through my performative actions. I clear the screen by pulling one of the Gametrak string-pots as far as I can reach. At 90% of my vertical reach, a threshold is broken that triggers the clearing of the screen. The first time I use this technique can be seen at 0:47 in the accompanying video of this work. I intentionally disguised this utility in my performative movements. Triggering this threshold also triggers a bell sound at a random (from a

---

<sup>20</sup> Processing Foundation. "Setup and Draw \ Examples \ Processing.org." Last modified 2018. Accessed March 19, 2018. <https://processing.org/examples/setupdraw.html>.

selection of predefined pitches) pitch. This sound is feedback to me that I breeched the threshold and the screen is clear. This feedback is important to me as I am not typically facing the projection screen during performance.

## **COMPOSITION**

The overall structure of this work is ABCDA. The beginning and ending act like bookends to the three primary sections. The ‘A’ sections are a muted, subdued selection of audio files performed from a granular synthesis module. The result is an audio file of spoken words that are triggered over and over, altered on top of one another. My goal was to create multiple conversations overlapping one another. The continuity in the material comes from the source of the audio, the identical language, and the processing applied to these sounds.

The sound worlds created from each section, and the necessary actions required to perform them, can be seen as a progression of layers in both sound and control. The first section is a single audio file controlled by a single joystick. The second section contains two audio files controlled by two joysticks, but each mapped clearly to a unique visual outcome. The third section contains two audio files controlled by two joysticks, plus the Absynth 5 texture and the audio file from the first section, all mapped to a more complex visual output. My compositional and arrangement process for this arrangement is mostly additive.

Other than the audio file of the bell sound that plays rhythmically when I breach a threshold with the controller, the primary sound world of each section is a solo, or

sometimes a duet of audio files of spoken text. The pitch of each iteration of these audio files is triggered by the Gametrak controller sending MIDI data. The pitches are varied, but still correlated to the pitch generated by the controller. During the final section, the pad texture is triggered and provides a rich harmonic structure of chords that change every 13 seconds. The chords for this part are set in Max/MSP and are in the key of B minor.

## **PERFORMANCE**

An application of my movements mapped to sound can be heard in the first solo section (B). My action is to pull one of the string-pots out of the Gametrak controller very slowly. This vertical movement is mapped directly to the timeindex of an audio file and can be seen at 0:55 in the accompanying video. An audience can observe that my pulling of the string causes an audible change to the audio file and the visual content. The position of a large dot is controlled by the X-axis and Y-axis data from the Gametrak. The Z-axis data controls both the size of the dot and the timeindex of the audio file. These movements are mapped to a smaller range of the string-pot. This mapping allows me to perform with the audio file by pulling a smaller range of the string-pot. This smaller physical movement contrasts the larger physical movements in the work.

In the final section, my movements shift to parametric control over the sound and visuals. The visual output during this final section is a series of points that are drawn at a user-controlled radius from the center. The radius is controlled by the Y-axis of the right joystick on the Gametrak. The X-axis of the left hand repeatedly draws the points around

the radius set by the Y-axis of the right hand. The final outcome is a series of smaller points that populate around a radius, creating circles on the screen. Moving my right hand forward on the Y-axis while my left hand is stationary, creates a straight line made of points that travels perpendicular to the drawing of the circles. As these patterns repeat with my actions, the screen becomes full with larger circles connected by lines. The visual result of my actions can be seen at 7:00 in the accompanying video.

Finally, I have a SoftStep 2 MIDI foot-controller that I use for various applications. The SoftStep 2 has ten switches. The switches are programmed to control various aspect of this performance. Among the controls are: clearing the Processing screen without playing an audio file; advancing the drawing modes in Processing; audio routing in Ableton Live and Max/MSP; and starting/stopping the chords in the final section.

## III.2 OURIANA'S MEDITATION

A multichannel, real-time, interactive composition for a MusicBox, Max/MSP, and Kyma.

### INTRODUCTION

*Ouriana's Meditation* is a play on the idea of a muse and inspiration. Ouriana was a goddess of music. In Greek lore, "Ourania and the other Younger Muses would entertain other gods, to sing, dance and retell stories, especially about the greatness of Zeus."<sup>21</sup> The question I ask myself in this work is "Who inspires a Muse?" A muse is thought of as providing inspiration to artists. However, what inspires a muse? Do they need to be inspired? Through the use of a MusicBox as an interface, this work explores this question through the sounds of bells and gongs. I often associate the sounds of bells and gongs with angelic stereotypes.

This composition is well-balanced in the frequency space and offers a wealth of material in extreme high and low frequencies. This work features audio files of bell sounds, finger cymbals, a triangle, a glass lid to a large stock pot, and a set of silver spoons strung together like small wind chimes. Other audio files include a 21 ounce

---

<sup>21</sup> "Ouriana in Greek Mythology - Greek Legends and Myths." Accessed March 19, 2018. <http://www.greeklegendsandmyths.com/ourania.html>.

HydroFlask water bottle, a siren, Rototoms, a Paiste Roto-Sound cymbal, small gongs, and two small, hand-cranked music boxes purchased from a toy store.<sup>22 23 24</sup>

Kyma is the primary sound synthesis engine. Synthesis techniques include simple audio file playback, replicators, and extreme pitch shifting. Four-way faders to realize the audio output. Audio files of tiny, hand-cranked music boxes both begin and end this composition.

The MusicBox as an interface works well for this composition. The metaphor of a muse elicits an expected response of magic and lore in an audience, and the jingly bell sounds produced through specific movements combined with the angling and shaking gestures used to shape the resultant sound makes it easy for an audience to create their own internal narrative. After one of the performances of this work, an audience member shared his perception of the narrative with me. His take on the narrative was that the interface was a magic book that once belonged to a muse and somehow I harnessed its magic to tell the stories in the book through my performance. This response was exactly what I want to provoke with this work.

My performative actions are magical to me. Although the onset of all sounds is caused by me pressing a small button, I mask my interactions with the button by angling the box to suggest my movements cause the onset of sound.

---

<sup>22</sup> HydroFlask. "21 Oz Standard Mouth Bottle | Hydro Flask." Last modified 2018. <https://www.hydroflask.com/21-oz-standard-mouth-flex/color,pacific,a,92,o,58>.

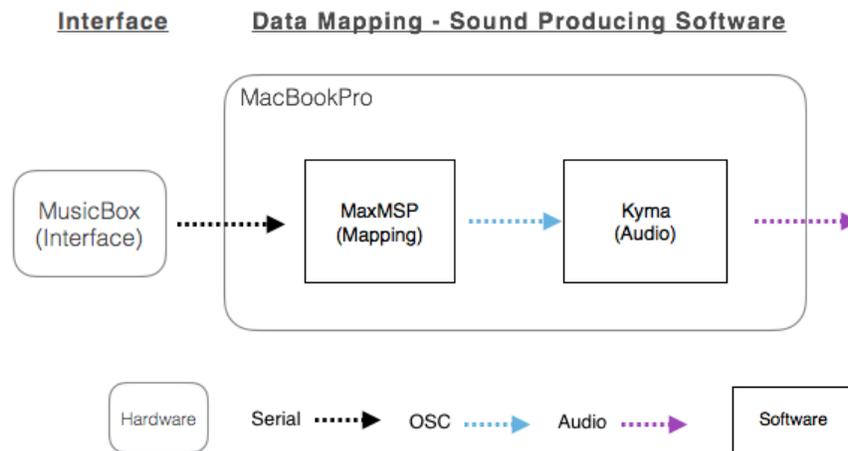
<sup>23</sup> Remo. "REMO Rototom®." Last modified 2018. Accessed March 19, 2018. <http://remo.com/products/product/rototom/>.

<sup>24</sup> Paiste. "Percussive Sounds Percussion Set." Paiste: Percussive Sounds. February 08, 2007. Accessed June 03, 2017. [http://www.paiste.com/e/product\\_any\\_det.php?menuid=67&modelid=37](http://www.paiste.com/e/product_any_det.php?menuid=67&modelid=37).

## INTERFACE, HARDWARE AND SOFTWARE

The data-driven instrument used in this composition consists of a MusicBox controller, Max/MSP, and Kyma (Figure 4). This particular data-driven instrument is the simplest of my DDIs in terms of its number of components and complexity of setup.

The physical interface for this composition is constructed from an acrylic box. The box was created from a line drawing in Adobe Illustrator and physically realized in frosted white acrylic.<sup>25</sup> The acrylic was cut using a laser-cutter.

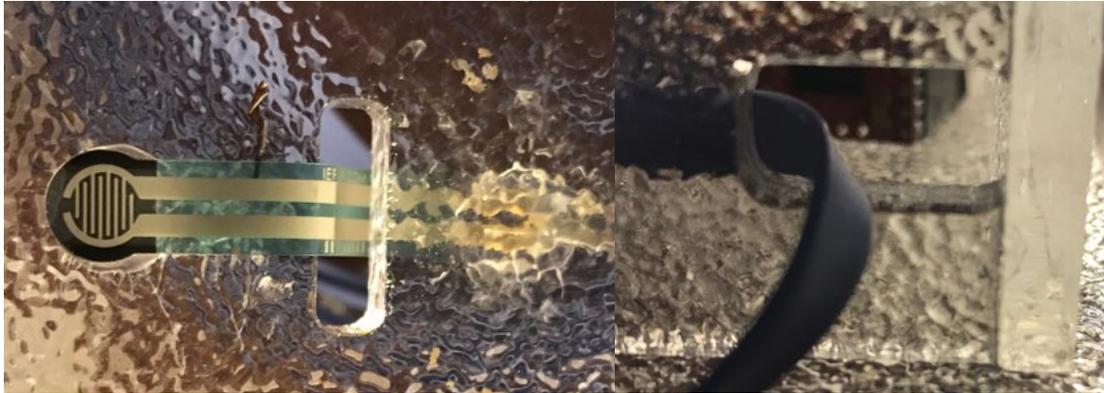


**Figure 4.** Diagram of data-driven instrument for III.2 *Ouriana's Meditation*.

The finished box is nine inches wide, seven inches deep, and two inches tall. The top and bottom of the box are these dimensions, while the body of the box is slightly smaller than the top and bottom to create an overhang. There is a small hole cut into the backside (performer side) of the box that allows a standard USB cable to pass into the body of the box (Figure 5). The top lid of the box has small acrylic hinges glued to the inside to allow the lid to open and close. The lid also features a very tiny slot to allow a

<sup>25</sup> Adobe. "Adobe Illustrator CC." Adobe. Accessed June 03, 2017. <http://www.adobe.com/products/illustrator.html>.

micro-FSR sensor to pass (Figure 5) . This tiny sensor can be seen at 1:34 in the accompanying video.



**Figure 5.** Two small holes cut into acrylic for FSR and USB cables to pass

The technology inside of this box is a button and an accelerometer sensor connected to an Adafruit Trinket (Trinket).<sup>26</sup> The accelerometer is glued to the center of the underside of the lid using standard hot glue. This placement aligns the X-axis of the interface (the sensor) with my physical X-axis, and the Y-axis of the interface (the sensor) with my physical Y-axis. This translation of movement is easily appreciable by an audience at almost any angle of view. In most cases, I map the data from the X-axis to control pitch in the resultant sound. I do this because I am controlling pitch often, and moving the interface horizontally (in relation to the audience) makes my performative actions easier for an audience to see.

Finally, the micro-FSR is connected to the Adafruit Trinket and acts like a simple button in this interface. Though the micro-FSR sensor measures a variable amount of pressure on its surface, my software generates an event when the amount of pressure

---

<sup>26</sup> Adafruit. “Adafruit Pro Trinket - 5V 16MHz ID: 2000 - \$9.95 : Adafruit Industries, Unique & Fun DIY Electronics and Kits.” Last modified 2017. Accessed March 19, 2018. <https://www.adafruit.com/product/2000>.

breaks a threshold at roughly 80% of the full force of me pressing my thumb against the sensor. The FSR sensor is small enough to pass through the tiny slot in the lid of the interface. The sensor is glued to a spot that is easy for me to reach with my left thumb (on the top lid of the interface). This design hides the micro-FSR from sight and helps to mask my interactions with the button. This masking of my actions is needed because I have choreographed specific physical movements to suggest the onset of every sound I perform.

I intend to suggest, through my actions, that these movements trigger the onset of sound by shaking the interface as if I am holding a box full of bells. My movements are intended to suggest that I am moving magical bells around the inside of the box, and these *magical* bells are generating the resultant sounds.

The software used in this composition includes Arduino IDE, Max/MSP, and Kyma. The performance interface connects to Max/MSP (on a laptop) as a serial device over USB. The accelerometer and micro-FSR sensor data is processed by an Adafruit Trinket microcomputer. The Trinket captures the input from the sensors and scales the data streams to a predetermined range before sending the data over USB to Max/MSP. The raw data from the micro-FSR sensor ranges from 0-1023 because the analog to digital converter (ADC) in the Trinket has 10-bit resolution.<sup>27</sup> However, the raw data from the accelerometer is between 0-89 expressed as degrees of rotation. In both cases, the raw data is scaled to a smaller range before being output to Max/MSP.

---

<sup>27</sup> SparkFun “Analog to Digital Conversion - Learn.sparkfun.com.” Last modified 2017. Accessed March 19, 2018. <https://learn.sparkfun.com/tutorials/analog-to-digital-conversion>.

The raw data values from the micro-FSR and accelerometer sensors are far greater than can be transmitted over USB serial in a single packet. This means that numbers larger than 255 (USB serial limit is 0-255) need to be split, sent over USB in multiple packets, then reassembled later in software. I chose a different solution to transmit the data over USB serial. I scaled the raw data from the sensors to 0-200 on the Trinket. I chose 200 because I knew it would survive the limits of USB serial, and it was an even number multiple to eventually scale to 0-1. I originally used a data range of 0-100 from the Trinket, but later realized I would achieve greater resolution with a range of 0-200, while avoiding the need for additional processing power to reassemble and scale the data. I use the OSC communication protocol to transfer the data from Max/MSP to Kyma in real time.<sup>28</sup>

The interface connects to Max/MSP over USB, Max/MSP scales the incoming data to a range that is appropriate for Kyma. In Max/MSP, the incoming data is scaled from its new raw data range of 0-200 as an integer, to a range between 0-1 as a decimal (float). Because Kyma is the only destination for data, and control in Kyma is largely expressed as a range from 0-1, no further data scaling was needed. Before the final output to Kyma, the data streams are smoothed. Once smoothed, the data is sent into Kyma to control audio in real time.

Data from the interface flows in the following path: The incoming integer from the Trinket is in the range of 0-200; This incoming range is then scaled to 0-1 as a decimal in Max/MSP. The output is then averaged in a parallel averaging algorithm that

---

<sup>28</sup> CNMAT. "Introduction to OSC | Opensoundcontrol.org." Website. Last modified 2018. Accessed April 25, 2018. <http://opensoundcontrol.org/introduction-osc>.

divides the incoming number, averages the two halves, then adds the two halves back together, before it averages one last time. This operation is performed every 33 milliseconds on the incoming data to generate a smooth data output from the accelerometer input. Smooth data is important in this interface because I am attempting to control the amplitude of sounds gradually by angling the interface. If the data stream is noisy (non contiguous values), the resultant change in amplitude will be less effective, compositionally (Figure 6).



**Figure 6.** Raw and smoothed data streams from interface

The MusicBox as an interface is really quite simple in terms of form and function. The interface houses a single button and an accelerometer. The button is used to start sounds, while the accelerometer is used to provide ongoing control over those sounds. These controls are incredibly basic when you compare it to other more traditional instruments. The interface in this work has only 1/88<sup>th</sup> the number of buttons (keys) as a typical piano. However, for this work 88 keys are not needed. Instead of pressing many different buttons to make different sounds, I press one button and make many sounds with accompanying choreography. The ability to change the functionality of an interface in real time is an advantage of a data-driven instrument. The choreographed movements replace the actual pressing of buttons in this work — as far as a spectator is concerned. This movement is a *sleight of hand* action that is used to help further the narrative of a

MusicBox by directing attention to my performative actions rather than the pushing of a button.

An audience can easily observe that the interface is some type of box. The performance begins with me opening and closing the lid. While the interface is in my hands during performance, it is easy for an audience to see the top surface and track its movement. My entire performance is choreographed to portray the interface as having objects inside of it. The combination of my performative actions and the opening and closing of the lid sets a very clear expectation for this interface to behave like some type of magic box.

The data generated by performing with this interface is mapped to sound production in similar ways throughout this work. Physically, the onset of sound is mapped to my shaking movements of the interface from side to side, while ongoing control over sound is mapped to my angling of the interface on its X and Y axes. This specific mapping scheme is connected directly to my performative actions. These performative actions can be seen at 1:17 in the accompanying video. Though pressing the micro-FSR triggers the onset of sounds, my movements are meant to suggest otherwise.

## **SOUND AND SYNTHESIS**

Kyma is the only synthesis environment used in this work. I used a Kyma Timeline to organize the sounds as events that unfold over time. Similar to other works in this dissertation, I through-composed each section of this work and intentionally created moments of pause at some of the transitions to allow a brief moment of improvisation. I

utilized the WaitUntil object in Kyma to allow for these brief moments. These are my favorite moments in this works because I am simply exploring the interface and its resultant sound.

Many of the bell sounds in this section are played from a Replicator prototype in Kyma. The Replicator prototype allows one press of the button to trigger several iterations of the bells at various timings and pitches. This approach to audio file playback provides the necessary sound world for my choreographed movements as I suggest the multiple sounds are a result of my movements. During the opening section (see 1:37 in the video), the overall pitch of the bell sounds move up and down. The rate of change is slow enough that the resultant sound appears to gently waver its pitch rather than modulate erratically. At 3:08 in the video you can hear the pitch moving downward. This result is caused during performance by the automation data in the Kyma Timeline. Because this change to pitch over time is automated in the Timeline, the result is consistent from one performance to the next.

The primary sound production during the middle section of this work is generated by SampleCloud prototypes in Kyma. The SampleCloud prototype plays an audio file using granular synthesis. Granular synthesis is vital to the middle section of this work, as it provides a gentle smearing of the audio files.

The solo sections rely on audio file playback and manipulation in real time. In these sections, pitch is the primary parameter being controlled during performance, by angling the interface horizontally, from side to side. In the first solo section that begins at 3:45 in the video, the audio file used during the first solo section is from a HAPI drum. I

trigger the onset of the sound by pressing the micro-FSR. This first solo has a rich texture because I am layering four copies of the same audio file at different, but related pitches. I refer to these sections as solos because I have created a musical environment to that allows me to improvise.

## **VISUAL DIMENSION**

The visual element in this work comes from an RGB par light placed on the ground directly in front of my performance position, aimed straight up. This placement allows me to perform with the interface over the light. The light is clearly visible at 2:40 and 3:22 in the video of this work. Since the par light is set to change colors based on audio sensitivity, I do not have to provide unique control to, or communicate with, the par light.

The par light is set to a mode that responds to the amplitude of live audio in its immediate surroundings. The sensitivity of the input on the light is adjusted to allow changes only on larger spikes in amplitude. This creates the illusion that the light is changing as a result of my movements. The light cycles through its three colors in the same order: red; green; and blue. Because the interface is clear acrylic, the light illuminates the interface from below and adds another element to the mystique of the narrative by illuminating the interface from below. The interface glows from the light under it, rather than light from inside of it.

## COMPOSITION

The musical structure of this work is similar to a composition written for a small jazz ensemble. Among its sections are a head, structured improvisation, solos, a bridge, and a dramatic ending. The head and ending are very similar in content and presentation. Each of the moments where Kyma waits for me to finish my musical phrase is structured improvisation. The two primary sections are me soloing. The middle section is the bridge that links the solo sections. The beginning and ending are like sonic bookends to the entire composition.

The structure of this composition does not follow any formal pattern. By formal, I mean sonata form or any other pre-defined musical form. Instead, I developed the sections based on process. My intent is to introduce the interface and the narrative through the opening sounds. The larger structure of this work can be broken down into smaller motifs of tension and release. The work begins with tension and is immediately released through the low pulsing sound. I transition to this low pulsing sound to invoke a calming response in the listener. From this new perspective of calm, I begin to add layers to repeat the process. This transitions appear at 1:01 in the accompanying video.

The two primary solo sections are quite similar to one another. They are both created from audio files of bell sounds and the interface functions the same for both sections as well. The difference is the sound world. In the first solo section, I am triggering softer bell sounds with a gentle attack. These sounds are processed in a way to generate a very smooth sounding timbre. Moving the interface along its X-axis changes the pitch by sliding into each new note. This creates a unique texture that is both

interesting and subdued. The four audio files used to make this first sound are recordings of a HAPI-drum steel tongue drum. I layer these four audio files at different pitches. However, instead of choosing pitches based on a musical scale, I chose specific frequencies and ratios. The audio files are triggered by my pressing the micro-FSR. However, the pitch of the audio files is controlled by the X-axis data from the interface. The raw data from the interface is in the range of 0-1 as a decimal. The X-axis value is multiplied by either 600 or 900, the product of that equation is then multiplied by 0.89 or 1.11 as a final scaling action (Table 1). The result of this equation is expressed in Hertz and used to generate frequency values. This section begins at 3:45 in the video.

The second solo section presents a bell sound that has a very bright and harsh timbre. This sound is not only louder than the previous solo section, it also contains more extreme high and low frequencies. This emphasis of high frequencies is caused from the source recording of a very high-pitched bar on a glockenspiel. The timbre of the modified version of this sound is created by pitch shifting the audio file down several octaves. The result is a smooth, yet sharp sonic texture that has a quick attack, a slow decay, with a metallic texture. This sound is created by four audio files in a similar way to the first solo section. However, the equations are slightly different. The X-axis value is multiplied by either 400 or 600, the product of that equation is then multiplied by 0.93 or 1.07 as a final scaling action. This section begins at 9:30 in the video.

**Table 1.** Equations used for frequency values in both solo sections of *Ouriana's Meditation*.

<b>Section</b>	<b>Equation</b>	<b>Range of Values</b>	<b>**Between Notes (highest)</b>
Solo 1	$(!PenX * 600) * 0.89 \text{ hz}$	0-534	C5-C#5
Solo 1	$(!PenX * 600) * 1.11 \text{ hz}$	0-667	E5-F5
Solo 2	$(!PenX * 900) * 0.89 \text{ hz}$	0-801	G5-G#5
Solo 2	$(!PenX * 900) * 1.11 \text{ hz}$	0-999	B5-C6

\*\* The highest possible frequencies fall between these pitches

Both solo sections are affected by the equations that generate pitch for the sounding audio files. The equations used for the two sections are not identical in output, but are identical in process. They each take the incoming data and apply a simple math operation to generate a new pitch. The pitches related to the frequencies I am generating exist between notes on a piano — outside of Western tradition (Table 1). Even though the actual frequency values may exist between the notes on a piano, the relationships of the four audio files in both solo sections is consistent and precise because of the scaling equations. The irony in this is that both sections actually create the same pitches at different octaves. Both pitch sets create an interval of an augmented fourth, also known as a tritone — or the Devil's Interval.<sup>29</sup> This Devil's Interval appears in a composition about a muse and heavenly bell sounds.

The order of these two solo sections can be viewed as an example of release before tension. However, the order was set to create a structure that dramatizes a structural crescendo from the first solo section to the second.

---

<sup>29</sup> “A Brief History of the Devil's Tritone | Mental Floss.” Last modified 2018. Accessed March 19, 2018. <http://mentalfloss.com/article/77321/brief-history-devils-tritone>.

## PERFORMANCE

This composition begins with the interface sitting on a platform stand that is roughly four feet high. My first interaction with the interface (MusicBox) is to slowly and carefully open the lid. This action happens in silence at 0:33 of the accompanying video. Opening the lid breeches a threshold in the accelerometer data that triggers the first sound. I pause for a brief moment before I quickly close the lid, which triggers the next transition with a huge explosion. The simple act of opening and closing the lid that starts all sounds in the beginning and ending sections of this work.<sup>30</sup> The opening of the lid can be seen at 0:33 and the closing of the lid can be seen at 12:12 in the video of this work.

The other primary performative actions in this work involve me angling the interface on the X-axis to trigger the onset of the bell sounds, and shaping their ongoing sound by shaking the interface gently on its Y-axis. This approach is visible in both solo sections, and at 1:47 and 4:33 in the video of this work.

I trigger tiny bell sounds by pressing the micro-FSR on the top of the interface. Since the action of pressing my thumb is not perceptible by an audience, I take advantage of the composition's theme by performing specific, choreographed actions to accompany the triggering of these bell sounds. These actions involve quickly angling and moving the interface from one X-axis extreme to the other to suggest that there could be bells that are bouncing around inside of the box. These movements closely match the resultant sounds. The result of these movements can be seen during the final 50 seconds of the video of this work.

---

<sup>30</sup> YouTube. "Wolfgang Amadeus Mozart - Wiegenlied (Lullaby), K. 350 - YouTube." Accessed March 19, 2018. [https://www.youtube.com/watch?v=vXRzq\\_8bJ3c](https://www.youtube.com/watch?v=vXRzq_8bJ3c).

During the middle section of this work, the X-axis and Y-axis data is used to control two audio crossfaders. Each axis controls a crossfader that switches smoothly between two different audio files. Manipulating the interface along either axis will mix between its two audio files. This mapping scheme allows me to use the interface to seamlessly crossfade between four audio files by angling the interface. These movements are very easily appreciable by an audience. These movements can be seen from 5:00 - 9:30 in the video of this work.

### III.3 CHANNELING RACHMANINOFF

A multichannel, real-time, interactive composition for Lux Codex, Max/MSP, and Kyma.

#### INTRODUCTION

*Channeling Rachmaninoff* is derived from the audio recording of Sergei Rachmaninoff performing his work, *Prelude in C# Minor Op. 3, No. 2*.<sup>31</sup> The performance interface used in this work is a custom-made light projector I designed for this composition. This light projector, the Lux Codex, uses piano rolls intended for player pianos to filter the light projected through its surface. A small web-camera is pointed at the piano roll to transfer the light data into software for manipulation.

The goal of this work is not to re-mix Rachmaninoff's composition. Rather, the goal is to use the recording of the composer performing his composition as shorter audio files to create the sound output for this work. All sound in this work is derived solely from the recording of Rachmaninoff performing his composition.

The visual output for this work is generated in Jitter, a component of Max/MSP. The video output from the web-camera is modified in real time using various methods to produce a colorful collage of colors, shapes, and patterns that are generated as a result of my performative actions with the Lux Codex.

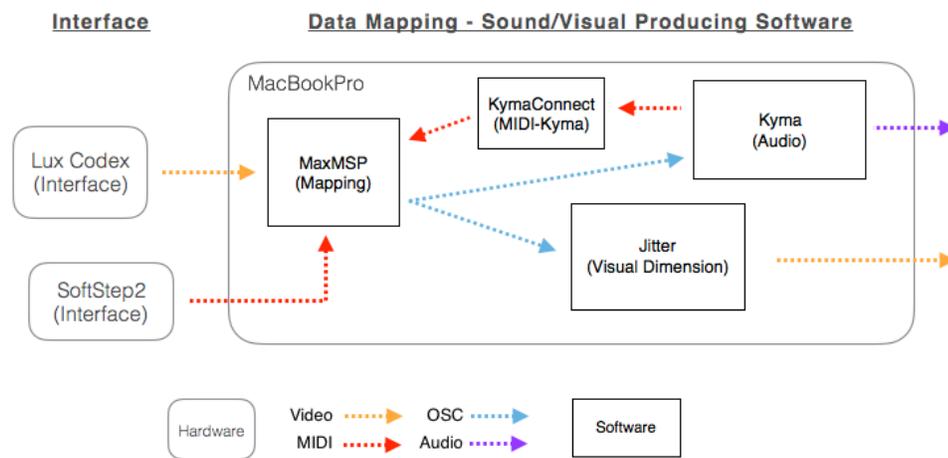
I accomplish magic by repurposing piano rolls to generate the data used to manipulate my repurposing of the audio recoding of Rachmaninoff. However, transforming light into data that is used to create sound is pure magic to me.

---

<sup>31</sup> Sergei Rachmaninoff. "Prelude in C Sharp Minor, Op. 3, No. 2." Decca, 1990.

## INTERFACE, HARDWARE AND SOFTWARE

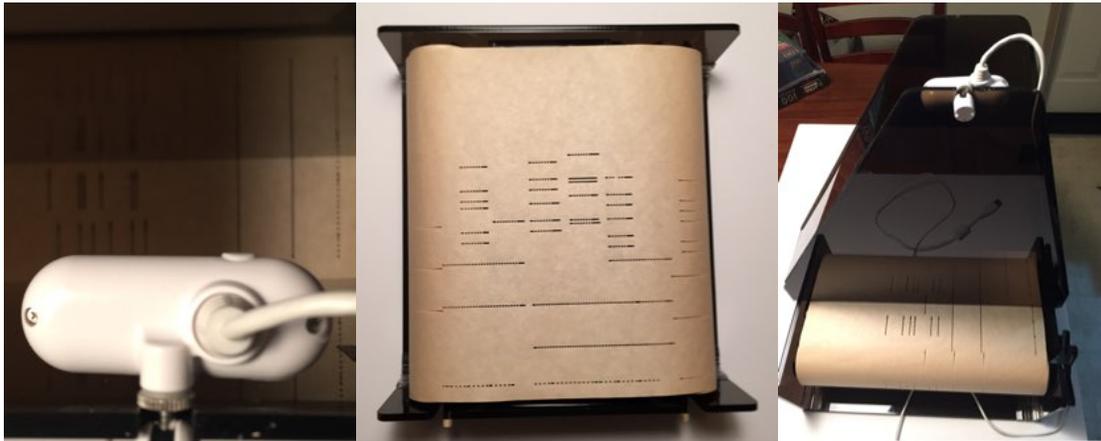
The data-driven instrument in this composition consists of a custom interface the Lux Codex, SoftStep 2 foot controller, Max/MSP, KymaConnect, Kyma, and Jitter (Figure 7). The Lux Codex is the primary interface and the SoftStep 2 is used to control aspects of Jitter.



**Figure 7.** Diagram of data-driven instrument for III.3 *Channeling Rachmaninoff*.

The physical interface for this performance is a custom light projector. I designed this interface in Adobe Illustrator and laser cut its pieces from dark acrylic. This light projector is the primary interface called the Lux Codex (Figure 8) used in this

composition. Lux Codex translates from Latin to *Light Code*.



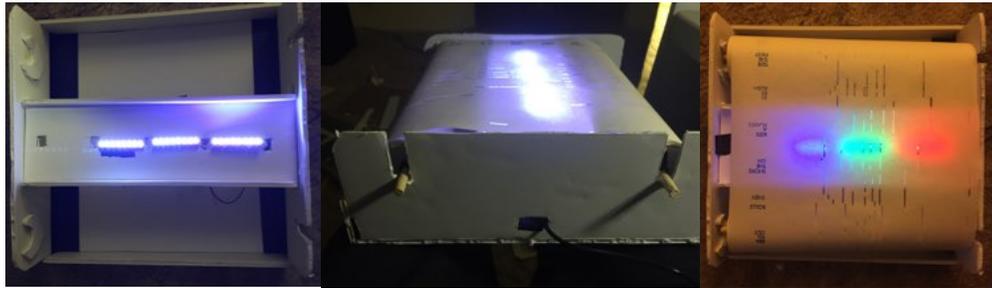
**Figure 8.** Photos of finished Lux Codex.

I created the Lux Codex for the purpose reinterpreting paper rolls intended for player pianos. The Lux Codex transforms the projected light through the piano roll into robust data streams. This interface functions by projecting red, green, and blue light upwards through the paper piano roll where it is captured by a small webcam that is connected to the computer. This arrangement can be seen clearly at 1:17 in the video of this work.

I control the Lux Codex in real time during a live performance. As I turn the take-up reel, the piano roll passes over the lights (see 2:47 in the video). As the holes in the piano roll move over the lights, the overall amount of a given color of light can change greatly. The overall intensity of each color is processed by Jitter and routed to both sound and visual producing algorithms. On the piano roll, pitch is arranged from lowest (closest to the performer) to highest (farthest from performer). This arrangement situates the red lights under the lowest frequency data from the paper roll. Depending on the content of the composition on the paper roll, the light intensity of each color will vary widely. My

performative actions in this work involve spinning the two take up reels in two directions — forward, and reverse (see 7:00 in the video). I can see the piano roll and its holes in relation to the lights. This helps me to estimate the amount of light, and the color that will be dominant at any point in the performance.

I created the first working prototype of this interface using white poster board (Figure 9). The Lux Codex is made up of two parts. The base holds all of the moving parts, lights, and interface handles. The top holds the webcam in the correct position. The size and shape of the top is vital to the webcam to properly function.



**Figure 9.** Working prototype of Lux Codex interface.

During my earliest tests, I was unable to capture the color data from the lights. I tried different processes in Max/MSP using Jitter to capture the data. I also used Processing to attempt to only look at a small strip (playhead) of the paper to determine the color values. I tried different cameras, piano rolls, lights, etc. I struggled to isolate the light coming from my LEDs from the ambient light in any room. After I finally constructed the top portion of the interface, it all started working. I realized that I needed to *physically* isolate the area between the lights and the webcam. Poster board was great for this because it is opaque. However, I quickly realized that white reflected too much light, so I switched to black poster board. Eventually, I chose to build the Lux Codex

from dark acrylic. I chose 0.25 inch acrylic to create better isolation for the webcam. My first version of the top was a giant, square-shaped box. This was an easy shape to cut out of poster board. After I understood where the camera needed to be in alignment with the light bar, I started cutting away at the unseeded parts of the giant square-shaped top. The final result is a top that resembles an overhead view of a grand piano (10:00 in the video). This was a very happy accident. I perform this work with the Lux Codex on a small table, me sitting on a piano bench, and the small foot controller on the floor. This arrangement appears to be a very *piano-like* construction.

The base of the Lux Codex is made up of a frame that holds the main light bar, both take up reels, and the top hood. The sides of the base and all of the parts (except for the take up reels) fold flat for travel. The feature of portability was important in my design.

The light bar holds 24 LEDs: eight red, eight green, and eight blue. The light bar has 24 small holes cut into it, in a straight line. The LEDs fit into these holes from the underside and are glued in place. All 24 LEDs are powered by a small AA battery pack, with a switch. Early versions of the Lux Codex used a series of Adafruit NeoPixels for the LEDs.<sup>32</sup> These LEDs are very bright, but require an Arduino (or similar) microcomputer to function properly. Using standard LEDs allows me to remove the microcomputer, and its sometimes delicate connections from this interface. This update also makes it easier for the Lux Codex to travel in times of heightened security measures at airports. This is especially important as a performance of this work requires all of the

---

<sup>32</sup> Adafruit. "NeoPixels : Adafruit Industries, Unique & Fun DIY Electronics and Kits." Last modified 2017. Accessed March 19, 2018. <https://www.adafruit.com/category/168>.

parts, plus my Kyma system. Except for the take-up rolls, all parts this interface can be disassembled to pack in thinner layers.

The software for this composition includes Max/MSP, Jitter, and Kyma. Jitter is included in Max/MSP and provides an environment for programming visual elements similar to audio. The webcam is connected to the computer via USB. Jitter accepts the incoming video and splits it into three columns (columns of a typical video signal). Each column captures a single color of light. Max/MSP generates an integer data stream from 0-255 to represent the brightness of each color. The raw video input signal is also used to help generate the visual output.

My software generates three unique data streams from the brightness of each color (see 6:07 in the video). The first data stream is the raw data from the incoming signal. The second is the raw data scaled to a range from 0-1. The third is the scaled data, smoothed. All three data streams are used in different ways throughout this work to create and manipulate different aspects of the sonic and visual output.

Proper operation of the Lux Codex relies on light detection. Light detection is a form of computer vision and can be very problematic for live performance. Light is different in every room or environment. The actual color of the light may seem like white to human eyes, but a sensitive webcam can detect a more accurate version of light. This means that the Lux Codex could behave in unexpected ways under different lighting scenarios. I addressed this inconsistency by creating a calibration patch in my Max/MSP software (Appendix B). I calibrate the interface to the performance space before every performance. Proper calibration involves turning the take up reels through the entire

length of the piano roll. During this process, Max/MSP records the full range of output from each color of light. This output is used to set the maximum and minimum ranges on the incoming data from the webcam for scaling. This calibration process is vital for the Lux Codex to properly function.

I use a SoftStep 2 foot controller with an attached expression pedal for various means of control during performance. Primarily, this foot controller is used for utility functions during performance, like activating the WaitUntil objects on the Kyma Timeline, changing visual filters, resetting the main melodic motif (if desired), and toggling on/off the smaller picture-in-picture of the raw video signal. The SoftStep 2 connects to my laptop via USB.

## **SOUND AND SYNTHESIS**

Kyma is the only synthesis environment used in this composition. I use Kyma to playback and manipulate audio files in real time, perform subtractive synthesis, and control various granular synthesis processes. Kyma also outputs MIDI data to control the visual output in Jitter.

I use a Kyma Timeline to arrange the sounds for performance. I through composed every section of this work, but there are also times where the Kyma Timeline waits for me to finish my musical phrase before continuing.

This composition begins and ends with a literal playback of Rachmaninoff's original performance is his *Prelude in C# Minor Op 3, No. 2*, as noted above. This audio file is being played from a TAU prototype in Kyma. Because the TAU allows for

independent control over the timing and pitch of an audio file, I am able to slowly step through the timeindex of the audio file. This particular audio file plays back at a rate of just a few milliseconds at a time. The result is a new melodic motif that is hidden deep inside of the complex waveforms of the audio file used in the TAU. When performed slow enough, this sound creates a melodic contour that repeats. This melodic contour has a very tight, chromatic texture, but is discernible as a clear melody. The timbre of this sound is more like bagpipes than a piano, its actual source. This work also ends with audio file playback generating the final few chords of this work as a sonic bookend to the beginning.

Granular synthesis plays an important role in this work as a primary sound synthesis technique. I use granular synthesis to playback audio files as smeared textures while maintaining some of the characteristics of the source audio. In Kyma, granular reverb is used on several non-granulated audio files. This result is a clear representation of the source audio, with granular synthesis applied to the reverberated representation of the same source.

Subtractive synthesis can be heard in sounds that are more textural than motivic. These sounds begin as simple oscillators or audio files and are filtered to create the resultant output. This process produces a high-pitched sound that fills the space around the primary motif. This sound can be heard at 3:37 in the video. I use this process on another sound that also plays at 4:54 in the video. The result of this process is a metallic, high-pitched, pad texture that occupies a separate frequency space as the primary motif.

## VISUAL DIMENSION

The visual dimension in this work was created using Jitter in Max/MSP. Jitter is a component of Max/MSP that processes video using the same visual programming interface as Max/MSP. The webcam captures the light projected through the tiny holes in the piano roll. The image from the webcam is divided into three horizontal columns. The raw video input from the webcam is displayed as a picture-in-picture in the lower right corner of the projection screen. This smaller picture displays the raw image of the light before processing.

The webcam is situated directly over the three colors of light shining through the paper roll. The incoming video signal is split into three vertical columns using Jitter. Dividing the video into columns makes it easier to detect a single color in any of the columns. The left column of video detects only red light, and the red LEDs are centered in this column. The green and blue LEDs are treated the same way as the red. This method allows for a more accurate capture of each color because each column only detects the color that is being directed at it. This method acts as a filter for the other colors for more accurate data collection via light. Once captured, the data from the light streams is used to control aspects of the sonic and visual domain, and provides the main visual content. Several Jitter effects are used to create a visual tapestry for this work. These effects are controlled and modulated in real time, using the incoming light data.

There are eight sections of visual output. The performance begins with a black screen (section 1). When the first motif is performed, a series of animations (as video files) are played together (section 2). These animations contain only small rectangular

shapes that are arranged horizontally on the screen (see 1:45 in the video), that move from left to right. This is a literal, visual representation of the piano roll moving over the lights as the small square shapes move across the screen from left to right. Each new section of visual output is triggered (changed) by MIDI controller messages sent from the Kyma Timeline to Max/MSP. The MIDI controller messages are received by Max/MSP and used to change the mode of visual output from Jitter. This type of control mechanism works because the arrangement and order of visual sections are consistent from one performance to the next. Automating this control also frees me up to focus on other aspects of my performance.

## **COMPOSITION**

This work is unique in this dissertation because it is an example of creative re-composition. Re-composition involves generating a new work based on an existing work. Often, re-composition can be used as a tool to achieve a greater understanding of the compositional choices a composer makes during his/her process. Creative re-composition is a very powerful tool for analysis. The original composition used for re-composition in this work is Sergei Rachmaninoff's *Prelude in C# Minor Op. 3 No. 2*. The sound material for this work was captured solely from an audio recording of the original composition, performed by Rachmaninoff. Because the data for this work is derived from the piano roll score of the same piece, I use Rachmaninoff to generate the data I use to manipulate the audio recordings of Rachmaninoff, performing Rachmaninoff.

In this work, the output of a piano performance is recorded to paper through a series of small rectangular holes that span the width of the paper. The placement of the holes in relation to the width is mapped as pitch by my software and a player piano. The holes are laid out on a long roll of paper. The length of this paper determines the length of the work in a player piano. The length of the holes determines the sustain of any given pitch. This piano roll is not only a result of a recorded performance, it can also be viewed as a musical score because it contains information regarding the unfolding of sound, over time. Timbre is controlled by the physical instrument. Pitch and duration are controlled by the holes in the paper. I use this *score* to then generate data from light passing through the holes. This data, derived from a performance/score of this work, is used to both generate and modify audio and visual output in real time. So, the Rachmaninoff (score) is used to control sounds created from smaller selections of Rachmaninoff, performing this very work. Rachmaninoff is mapped onto Rachmaninoff to perform Rachmaninoff, and it is all controlled by paper and light.

This composition begins with the first few measures of the prelude played back in real time. Under this audio file is another version of itself that is being manipulated beyond recognition. This combination sets up a transition into the new sound world. The opening motif is represented by what sounds like a recorder of some type. This sound is generated in Kyma, by the TAU Player. The audio file is the same content from the opening of the song. In this example, I am using data from the Lux Codex to determine when a new note sounds from the TAU. While the range of a timeindex in a Kyma sound has a range of -1-1, the step size in this example is 0.0001. This extremely granular

approach to playback allows me to find unique moments in a sequence that can be played, paused, or reset.

*Channeling Rachmaninoff* features granular synthesis as well as sample manipulation in Kyma. The middle section employs a Kyma Chopper prototype.<sup>33</sup> This effect chops, or cuts, the audio file into smaller pieces for playback. The result of this effect is similar to playing a sound and very quickly turning the volume up and down, repeatedly. This oscillation of amplitude happens at varying tempos that is generated by the presence of blue light.

The overall sound output from this composition is at times soft and motivic, while at other times it is loud, brash, and even [sonically] abrasive. The visual element of this performance blends seamlessly with the sonic output, and both elements complement one another during performance.

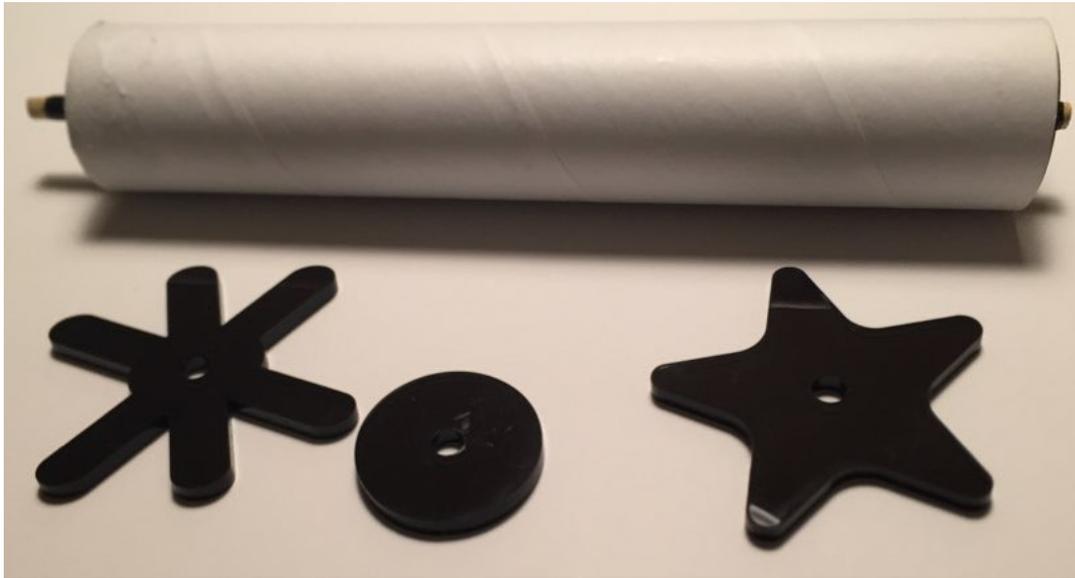
## **PERFORMANCE**

My performative actions in this work are very different from the other works in this dissertation. The Lux Codex is a light projector for piano rolls. In order to generate data, continuing moving the paper over the light bar necessary. As such, data is generated at the rate the paper roll moves over the lights. This method of interaction provides complete control over the data generation, in real time. This method of operation is inspired by player pianos as the paper roll in this work is intended for use in a player piano.

---

<sup>33</sup> Stolet, Jeffrey. *Kyma and the SumOfSines Disco Club*. Edited by Adam Shanley. San Bernardino, CA, 2011.

The Lux Codex houses two cylindrical tubes that have a long, threaded rod running through the center, lengthwise (Figure 10). The threaded rods have star-shaped handles that are removable for travel. The combination of a cylindrical tube and the threaded rod creates a take up reel.



**Figure 10.** Photo of early prototype of the cylindrical tubes

Primarily, I interact with the two star-shaped handles connected to the end of each threaded rod (see 0:50 in the video). I slowly turn the handle in my right hand in order to move the paper across the lights. I keep my left hand on the left handle to control its movement and to sometimes reverse directions. During this performance, I move the paper in both directions (see 6:40 in the video), both slow and fast (see 7:33 in the video), and sometimes I freeze the paper's position over the light bar (see 2:07, 2:23, and 4:47 in the video). In addition to stopping the movement of the paper, I use an expression pedal to actually freeze (read sustain) the sounding audio by pressing the expression pedal with my foot. The movement of the visual composition is also mapped directly to the speed of

the data input from the light bar. So it is clear to an audience that my performative actions are directly controlling the visual and sonic output. Even if it is not clear *how* I am controlling them.

I use a SoftStep 2 foot controller and expression pedal during the performance of this work. I control the Kyma Timeline, turn on and off the live camera feed, and toggle a video effect during this section of the work. The SoftStep 2 works perfectly for this application. The expression pedal is used to freeze the playback of both sound and visuals. I use this form of control as a motivic device. Freezing playback will sustain any sound currently audible. Freezing is similar to holding a note on a wind instrument.

## III.4 CURVES & LINES

A multichannel, real-time, interactive composition for iPad. iPhone, Phonopaper (iOS App), LoopyHD (iOS App), and Dedalus (iOS App).

### INTRODUCTION

This composition is unique in my dissertation because I do not use a computer at any step. Instead, I use an iPhone, iPad, an iTrackDock (iTrack), a SoftStep 2 MIDI foot controller, and a few audio applications on the iPad. And while I realize a semantic argument can be made that the iPhone and iPad are computers, I consider them mobile devices and not computers. The distinction for me is the difference in their capabilities and the tools available for laptop computers and mobile devices. Apple mobile devices are incredibly powerful and have carved their own niche in music production. There are far fewer applications for the iOS platform, but the ones that exist often have very specific, yet advanced capabilities.

Composing an electroacoustic work using only mobile devices was a challenge I set for myself. I was an early adopter of making music with mobile devices. I have been using them for several years in some way. This work illustrates that mobile devices and accompanying hardware and software are legitimate tools for composing electroacoustic music.

Like every other composition in this portfolio, I am able to manipulate pitch, amplitude, timbre, duration, and location effectively during a live performance. In addition to controlling these five characteristics of sound, individual iOS software offers various options for data mapping, scaling, and routing. Therefore, this collection of

mobile hardware and software can be defined as a data-driven instrument. The control input modules, mapping/routing modules, and sound production modules are all embedded in software on the iPad.

This work was inspired, in part, by Daphne Oram's Oramics synthesizer.<sup>34</sup> Oram's synthesizer used clear film that the composer used to draw shapes upon that are interpreted as changes in frequency over time. In *Curves & Lines*, drawings on paper are represented as waveforms that change over time (Appendix C). There are more than a few fundamental differences between Oram's synthesizer and my iPhone. However, the underlying concept is the same. User drawn images of lines and dots, or waveforms in the Oramics synthesizer, are translated to audio. *Curves & Lines* is the second work in this dissertation that relies on a type of computer vision to generate sound.

The fact that I am able to compose an electroacoustic composition by taking photos of my drawings is certainly magical. That I am able to realize this entire composition using only Apple mobile devices and applications is both a result of a personal challenge and a magical feat, to be sure.

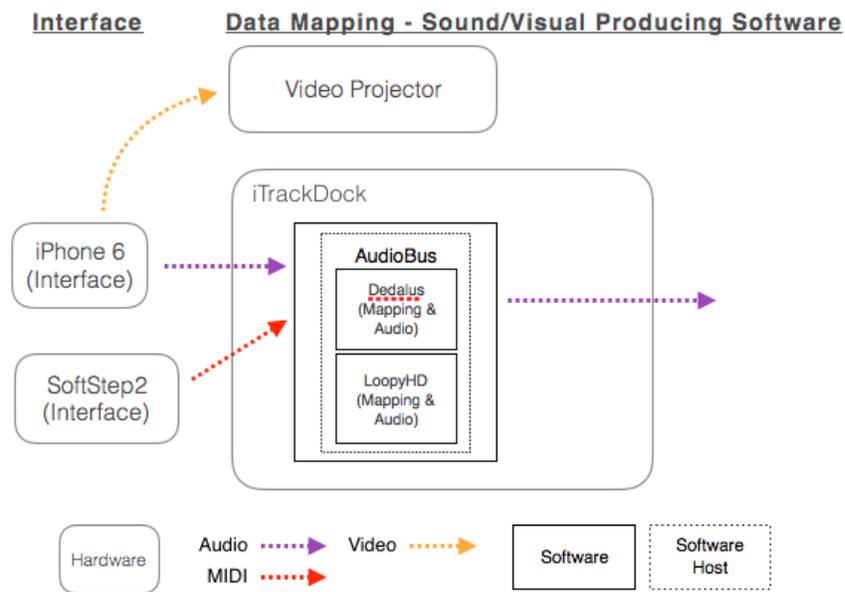
## **INTERFACE, HARDWARE AND SOFTWARE**

The data-driven instrument used in this composition consists of an iPhone 6, iPad Pro, iTrackDock, SoftStep 2, PhonoPaper software, AudioBus 2, Dedalus, and LoopyHD. This data-driven instrument does not include a desktop or laptop computer (Figure 11). The primary interfaces used in this performance includes an iPhone and SoftStep 2 foot

---

<sup>34</sup> Daphne Oram. (n.d.). Oramics. Retrieved from <https://daphneoram.bandcamp.com/releases>

controller. In addition to these hardware modules, the score for this work is a series of drawings of curves, lines, and dots. Performance of this work involves capturing these drawings in software running on the iPhone. The iPhone software translates the drawings to waveforms and outputs as audio from the iPhone’s headphone jack. This output is connected to the audio input of the iTrack. The iTrack has an Apple Lightning connection and provides high quality audio input and output for Apple iOS devices, as well as a single USB input for MIDI controllers. In this performance, the SoftStep 2 is connected to the iTrack via the USB input and provides a means of control over various iOS software applications.



**Figure 11.** Diagram of data-driven instrument for III.4 *Curves & Lines*.

The iPhone connects to a projector through an Apple Lightning-to-VGA adaptor that allows real-time output of the contents of the iPhone screen. In addition to the video adaptor, an 1/8” mini stereo-to-dual 1/4” audio cable connects the iPhone’s headphone

jack to the iTrack inputs. This connection method allows the audio output from the iPhone to be sent to the iPad for manipulation and recording, while also sending video to a projector. Both connections are needed for this work. This connection requires a fairly bulky video adaptor and a second connection to the headphone jack. In addition, I often need to power the iPhone during use. This requires a third cable, connected to the video adaptor, to provide power to the iPhone. During performance, these cables are taped together with small sections of electrical tape to prevent tangling and accidental unplugging of these connections.

Phonopaper is a type of computer vision application and is the primary sound generator in this work. Phonopaper is a movie application that has multiple functions. Phonopaper can record up to ten seconds of audio and create a graphical representation that can be printed on paper. Second, Phonopaper is capable of interpreting these graphical representations as sound. These graphical representations created by this application are similar to QR codes.<sup>35</sup> Phonopaper can read these graphical representations and convert them back to the source audio from which they were made. From the developer's website, "PhonoPaper is based on the Virtual ANS engine - it is a software simulator of the unique Russian synthesizer ANS."<sup>36</sup> This website also provides links to download a one-page visual template for use with Phonopaper. This template has markings for frequency along the Y-axis and time along the X-axis. I used this template for all of my drawings in this work (Appendix D).

---

<sup>35</sup> Dictionary.com. "QR Code | Define QR Code at Dictionary.com." Accessed March 19, 2018. <http://www.dictionary.com/browse/qr-code?s=t>.

<sup>36</sup> Alexander Zolotov. "PhonoPaper on the App Store." iOS APP Store. Last modified 2015. Accessed November 13, 2017. <https://itunes.apple.com/us/app/phonopaper/id865947553?mt=8>.

The process of capturing a drawing can be seen at 0:37, 1:30, and 2:00 in the video. As the playhead in Phonopaper moves over the captured drawing/waveform from left to right, audio is generated in the software. The duration of these audio representations are 10 seconds in length. However, the *speed* control in the software allows for playback rates between 0-200%. This is fixed by the software, but not considered a limitation in this work. The primary screen for Phonopaper is displayed as a smaller picture-in-picture in the lower right corner of the video. This view monitors the live video output of the iPhone and is the image sent to a projector during the performance of this work.

The drawings play in Phonopaper at a pitch center that is directly related to where the curves and lines exist in the still image (see 2:40 in video). Curves and lines appearing closer to the top of the frame (Y-axis) are represented by a higher pitch. Curves and lines appearing closer to the bottom of the Y-axis are played back at a lower pitch. The thickness of the curves and lines is translated into increased amplitude (Appendix D). Phonopaper offers three user operated controls: speed, output, and position. Two of these controls are horizontal faders that are presented as an overlay on the main application screen on the iPhone. The *speed* fader appears from left to right across the top of the screen and controls the playback rate of the waveform currently loaded, while the *output* fader appears from left to right across the bottom of the screen and controls the overall amplitude output from the application. The playback location moves from left to right on a continuous loop, as I control it during performance by tapping on the iPhone screen.

The audio interface used for this composition is the Focusrite iTrackDock.<sup>37</sup> The iTrack is a high quality audio interface that features a Lightning docking station for compatible iPads and iPhones. An Apple iPad Pro is connected to the iTrack via a Lightning extension cable and, together, these two pieces of hardware form the nucleus of sound computation and manipulation, as there are no laptop computers used in this performance.

I use a Keith McMillen SoftStep 2 MIDI foot controller. This foot controller has ten smaller switches that are each assigned to control a specific feature of this performance. The SoftStep 2 can be seen briefly at 4:38 in the smaller picture-in-picture frame. The SoftStep 2 sends MIDI controller data to the iTrack via USB. A Roland Expression pedal is connected to the SoftStep 2 and is used for control over individual track volumes, playback, record, and mixing parameters in LoopyHD. The MIDI controller data from the SoftStep 2 is also sent to Dedalus for control over effect parameters, preset switching, and master volume.

AudioBus is a routing application that provides the software audio connection between the input of the iPad, any combination of applications, and the output of the iPad. AudioBus is the iPad software that makes the necessary connections for this work. During operation, AudioBus acts as the host application where other audio applications are connected. While it is great to make music using a mobile device, it is useless to a performing musician unless those applications can function together. AudioBus provides

---

<sup>37</sup> "iTrackDock." iTrack. May 03, 2016. Accessed June 03, 2017. <https://itrack.focusrite.com/products/itrack-dock>.

this functionality. Audio applications must be compatible with AudioBus, though the AudioBus API is available for free to all iOS developers.<sup>38</sup>

LoopyHD is the last application in the audio chain. LoopyHD is 12-track looping software that offers several basic controls for each recorded track. LoopyHD also allows its controls to be mapped to external controllers. The SoftStep 2 controls every aspect of LoopyHD that I require for this performance, including previous track, next track, play/mute, reverse, record, clear, halve the recording length, double the recording length, and track volume.

## **SOUND AND SYNTHESIS**

The software for this composition is a combination of four iOS applications: Phonopaper, Dedalus, LoopyHD, and AudioBus. Phonopaper operates on the iPhone and is the primary audio generator. Dedalus is an iPad application that processes an incoming audio stream. LoopyHD is an iPad audio application that records the output from Dedalus. LoopyHD can record, play, reverse, record, and clear any of its tracks. This level of control is provided by the SoftStep 2.

Phonopaper generates audio waveforms based on my drawings. This results in sounds that are noise-based with sometimes only a subtle reference to pitch. Y-axis position of my curves, dots, and lines are translated to frequency, and their length along the X-axis of the screen to duration. This method of sound generation seems very

---

<sup>38</sup> AudioBus. "Audiobus Developer Center." Accessed March 19, 2018. <https://developer.audiob.us/>.

primitive at first. However, there are enough controls to manipulate in real time to generate a musical outcome.

Dedalus is the primary sound modifier in this composition. Dedalus performs granular processing by analyzing the incoming audio, and is capable of modifying pitch, grain size, and scrub frequency of the incoming audio in real time. Dedalus also provides control over delay, feedback, filters, and several other parameters that modify the output in real time.

Dedalus also offers a unique combination of delay and feedback. This section of the software includes controls over feedback, overdrive, frequency dampening, and a dry/wet parameter to mix the dry signal with the effected signal. Each of these parameters provide controls that are accessible from the iPad screen. In addition to offering real-time control, Dedalus also saves presets. I use six different presets in this composition. I created each of these presets to have very different sonic outcomes. The presets are named after their timbral characteristics: SparkyNoise, Fizz, Razoroid, Grain, Socks, and Paper. These presets are used in this same order during performance and each provides unique results. Dedalus can be seen in action during the accompanying video at 3:30, 4:45, and 6:00.

SoftStep 2 MIDI data is sent to Dedalus to control to master volume of the software, while other parameters are changed during performance through the screen interface. The granular process in Dedalus has a control for randomness. This parameter changes the output sound from focused to diffuse very quickly. I use this control to adjust

for intensity of the sound output during performance. Diffuse textures are easier to layer with multiple audio recordings.

LoopyHD receives audio from Dedalus (via AudioBus) and records that input into the selected track. Once recorded, I am able to manipulate it in simple ways to modify the overall time of any recorded track. Loopy can be seen in the accompanying video at 2:00. You can watch a track being recorded in LoopyHD at 3:55 in the video.

The synthesis techniques used in this composition include additive and subtractive synthesis. The raw audio from the iPhone is processed through subtractive processes Dedalus, while recording twelve instances of this processed audio is an example of more extreme additive synthesis. The larger rhythmic patterns in this work are created by the layering of multiple sources. Each new track recorded in LoopyHD has a pitch center, rhythmic structure, and rate of playback that is unique from the other tracks. This method of layering is intentional as I build a larger sonic texture over the course of this performance.

## **VISUAL DIMENSION**

The visual dimension in this work is provided by the video output from the iPhone. The Phonopaper application stays active during the entire performance. The image on the iPhone screen is always live on a larger projection screen behind my performance position. My performance setup is made up of two tall tables. One table holds the iPad and the other table holds the line drawings and a small desk lamp to properly illuminate the drawings during performance (Figure 12). The Phonopaper

website claims that the application works equally well in all types of light, but I have found that brighter, whiter light produces better results while capturing my drawings.



**Figure 12.** Performance setup for *Curves & Lines*.

## COMPOSITION

This composition has no definite sections. Even though I am using a very specific process each time I perform this work, the outcome is always very different from one performance to the next. This is intentional as I see this work as a composition of *structured noise*. Because the ambient light and acoustics of a given performance space greatly affects this composition, I must be constantly aware of each new drawing I capture for recording input and playback amplitudes as I layer the twelve tracks together.

The underlying compositional narrative in this work lies in the actual drawings. Each one is different from the next. Some have a series of dots while others include dark lines or curves, and even others include a combination of dots, curves, and lines. This selection of drawings is intentional in this work.

Each drawing is converted to audio before being recorded to another iOS application LoopyHD. LoopyHD allows recording, looping, reversing, panning, and

amplitude controls. The drawings can then be looped in reverse to emphasize the noise aspect of this work. This technique of layering multiple looping audio files is a compositional process that I am very comfortable with. I repeat this process throughout this composition. Because I have control over each individual track in LoopyHD, I am able to create a new section by starting or stopping certain tracks, reversing others, and even adjust the amplitude of others, all in real time.

## **PERFORMANCE**

My performative actions in this work are made up from my process: use the iPhone to capture a drawing, manipulate the resulting audio, record it, and maybe reverse it. I repeat that process multiple times to build up and tear down the overall texture at any given point in this composition. My primary performative actions are adjusting amplitude, rate of playback, and playhead position in real time. I perform all of these action directly on the iPhone screen using Phonopaper. You can see an example of changing amplitude at 1:00, 2:12, and 4:18. You see the rate of playback (*speed* in Phonopaper) changing at 1:34, 2:12, and 4:12. You can observe that I change the playhead position at 1:45, 2:20, and 3:45. These actions are less appreciable during performance without the smaller picture-in-picture view as the visual feedback from Phonopaper is important to translate my small interaction on an iPhone screen to the sounds I am generating.

Capturing the drawings can be a very delicate process, because the quality of the captured drawing is affected by the type of light and amount light used to illuminate the

drawings during performance. I have the best results using a light that is whiter, rather than yellow. This appears to help Phonopaper determine the elements of the drawing from the background. Since Phonopaper translates the drawn objects as sound, it is important to generate an accurate capture in Phonopaper.

Drawings that contain too much connected, vertical whitespace tend to be interpreted by Phonopaper as noise. Therefore, I capture a small, horizontal line at the top of the frame. This technique acts as a visual high-cut filter by removing all of the noise, typically created from whitespace. I stumbled upon this technique accidentally, but ultimately incorporated it into my regular performance techniques for this work.

After the onset of sound in this work, my performative actions are largely split between capturing new drawings with Phonopaper, changing parameters in Dedalus using the screen, and triggering changes in LoopyHD from the SoftStep 2. Building the composition by process allows me freedom to deviate from any of the drawings and use Phonopaper to translate real time video input as audio.

Data mapping strategies are very straightforward. I control amplitude in two ways in this work. First, foot switch #8 on the SoftStep 2 is mapped to the master volume of Dedalus. There are only two values needed for this control — off and on. Because of this need, I am able to use a simple toggle function. Second, I control amplitude of the 12 tracks in LoopyHD with a MIDI expression pedal connected to the SoftStep 2. This expression pedal provides a range of data that is mapped to the individual tracks in LoopyHD, so I can create a more complex mix during performance. The expression pedal independently controls recording and playback volume for each track in LoopyHD. In

addition to this type of simple mapping, which I use for several parameters in this work, I also have many time-variant controller modules activated in Dedalus on various parameters. These controllers and their destinations change in each preset inside Dedalus. I use LFOs at multiple scalings to control effects in real time. In the preset *Razoroid*, the delay parameter is controlled by an LFO generating a square wave, so it has two states. In the preset *Fizz* the delay parameter is controlled by an LFO similar to the previous preset, but also includes an LFO at a higher frequency sine wave that controls the panning of the output sound. I chose a sine wave for panning to provide a smooth change from one extreme to the next. Dedalus also provides simple range scaling and data compression options to customize the incoming MIDI controller data. Each parameter has a user controlled output range. This allows me to send the full MIDI range (0-127) to Dedalus and control the scaling on input — before the control data reaches the desired parameter.

The drawings used in this work represent the best of dozens that were created for experimentation. Using Phonopaper, I am able to capture one drawing at a time and use its visual content as the basis for the sound. The first drawing is only a series of dots. When I capture this drawing I intentionally position the dots higher on the Y-axis in the frame. This results in the dots producing higher, shorter frequencies. After adjusting parameters in Dedalus to create the desired sound, I activate LoopyHD to record the audio. Typically, I record this new audio track for around a minute. I use this length to reduce the number of iterations of a particular track during a performance.

## III.5 EUPHONY

A multichannel, real-time, interactive composition for Wacom tablet and Kyma.

### INTRODUCTION

Similar to other compositions in this dissertation, *Euphony* derives its sound world largely from audio files of spoken text. These audio files of spoken text are synthesized in Kyma using various methods, including the TAU algorithm. Each of these audio files is made up of a series of sentences compiled from a larger work. These new combinations of sentences create a new narrative that implies distorted dreams and recollection. This composition was inspired by a course I attended in recomposition and analysis.

*Euphony* utilizes a simple interface for real-time control — the Wacom Intuos Pro Tablet. This interface provides multiple forms of control that I use to perform the work. This combination of controls allow me to find interesting moments made up of complex waveforms being played back a few milliseconds at a time. The result is a beautiful new timbre.

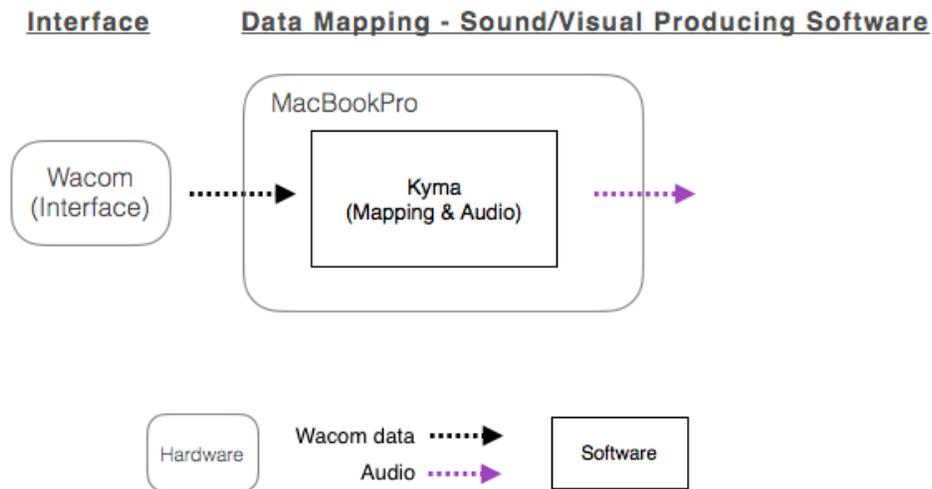
*Euphony* is my attempt at recomposition. In this work, I am combining text from various sources to create an entirely new narrative that speaks to distorted dreams and realities. The text used was extracted from three very different sources — a children's book, a haiku, and poems by Wilhelm Müller from Schubert's *Winterriese*. These sources are carefully combined to create a new result from borrowed words. I use Kyma to record moments of my composition in order playback these segments at later moments in the

work. The recorded segments of the text represents distorted memories. At the end of this work, the distorted voices play together to form a euphonic combination of sounds. These recorded segments are not like their source.

In *Euphony*, there is magic in constructing a narrative from multiple, different sources. Similar to breaking an ancient codex, I constructed the new text using a specific process in order to arrive at a new ending. Finding the special, musical moments in spoken text, and using those moments to create a new narrative is a special sleight of hand.

### INTERFACE, HARDWARE AND SOFTWARE

The data-driven instrument used in this composition consists of a Wacom Intuos Pro pen tablet and Kyma software operating on a MacBook Pro (Figure 13).<sup>39</sup>



**Figure 13.** Diagram of data-driven instrument for III.5 *Euphony*.

<sup>39</sup> Wacom. "Wacom Intuos Pro 2." Wacom Intuos Pro: creative pen tablet | Wacom. Accessed June 03, 2017. <http://www.wacom.com/en-co/products/pen-tablets/intuos-pro>.

The physical interface for this composition is a Wacom Intuos Pro pen tablet. Created by Wacom for visual artists, the pen tablet is a multidimensional controller that offers artists an unparalleled, high-definition control surface. The Intuos Pro tablet has six buttons, a continuous dial, and a built-in wireless interface for connecting to a computer. I do not use any of these features because there is no implementation in Kyma for these buttons and dial.

All Wacom tablets include a pen. The pen is an integral component of this interface as it provides several forms of input. The pen has two ends: a smaller end (nib) used for drawing, and a larger end (eraser) used for erasing. Both ends look like their intended function. The smaller end comes to a fine point like a pencil, while the larger end looks more like a rounded eraser. The pen tablet combination here offers 2048 levels of pressure sensitivity. In addition to the two ends, the pen also houses two unique buttons. In combination, the pen offers multiple, unique data streams.

In this work, I rely on more than one data stream from the pen and tablet at one time. In fact, the generated sound is produced as a result of both the X-axis and Y-axis values at any given moment. In almost all cases, I control pitch of the sound I am performing by moving the pen along the Y-axis of the tablet. I also map the timeindex of an audio file to the X-axis of the tablet. This mapping allows for a left-to-right orientation of the timeindex of the audio file I am performing, while altering the pitch by moving up or down along the Y-axis. This combination of controls is well suited for this performance and can be seen from 1:57 - 2:15 in the video. At times, I combine the X-axis and Y-axis values with a third data stream — the angle of the pen in relation to the

tablet surface (see 1:20 in the video). This allows me to perform subtle modulations to the sound.

The specific tablet used in this work measures 12.6 inches wide by 8 inches tall. The actual sensor area is 6.5 inches wide by 4 inches tall. I chose this tablet because of its compact sensor area. I also tested a tablet that was three times the size of the Intuos Pro. On either tablet, the timeindex of an audio file is placed along the full X-axis of the tablet, regardless of size of the tablet or the length of the audio file. Because the larger tablet has greater X/Y resolution over a larger area it was necessary to perform much larger, more sweeping gestures to produce the smaller motives I desired. Some of my musical phrases would be very difficult to reproduce on the larger tablet. The smaller tablet provides a greater level of accessibility for this work.

The only software needed for this composition is Kyma. Kyma provides the means for the Wacom tablet to connect directly without the need for any intermediate software. I used a Kyma Timeline to organize the sounds as events that unfold over time.

Even though I through-composed each section of this work, I intentionally created moments of pause at each transition to allow a brief moment of improvisation. I utilized the WaitUntil object in Kyma to allow for these brief moments. Some very interesting (to me) music happens in these moments because I am simply exploring the interface and sound as I transition to the next section.

I communicate with the WaitUntil object in two distinct ways. Most of the WaitUntil prototypes in this work are passed by touching the pen to the upper left corner of the tablet. I chose to map this control to that corner because I only use that corner of

the interface for specific sounds, though not often. When the pen breeches a threshold close to the corner of the tablet, the Kyma Timeline will continue. Other times I pass the WaitUntil object by turning the pen over and touching the eraser end to the tablet surface. In both cases, I mask the utility of my actions by generating sound at the same time. Moving the pen to the upper left corner triggers an audio file of a male voice singing “Mein Herz” (also from Schubert’s *Winterreise*). This text can be heard at 2:41, 2:54, and 3:42 in the video. Turning the pen over and touching the eraser to the tablet surface triggers an audio file of modified bell sounds that have a *dreamy* quality. This can be heard at 8:11 in the video.

Kyma provides visual feedback in the form of the Timeline and its markers, the VCS (Virtual Control Surface) a timer, and color coding the sections of the Timeline window. For every section, there is some control I need to monitor. These controls are properly displayed and switched in the Timeline, by the markers. Primarily, I am monitoring my location on the tablet as represented by a feedback control on the VCS. However, I am also monitoring the recording and playback functions in each section.

## **SOUND AND SYNTHESIS**

The sound world for this composition was created in Kyma. Sound synthesis techniques include TAU processing, granular synthesis, and sample manipulation. In addition to synthesis, I am also utilizing the recording abilities of Kyma in order to capture moments of my performance for layering and further manipulation and control.

The sources of the sound design for this composition are primarily spoken text (Appendix E). I chose several non-contiguous lines of text from *Winterreise*, poems by Wilhelm Müller. I recorded myself speaking this text and perform all edits and manipulations in Kyma. I also used an audio recording of a nine-year-old girl. In addition to these sources, I borrowed text from A.A. Milne's *Where the Wind Blows* and *Summer Midnite* by Sazaria Quarles. I chose this combination because it created a new, dreamlike narrative.

Since this work draws upon accounts of distorted dreams and dreamlike states, I incorporate Kyma's MemoryWriter object to capture, in real time, the performance of these voices and their rich timbres to be used at a later point in the composition as disfigured memories. These recordings are then modified in specific ways to create the distorted memory/dream sounds. These audio modifiers include harmonic resonators, audio delay, granular reverb, and reverb. Inside of each of these modifiers are parameters that are controlled in real time using the Capytalk language.<sup>40</sup> One of the recordings plays through its timeindex from start to finish while another plays back and forth as a palindrome. Some modifiers have a Capytalk expression that performs sweeps through a range of frequencies. These recorded representations are vastly different from the version presented in real time. The modifiers and processes used for these sounds are not duplicated at any other time in this work. The types of modifications, the modulation of their parameters, and the spatial placement of these sounds all reinforce the song's narrative of twisted dreams and reality.

---

<sup>40</sup> Symbolic Sound. "Kyma•tweaky: Learn Capytalk." Accessed March 19, 2018. <http://www.symbolicsound.com/cgi-bin/bin/view/Learn/Capytalk>.

In addition to the audio files of spoken text, I also incorporate elements of Schubert's *Winterreise* by recording prelude and postlude material from each of the movements I borrowed text. These preludes and postludes are slowed to only a few percent of their original speed. However, they are retained at the original pitch, minus two octaves. This shift down offers a lush texture in which its source is directly connected to the text, but smeared in a way that provides a generous amount of low-frequency content as an underlying foundation for the composition. This work is a blend of all these elements and is meant to further explore the narrative of interrupted dreams.

## **COMPOSITION**

Euphony is made up of definite sections. Each section is a progression of my compositional process. Contrary to the title of this work, there is more dissonance than euphonic harmony. This composition features several audio files of spoken text from various sources. Each of these audio files provides a smaller piece of the larger narrative in this work of distorted dream states.

Each section is composed of spoken text and a low frequency texture. I control the spoken text, in real time, through my performative actions on the Wacom tablet. The low frequencies play automatically and provide a harmonic structure for the spoken text. Each of the low pedal sounds are taken directly from recordings of prelude and postlude material from *Winterreise*.

The new texts used in this composition are formed from various poems by multiple authors. The selection of these texts were intentional to create a new narrative for this composition (Table 2).

**Table 2.** Newly formed text used in III.5 *Euphony*.

New formed text from <i>Euphony</i>
All the birds began to sing
How to find a way out? I'm used to going astray. And every sorrow to its grave. I have found shelter.
Warm midnight falling, stars shinning, dancing brightly, peaceful all at once.
I dreamed of many colored flowers. The way they bloom in may. I dream of green meadows — of merry bird calls.

The distorted dream state narrative is apparent in the selection of texts. The narrative is reinforced by short audio recordings made from the spoken text. These recordings present each spoken text audio file in a heavily modified version. Each of the audio recordings has similar audio modifications applied. These modifications include pitch, amplitude, timbre, duration, and spatial location.

Each section builds in intensity by adding new layers and creating new recordings. All four recordings come together in the final section without their unaltered sources. The result is a wash of sound produced by the modified versions of the recordings. The recordings are mixed together in real time and create a texture of heavily modified sounds meant to represent memories of distorted dreams.

The selection and order of texts in this work creates both a new narrative and opportunities for performative exploration. I use the technique of text-painting at various times in the performance of these recordings. Text-painting is the treatment of sung lyrics

accompanied by an emphasis on a particular aspect of a word to strengthen its meaning. At 2:10 in the video I use this technique on the word *grave* by lowering the pitch of this word in real time through my performative movements. The reference is that graves are under, or below us. Therefore, I lower the pitch of this word to further emphasize its meaning.

I also trigger the words, “Of merry bird calls” several times in immediate repetition. Because I map the Y-axis position of the tablet to the pitch of my sounds, I am able to repeat this phrase over and over at different pitches. The result (to me) sounds like a chorus of mocking birds at various pitches. This can be observed from 7:20 - 7:40 in the video.

All sounds in this composition are dry (no reverb), and I only apply reverb at specific times during performance. Primarily, I use reverb to create a drastic change in ambience to emphasize certain words or phrases. At 7:25 in the video, I add reverb to resultant sound to emphasize the words, “The way they bloom in May.” Generally speaking, I use pitch to emphasize words that imply lower or becoming smaller, while I use reverb and ambience to emphasize words that imply higher or becoming larger.

## **PERFORMANCE**

Based on my mapping of pitch and time, playing at the bottom of the tablet will generate low frequencies (< 30 Hz) in the lower left corner of the tablet. Because the audio files I am performing with during this work are generated from the TAU player in Kyma, I am able to manipulate time and pitch independently. When the pen is pressed

into the bottom left corner I am able to trigger the very first onset of sound in the spoken word audio file, at the lowest pitch possible. Keeping the pen in the same X-axis position and moving along the Y-axis a few millimeters produces a rich, deep clicking sound. I can also enable reverb for the audio files (typically a very dry sound) by pressing the upper button (pen-button-2) on the pen (see 0:30 and 3:35 in the video). This reverb is connected in the signal flow — **after** — the recorder, so the particular reverb effect that I enable is only a real-time effect and not part of the recordings. The recordings use unique reverb settings when played back at the end of the composition. In addition to reverb, the angle of the pen is mapped to a parameter that sometimes controls the formant control of a harmonic resonator and other times controls a lowpass filter used to help better articulate the voices. My angle-based movements can be seen at 1:20, 1:34, and 5:14 in the video.

I trigger audio recordings of guitar harmonics by using the eraser end of the pen on the tablet (see 3:02 and 3:48 in the video). Contrary to the primary mapping scheme of this work, I chose to map the pitch of these harmonics to the X-axis of the tablet (see 3:02 in the video). I did this because I was only altering one parameter in real time and mapping it to the X-axis places my movements in a left to right orientation to the audience. This movement is more easily perceptible and appreciable by an audience.

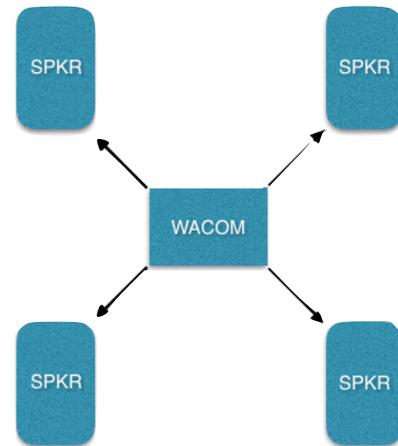
I use the pen at the very left edge of the tablet to trigger the first few milliseconds of an audio file that is assigned to the X-axis of the tablet. I start the sound by placing the pen in the lower left corner of the tablet. However, my performative actions are a series of movements along the Y-axis that will change the frequency of the resultant sound. I do

this several times, each time going further along the Y-axis. The result is a rising pitch that builds to an ultimate climax of both pitch and amplitude. The result of this section always reminds me of a *call to prayer* because the resultant sound is a melismatic motif that sounds like a double reed instrument. With the addition of a filter this emulates a vocal wailing timbre (see 6:20 in the video). I used a Capytalk expression in Kyma to increase the amplitude of this audio file (also along the Y-axis), exponentially. This simple technique provides a climactic swell of amplitude at the highest notes.

At one point later in the work, there is one long audio file of newly formed text situated along the X-axis from left to right speaking the text mentioned previously. However, this same audio file is also situated along the X-axis from right to left. I set the relative amplitudes of each iteration (both directions) so that when played from left to right the text was easily discernible, but when playing from right to left the backward version is more prominent with little outbursts of text popping through the mashed sounds of the reversed audio file (see 7:40 in the video).

The final section contains the four recordings previously captured. These sounds are mapped to the four corners of the tablet. The tablet functions as a four-way audio fader that is controlled by the pen location. When I place the pen in one of the extreme corners, only one recording is heard. When the pen is anywhere else on the surface, the sound produced will be a mix of all four sounds at a relationship that is equal to the distance of the pen tip to any given corner of the tablet. When the pen is placed in the center of the tablet all four recordings will be played at equal amplitude. This mapping allows for a high degree of control over the resultant sound. In addition to mapping these

sounds — physically to the four corners of the tablet, these sounds are also spatialized to their own corner of a four channel speaker system because the physical position on the tablet aligns with the speaker mappings in the performance space (Figure 14). I used this similarity to spatialize the four recordings from the audience perspective.



**Figure 14.** Mapping the four corners of a Wacom tablet to align with a quadraphonic speaker system.

When I place the pen in the lower left corner, I trigger only the first recording (*all the birds began to sing*) in the front right speaker, which is the closest speaker to the corner of the tablet I am touching. As I move the pen to the lower right corner, I crossfade to the second recording (*how to find a way out...*). *This second recording plays only from the front left speaker, which is closest to the corner of the tablet I am touching.* Moving the pen to the upper right corner, I crossfade to the third recording (*warm midnight falling...*). The third recording plays only from the left-rear speaker, which is the closest to the corner of the tablet I am touching. Moving the pen to the upper left corner I crossfade to the fourth recording (*I dreamed of many colored flowers...*). The fourth

recording plays only from the right-rear speaker, which is closest to the corner of the tablet I am touching. From an audience perspective, the four corners of the tablet represent the four speakers in the performative space. As I draw a large rectangle on the tablet's edge, the sonic output is always one sound crossfading to a second. To end this work, I use larger, circular gestures on the tablet to radically move the sounds in and out as the final sounds fade to silence (starts at 9:20 in the video).

Most of the sounds used in this work are triggered, or activated, by the physical act of touching the pen tip to the tablet. The location of the pen on the tablet surface is transmitted as an X/Y coordinate pair to Kyma. In all cases, the Y-axis data controls the frequency of the audio file or synthesis, while the X-axis data controls the timeindex. As I draw the pen from left to right, the current voice audio file progresses through its timeindex. As I move the pen up or down on the surface, the result is an analogous change in pitch. All of the voice audio files are performed in this way, with similar mapping frameworks.

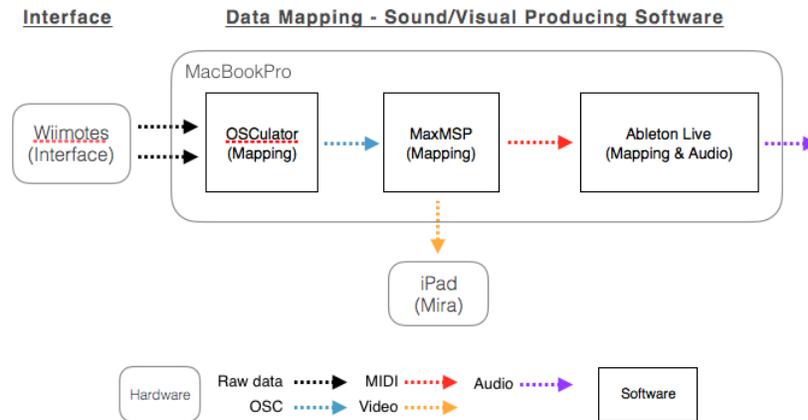
### III.6 THE INSISTENCE OF SAMENESS (IS)

A real-time, multichannel composition for Two Wii Remotes, Max/MSP, and Ableton Live.

#### INTRODUCTION

My composition, The Insistence of Sameness (IS) was composed with the intent to encourage a dialogue about autism. The sounds used are of two major types: chordal pads and intrusive noises. The juxtaposition of these two types of sounds is analogous to the *noise* inside the mind of many people on the Autism Spectrum.

The data-driven instrument used in this work is a set of Nintendo Wii Remotes (Wiimotes), a software mapping layer in Max/MSP, and a sound producing layer in Ableton Live (Figure 15). Wiimotes are video game controllers created by Nintendo for use with their game consoles. The Wiimotes are slightly larger than palm-sized and fit comfortably in a performer's hand, and effectively become extensions of a performer. Each Wiimote is capable of producing fourteen unique data streams in the form of buttons, triggers, and faders for control over sound.



**Figure 15.** Diagram of data-driven instrument for III.6 *Insistence of Sameness (IS)*

The sound synthesis is a combination of audio file playback and manipulation, granular synthesis, and audio processing through various VST and Max for Live devices.<sup>41</sup> <sup>42</sup> Spatialization is sometimes automated through several different Max for Live devices, and other times controlled by my angling of the Wiimote (see 3:10 and 7:38 in the video). However, pitch, amplitude, timbre and duration are always controlled by me in real time.

The printed score for this composition is a visual representation I created that includes section numbers, movement suggestions for each of the Wiimotes, and timing for each section (Appendix F).

My performative actions in this work are meant to replace the (sometimes) erratic thinking and irregular body movements referred to as *stims* by the Autistic community.<sup>43</sup> The Wiimotes generate multiple data streams and offer almost endless combinations of button and accelerometer values for use in a live performance.

Mapping my performative actions I am able to control sounds with very specific, yet exaggerated movements. This method of mapping is necessary to support the underlying narrative of this work. Translating my movements to sound using video game controllers is its own brand of magic.

---

<sup>41</sup> VST — Virtual studio technology (VST) is a digital interface standard that is used to connect and integrate software audio effects, synthesizers and effect plugins with recording systems and audio editors.

<sup>42</sup> "Max for Live | Ableton." Accessed March 19, 2018. <https://www.ableton.com/en/live/max-for-live/>.

<sup>43</sup> "Stimming." Autism Wiki. Accessed June 03, 2017. <http://autism.wikia.com/wiki/Stimming>.

## INTERFACE, HARDWARE AND SOFTWARE

The Wiimote is a video game controller for the Nintendo Wii video game system.<sup>44</sup> The Wiimote features a total of eleven buttons of various sizes and shapes, 3-axis accelerometer, and a vibrating transducer. Although the Wiimote has a port for connecting peripherals, none were used in this performance. The Wiimote houses an infrared sensor in the tip. This sensor was also not used in this performance. The controller is powered by two AA batteries and connects to a computer via Bluetooth wireless communication.

Because the controller was built by Nintendo and intend for use with their game consoles, a helper application is needed to maintain a connection and map the incoming data streams. I used OSCulator for this purpose. OSCulator is a third party application that facilitates the simultaneous connection of multiple Wiimotes (and various other game controllers). OSCulator both provides the connection to the Wiimotes as well as basic data mapping and routing.

OSCulator allows a connection from the Wiimotes to the computer via Bluetooth, and also provides a basic level of data routing. Ableton Live is the only sound producing mechanism. The Ableton Live session contains twelve tracks. There are four tracks dedicated to chordal pad sounds, four tracks dedicated to transitional sounds, and four tracks dedicated to noise sounds.

The raw data received by OSCulator is in the range of 0-1. This incoming data range is passed along to Max/MSP via the OSC protocol. In Max/MSP, the incoming

---

<sup>44</sup> "Wii U - Accessories." Nintendo. Accessed June 03, 2017. <http://www.nintendo.com/wiiu/accessories>.

range of 0-1 is scaled to the MIDI range of 0-127. All data mapping, scaling, and routing in this work is controlled in Max/MSP. The data mapping profile in this work is really very simple. All incoming Wiimote data is scaled to 0-127, packaged as MIDI controller data, and output over MIDI from Max/MSP to Ableton Live.

Max/MSP plays an important role in this work because it is the intermediary between all software. Max/MSP is responsible for receiving the input from OSCulator, scaling that input from 0-1 to 0-127, formats that data as a MIDI controller message, provides visual feedback for the performer, and sends MIDI data to Ableton Live to provide ongoing control over sound.

All data mapping and scaling is performed in Max/MSP. There are specific mapping and scaling operations for each data stream from each Wiimote. For instance, the X-axis data from the Wiimote controls spatial panning of one sound and the frequency parameter of a frequency-shifter device for another. This control is scaled from 0-1 to 0-90 of the MIDI range. The X-axis is also sent to the playback position of the first pad sound. In this case, the X-axis is scaled from 0-1 to 0-70 because that was the necessary range for this particular sound, in this performance, based on my performative actions. There are slightly different scaling values for each data stream before being sent to Ableton.

The Wiimote's functionality is well suited for this composition. Buttons are used for starting and stopping sound events, while the accelerometer is used (in both X- and Y-axes) to provide ongoing control over sound events. In addition to starting and stopping sounds, the buttons are used for a variety of other purposes: navigation, harmonic scale

modulation, plugin effect mode changes, initiating random processes, and starting and stopping timers. I intentionally mapped the buttons to specific controls. For instance, the button that selects the combination of sounds for a given section (UP on the four-way button cluster), is the farthest from my reach because if I inadvertently hit that button at the wrong time, the outcome will be very noticeable to an audience. By contrast to that mapping, the buttons that trigger the onset of sound (both A and B buttons on both Wiimotes) are the two buttons that are the easiest to reach. This mapping is intentional as I use these buttons more than any of the others.

The plus and minus (+/-) buttons on the left Wiimote advance the current view in Ableton. Using these buttons allow me to focus on a specific track (in Live) for visual feedback during performance. I mapped both of these buttons to control the currently selected track in Ableton Live. Pressing the plus button moves the focus to the next track to the right, while pressing the minus button moves the focus to the next track to the left. This level of control is important to this work because it allows me to observe various processes during performance.

This Max/MSP package is well organized. I created a menu from button that appear horizontally across the top edge of the Max/MSP window. These buttons will open up windows to control various aspects of the software. These windows include, MIDI CC, Mapping, Structure, Time, and Mira. MIDI CC is a helper window that lists all MIDI controllers used in this work, and their destinations. Mapping is a small helper that opens a window and provides an interface for assigning new controller values in Ableton.

Structure opens a window that contains the code needed for advancing the sections of the

composition. Time opens a window that has the code needed to keep track of duration of each section during performance. Mira opens a window that includes the code needed to display a basic amount of feedback on an iPad screen. Mira is a feature of Max/MSP that allows for the displaying of Max/MSP interface controls on an iPad. I use Mira for visual feedback of each control (Appendix G).

## **SOUND AND SYNTHESIS**

Aside from audio file playback, the primary synthesis in this work is subtractive. Each of the sounds are created from some type of audio file, and each audio file is modified through a series of filters. Some of these filters are controlled in real time during performance. Among these filters are low-pass, high-pass, band-pass, band-reject, frequency-shifter, reverb, and delay.

There are a total twelve audio tracks in the Ableton Live project file I use for this performance. The breakdown of the tracks is as follows: three pads, four variations of noise, and four rhythmic loops. The first pad sound is created from the Granulator II Max for Live device mentioned previously in this dissertation. This Max for Live device plays a recording of jungle birds recorded in Belize at Altun Ha.<sup>45</sup> I control this sound in real time by rotating the left Wiimote on its X-axis. This action moves the playback window inside this Max for Live device. The second pad sound features several different audio recordings taken from Altun Ha. These audio recordings are loaded into a software synthesizer. In this synthesizer, I use spectral editing tools to erase and modify parts of

---

<sup>45</sup> TravelBelize.com. "Altun Ha | Belize | Travel To Belize | Ruins Of Belize | Jade Head —." Accessed March 19, 2018. <http://altunha.com/>.

each sound. This technique results in a rich sound that would be otherwise difficult to reproduce with another method of synthesis.

The noise sounds in this work are derived from various sources, including a small wood ratchet recorded on an excursion in Crater Lake, a 3D printer during operation, operational sounds of a Lubitel 166+ camera, and bird sounds recorded at Altun Ha.<sup>46 47</sup> These sounds are meant to metaphorically represent chaos and noise. I used filters to further sculpt these sounds to create a palette of audio files that work well together. The wood ratchet sound is played from a Sampler, the 3D printer and bird sounds are performed from a Granulator II device, and the camera sounds are performed in a Multisample device in Ableton Live. Each of these processes were chosen based on the desired output. The Multisample device in Ableton was the best tool for playing back a collection of audio files in succession, with real-time control.

## **COMPOSITION**

The chordal pads are composed from audio recording of jungle birds from the rainforest in Belize. The noise sounds are derived from multiple white noise audio files, a collection of camera sounds, jungle birds, and a Makerbot 3D printer during its operation. There are also bell sounds, flute-like sounds, short rhythmic loops, water

---

<sup>46</sup> National Park Service. "2015 Artist in Residence - Crater Lake Science and Learning Center (U.S. National Park Service)." Last modified 2015. Accessed March 19, 2018. <https://www.nps.gov/rlc/craterlake/artist-in-residence.htm>.

<sup>47</sup> Soviet Cams. "Soviet and Russian Cameras - Lubitel-166." Accessed March 19, 2018. <http://www.sovietcams.com/index.php?2022299833>.

sounds from Zion Canyon National Park, and lush string sounds.<sup>48</sup> These chordal pads can be heard at the 2:00 mark in the video.

The overall structure of this work is fluid. However, there are seven, definite sections. The composition begins with a slow, crackling sound of white noise through a filter. This sound creates a context for the work. Shortly after introducing the noise sound, I add a pad sound under the noise. The pitches of the pad sound are controlled by a table in Max/MSP. A table is similar to a spreadsheet — it contains rows and columns and can store a large amount of data. In this case, there are several tables that each hold the values of different scales. The scales I use are D Major, A Major, E Major, G Major, F# Major, C# Major, and D Major an octave below. I chose these scales because I like their sonority and they all overlap in interesting ways. This selection of scales provides a harmonic movement in the pad sounds. Since I trigger the onset of the pad sounds by pushing a button, I am able to push several times to generate more than one pitch at a time. This creates a larger, thicker texture and helps to build the tension in the composition.

Each section is a combination of noise sounds and lush pad sounds. This juxtaposition of sonic textures is my way of representing my personal experience with someone on the autism spectrum. From my point of view there is a duality that exists in this particular individual. From my point of view, the duality of harmony and dissonance can be overwhelming to balance for a person on the spectrum. Achieving balance is a balancing act — a battle of sorts. External influences can shift the balance one way or

---

<sup>48</sup> Expedia. "Visit Temple of the Sinawava in Springdale | Expedia." Accessed March 19, 2018. <https://www.expedia.com/Temple-Of-The-Sinawava-Springdale.d6298730.Vacation-Attraction>.

another. Sometimes, excessive good emotions can lead to even more excessive bad emotions, and often the mood swings are like that of a pendulum. Often, the transition from high to low (in terms of mood and demeanor) can happen in an instant.

My goal in this work is to present various combinations of harmony and dissonance that overlap one another in various ways. At times, the dissonance cuts through the harmony for a brief, painful moment. Other times, it is the harmony that must break through the dissonance. I make every effort in the performance of this work to emulate the experience of being on the spectrum, from my limited perspective.

## **VISUAL SCORE**

The score for this composition is a series of color-coded lines that extend from left to right on a landscape oriented page. The instructions for the left hand occupy the top half of each page, while the instructions for the right hand occupy only the bottom half of each page. The instructions are a set of lines and graphics that indicate which button to press and which way to angle the Wiimotes to generate sound. These instructions indicate the primary form of interaction for a given section through a series of graphic symbols (Figure 16). These instructions are correlated with the accelerometer axes.

The start of each section is marked in the score by a large number (1-7). There are a total of seven sections. The markings within each section are horizontal lines that represent time — no specific amount of time, rather a ratio compared to the surrounding sections. The score shows the relationships between the active sounds.



**Figure 16.** Examples of visual instructions from the score

## **PERFORMANCE**

My performative actions in this work are exaggerated physical gestures that seem to create the matching frenetic source of sounds. My left hand generates the onset and ongoing control over the pad sounds (see 1:30 in the video). My right hand generates the onset and ongoing control over the noise sounds (see 2:42 in the video). I separate these sound in this manner because I want an audience to have expectations for the sounds generated from each hand/controller.

To further separate the sounds generated by the two Wiimotes, I map all noise sounds to my right Wiimote and generate sound by pressing button ‘A’. I provide ongoing control over the sound by moving the Wiimote up and down along the Y-axis. The Y-axis data is controlling the cutoff frequency of a filter, so waving it along the Y-axis produces an obvious change to the audio. I map all chordal pad sounds to my left Wiimote and generate sound by pressing the ‘A’ button, like the right Wiimote. However, I use the X-axis of the left Wiimote to provide ongoing control over the sound. By clearly mapping

specific movements to specific sounds, I help the audience to set expectations for the sonic result of my movements.

In all cases, each controller functions in parallel with one another. However, both controllers share the global controls such as: scale, section, timer start/stop, track focus, and window size of a granular synth where applicable. I set up the controllers in this way to share the duties between hands/controllers. I do not want to perform all of the *administrative* functions with one hand. Splitting these functions across two controllers allows me to control all necessary aspects without being too obvious that I'm performing non sound producing movements.

Some of my performative actions are not necessary to produce sound. Rather, they are choreographed to accompany the emotion behind the generated sounds. In performing these actions, I am helping the audience connect my performative actions, the resultant sounds produced, and the narrative of this work in terms of advocacy. I could produce some of these sounds by performing very small movements with the Wiimotes. However, the larger physical gestures in my performance of this work are important to help guide the emotion of the performance and set expectations for the audience.

### III.7 SUB-ROSA

A real-time, multichannel, interactive composition for GEMS interfaces, iPad Mini 4, Max/MSP, and Kyma.

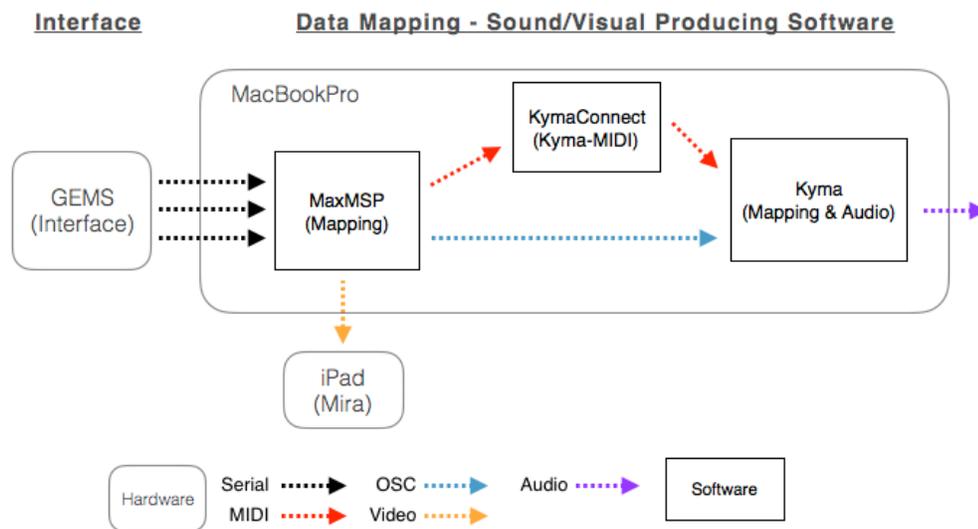
#### INTRODUCTION

Sub-Rosa is Latin for *Under the Rose*, a phrase that has origins in both European and biblical histories. In this work, I pursue the narrative of underlying messages and their meanings.

The data-driven instrument I use in this performance is a set of three physical interfaces, an iPad Mini 4, a mapping layer in Max/MSP, and sound synthesis in Kyma (Figure 17). The smaller interfaces take the physical form of a Gem, Geode, and Cone (see 0:45 in the video). Each of these interfaces require a unique form of interaction to produce data in real time. I designed and created these interfaces for this work. In addition to these interfaces, I use an iPad Mini 4 to control basic mixing functions (see 1:00 in the video) using a function of Max/MSP, Mira. All three interfaces and the iPad fit onto a vinyl surface I created for this performance. This surface has various graphic elements that aid the visual presentation of the three interfaces. This vinyl surface can be seen at 1:27 in the video.

The structure of this work is not the same from performance to performance. A flexible structural arrangement is intentional and provides musical variety. This entire work can be considered as four sections of structured improvisation. Each section overlaps with the next in various ways to create a fluid arrangement.

Max/MSP is the primary software between the laptop and the interfaces. Max/MSP handles all data acquisition, scaling, mapping, and routing for this work. Kyma is the only sound synthesis environment used. Unlike other works in this dissertation that use the Kyma Timeline, this work uses the Kyma MultiGrid to arrange the combination of sounds heard at any point during the performance of this work.



**Figure 17.** Diagram of data-driven instrument for III.7 *Sub-Rosa*.

The visual component in this work is generated from the sight of me manipulating the three interfaces on the vinyl surface. Each interface contains LED lights that illuminate the interface from the inside. Each interface is represented by a different color. The Gem glows green, the Geode glows red, and the Cone glows blue. The Cone is the only interface whose data is mapped to the brightness of its LEDs.

I attempt to create a magical narrative in this work through the use of the Fibonacci sequence. I use the relationship of the first several members of the sequence as pitch relationships in this composition. Translating the relationship of the start of the

sequence to pitch relationships is my attempt at creating an environment of magic in this composition.

## **INTERFACE, HARDWARE AND SOFTWARE**

The three GEMS interfaces for this work are custom-made, live performance controllers designed and realized by me using Adobe Illustrator, Blender, and a 3D printer.<sup>49</sup> <sup>50</sup> The GEMS take the form of a multisided Gem, a Geode, and a Cone. These controllers are slightly larger than hand-sized. Each GEMS interface has a unique shape and, therefore, requires a unique form of interaction. The multisided Gem can be tilted on its ends or can be rotated horizontally to one of its ten facets. Each facet causes the Gem to sit at a slightly different angle and generates a unique trigger value. The Geode rotates 360° degrees, horizontally, and looks like a Geode from the top, but functions as a simple potentiometer. The Cone rotates 360° when on its side, and its rotation can be interrupted and reversed in real time by the performer. Each of these interfaces can produce multiple individual data streams. However, for this work I utilize only a single data stream from each controller.

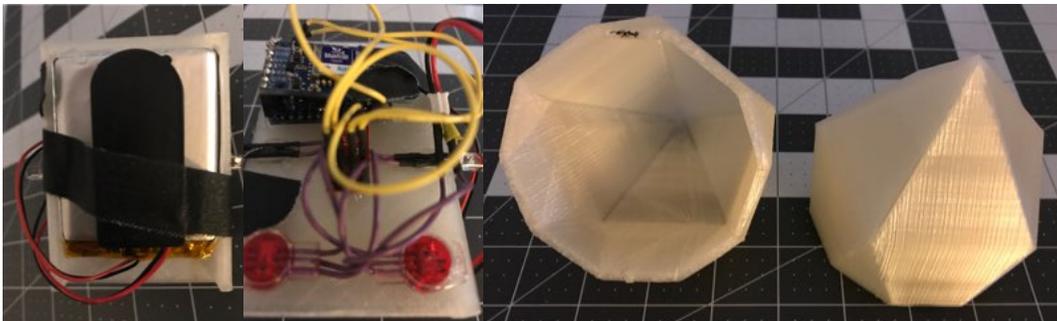
All three interfaces house identical electrical components: an Adafruit Trinket Pro 5v, a 9DoF sensor, a wireless communication board using Bluetooth protocol, a 3.7v battery, and several LED lights to illuminate the inside of the interface.

---

<sup>49</sup> Blender. “Blender.org - Home of the Blender Project - Free and Open 3D Creation Software.” Accessed March 19, 2018. <https://www.blender.org/>.

<sup>50</sup> Geodesic Electronic Movement Sensors. These Custom interfaces were created by Steve Joslin for the performance of *Sub-Rosa*.

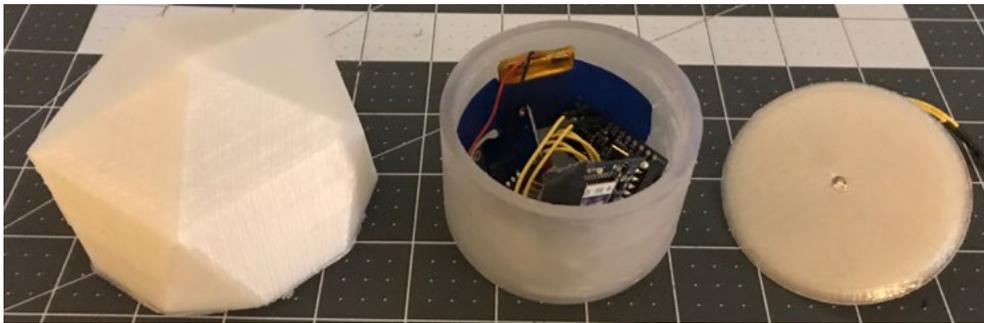
The Gem's facets act as a series of ten switches. Any of these switches can be activated by rotating the Gem. I designed the Gem in Blender and printed it using a Mod- t 3D printer. Since my 3D printer has a limited build size, I was forced to cut the 3D model of the Gem in half to print it in two parts (Figure 9). The result is a large, empty compartment inside of the Gem. I also designed and printed an internal part for this interface (Figure 18). This internal part holds the battery in place on one side and the remaining components on the other side. This new part is removable for servicing the various components and is sized to allow both halves of the outer Gem to come together and be fully concealed inside.



**Figure 18.** Photos of Gem parts.

The two outer pieces of the Gem are held together by white electrical tape. The white tape blends well with the color of the Gem. Mounting the internal components in the center of the internal space of the Gem also prevents blind spots from the LEDs. If anything were attached to the inside wall of the Gem, it would occlude the light from the two internal, green LEDs. Mounting all components in the center of the Gem prevents this. The Gem captures the X-axis position and that number is mapped to the MIDI range of 0-127 before being transmitted to the laptop.

The Geode sits on a round, cylindrical base. The top portion that the performer interacts with is shaped like a Geode. The round bottom allows for the storage of electronics and smooth movement while spinning. The Geode acts as a potentiometer. I designed the Geode in Blender and printed its parts using a Mod-t 3D printer. The Geode is made from three basic parts: base, plate, and the top (Figure 19). The base is round and roughly the size of a spray paint can top. The base holds the electronic components. The plate is a thin disk with a diameter that is the same as the base. The plate has a small hole in the center that holds a red led that illuminates the Geode's top. The top is the most visible part of this interface, sits on top of the base and plate, and is the part that I physically interact with the most. The Geode uses the magnetometer on the 9DoF sensor to determine its position in relation to magnetic north. This data is expressed in degrees between 0-359°. This range is mapped to the MIDI range of 0-127 before being sent to the laptop.



**Figure 19.** Photos of Geode parts.

The Cone is flat on the bottom and comes to a sharp point on the other end. When turned on its side, the Cone spins to generate data. The Cone also acts as a potentiometer.

I designed the Cone in Blender and printed its parts using a FormLabs2 3D printer.<sup>51</sup> The Cone has a different look to it, compared to the Gem and the Geode, because it was printed using different materials and a different process. The Cone has a smoother, more transparent texture. The Cone consists of two parts: a round platform base, and a cone shaped top (Figure 20). All of the electronic components are attached to the platform base. There are six blue LEDs that outline the inner edge of the platform base. These LED lights illuminate the inside of the cone and are visible through the side of the cone during performance. The Cone generates data using the Y-axis data from the 9DoF sensor and maps it to the MIDI range of 0-127 before transmitting to the laptop.



**Figure 20.** Photos of Cone parts.

In addition to the three interfaces, I also created a vinyl surface for performing this particular composition. I designed this surface in Adobe Illustrator and printed it on vinyl on a professional printer. This surface measures 18 x 24 inches and has a graphic element/shape that represents each GEM interface (Appendix H). Each interface is intended to be placed on the surface in its specific location for performance. The Gem is

---

<sup>51</sup> Formlabs. "Form 2 | Formlabs." Accessed March 20, 2018. [https://formlabs.com/store/us/form-2/?utm\\_source=google&utm\\_campaign=GP\\_US-NA-Purchase-Search\\_Brand-Trademark-EN-Exact-Paid-Adwords&utm\\_term=formlabs\\_3d\\_printers&utm\\_medium=cpc&gclid=CjwKCAjw4sLVBRAIEiwASblR-y5I3VqW3mmx0gvLbO9xDUdiigQd7CLcxrt01GNKkqbS-0ii](https://formlabs.com/store/us/form-2/?utm_source=google&utm_campaign=GP_US-NA-Purchase-Search_Brand-Trademark-EN-Exact-Paid-Adwords&utm_term=formlabs_3d_printers&utm_medium=cpc&gclid=CjwKCAjw4sLVBRAIEiwASblR-y5I3VqW3mmx0gvLbO9xDUdiigQd7CLcxrt01GNKkqbS-0ii).

initially located in the upper left corner, the Geode is initially located in the upper right corner, and the Cone is initially located in the center portion of the surface. My performative actions for the Gem and the Geode are similar in that both interfaces spin on a single axis. The Cone rotates around the center of a much larger circle. The diameter of this circle is determined by the height of the Cone since this interface is on its side during performance. The graphic design on the surface reflects the basic shapes of the interfaces and the colors associated with each interface's LED lights. There is also a large rose printed under the section reserved for the Cone. This is intended to be a subtle reference to the title of this composition.

Each interface sends an integer value over a Bluetooth Serial connection. The first point of contact with the interfaces is Max/MSP. Max/MSP receives the data from the Adafruit Trinket. The data coming from each Trinket is scaled to the MIDI range of 0-127 before transmitting to Max/MSP. Max/MSP packages this data as MIDI controller values before forwarding the data to Kyma.

The Gem has a unique block of software that is used to lock in the trigger values for each facet. The problem I faced with this sensor is that the values for each facet tend to drift sometimes by a variation of three whole numbers. This variation prevents consistent operation. I solved this problem by creating a simple calibration patch in my Max/MSP software (Appendix I). This calibration process involves turning the Gem onto each of its facets, recording that value in Max, and filtering out values that interfere. The entire process takes just a few minutes during setup.

In addition to processing the input from each interface, Max/MSP uses that data to generate MIDI controller values and forwards MIDI data to Kyma to control sound selection and amplitude of each sound in real time. I use a feature in Max/MSP called Mira to accomplish this (Appendix J). Mira allows me to create a simple user interface in Max/MSP and display it on an iPad. I use an iPad Mini 4 and Mira during the performance of this work.

For this work, I decided to use a Kyma MultiGrid rather than a Kyma Timeline.<sup>52</sup> A MultiGrid is not in a time-domain. A MultiGrid is a series of tracks that can hold multiple sounds. Each track is laid out vertically and can hold as many sounds as needed, but can only play one at a time. In this mode, Kyma behaves very much like Ableton Live. Each track plays a single audio file at a time and has various controls to adjust for fade in/out times. Each track is set to crossfade seven-seconds for seamless playback.

## **SOUND AND SYNTHESIS**

Kyma is the only sound synthesis environment used in *Sub-Rosa*. The techniques used here include granular synthesis, multi-sample playback, oscillators driven by math functions, analysis and resynthesis, and cross-synthesis.

The data output from the Gem interface is used to trigger and control SampleCloud prototypes in Kyma. These SampleCloud objects use audio files of a gong and a tiny finger cymbal as their sources. The contrast in sonic output of these two audio files performed as SampleClouds creates a pleasing texture. As the output values of the

---

<sup>52</sup> Stolet, Jeffrey. *Kyma and the SumOfSines Disco Club*. Edited by Adam Shanley. San Bernardino, CA, 2011.

Gem data increases, the pitch and frequency-jitter also increases. Frequency-jitter is a parameter in a SampleCloud prototype that controls the amount of random deviation from a given value.

The Gem also controls the playback of several audio files. These audio files include two HAPI-drum audio files that are pitched at multiple intervals based on the input from the Gem. Each of the audio files loop at a rate that is controlled by their pitch. Lowering the pitch decreases the rate of playback for audio files. Raising the pitch increases the rate of playback for audio files. The pitch offset is relative to the original audio file. This offset is further controlled by rotating the Gem.

Multi-sample playback can be heard in the third section. The Geode data is mapped to the playback rate of the audio files. The MultiSample is unique because it plays a folder of audio files in succession. The onset of the sounds is controlled by Kyma. However, ongoing control over the rate of the output of the audio files is controlled by the output values from the Geode.

The synthesis processes controlled by the Cone interface all have one thing in common — they all have a user controllable timeindex. Because of this, I can use the Cone to scrub the audio files. Since most of the sounds the Cone controls have a voice component, manipulating the Cone generates sound that appears to be speaking. The addition of filters accentuates the midrange frequencies present in the audio files. These filters are also affected by the Cone data.

The chordal pad sound at the end of this work is created by a Morph3DSampleCloud prototype in Kyma. This prototype is similar to the previously

mentioned SampleCloud, except that it assigns up to eight different audio files in the *virtual* corners of a 3D space. The Cone data controls this which audio file(s) are heard at any given time. This sound is different from the rest because the Cone controls the pitch of the resultant sound output, while the Geode controls the octave the Cone is functioning in. This mapping scheme allows me to alternate between two pitches by rolling the Cone and shifting the pitches up (or down) by an octave.

Cross-synthesis can be heard in all of the sounds controlled by the Cone. Each of these sounds is filtered by the analysis of a different sound. This process accentuates a specific band of frequencies from one audio file based on the analysis of a second audio file and contributes to the overall sound world.

## **VISUAL DIMENSION**

The primary visual content for this work is provided by the LED lights inside each GEM and the vinyl performance surface. The LEDs illuminate the interfaces from the inside. The LED used for the Gem and the Geode are static and set to their brightest output. The brightness of the Cone's LEDs are determined by the Cone's current data output value. Rolling the Cone will produce a fading effect from the internal lights.

In addition to the shapes and colors of the GEMS, I created a vinyl surface for live performance. This surface is printed on blackout vinyl to help reduce the glare from stage lighting. This surface has designs printed on it that correspond to each GEM and its physical shape. The performance surface does not in any way change the behavior of the

GEMS, but it does provide a visual context for the audience (and performer) during a live performance.

## **COMPOSITION**

This work is performed in four sections, however, the overall shape is amorphous. The structure is fluid and the sections overlap in various ways. Each section features one of the interfaces, with an ending that requires parametric control over a sound. The final section is triggered by the Gem and modified by the Geode and Cone in real time.

The opening section of this work utilizes the Fibonacci sequence for the low frequencies that move to high frequencies. These frequencies are triggered by my turning of the green Gem controller and can be heard from 2:22 in the video. This rising chordal push is accompanied by a metallic, scratchy sound that is spatialized across the stereo field. The speed of this metal sound is controlled in real time by the performer interacting with the Geode, but is changed at a rate similar to that of the low frequency sound. The Cone is then used to present a very lengthy audio file. As the Cone rolls around, the character of the noise is modulated by a simple bandpass filter. So the act of rotating the Cone both plays through the timeindex of the current audio file and controls the filter.

The next section is focused around similar frequency relationships, but a different sound world. This section is thin in terms of low frequency content to balance the abundance of low frequencies in the previous section. Each of the controllers helps to accelerate the playback rate of audio files that create a frenetic mix of obtuse sounds at higher frequencies.

The final section has more balanced frequency content. The sound world in this final section calms down in terms of overall energy. There is a move away from the noise world and a presentation of a more subdued mixture of sounds. This final section features a granulated bell sound whose pitch is controlled by the rotation of the Cone.

## **PERFORMANCE**

The Gem controls the pitch of seventeen oscillators. As the Gem is turned on its X-axis, the facet that is currently at the bottom becomes the active switch and its value is used to change a frequency in Kyma. These frequency values are derived from the beginning of the Fibonacci sequence.<sup>53</sup> As I rotate the Gem from left to right I generate a new frequency value that corresponds to the next number in the sequence. There are ten facets and therefore, ten switches. The ten facets are mapped to the first ten numbers in the Fibonacci sequence to determine the relationship of the frequencies generated by the Gem.

Spinning the Geode around its Y-axis increases its output value. The primary application for this interface is to control the rate of playback for a folder of audio files. Spinning the Geode clockwise increases its output value, while spinning counter-clockwise decreases its output value. The Geode can spin past the point that generates its maximum output. In this case, the data output will cycle back to 0. Spinning the Geode counter-clockwise will create the same result, in reverse. The data output will cycle back to a value of 127 when turned past the point of a 0 output.

---

<sup>53</sup> D. B. Spangler,. (2014). Dipping into the Fibonacci Numbers. *Mathematics Teaching in the Middle School*, 19(5), 266–268. <https://doi.org/10.5951/mathteachmidscho.19.5.0266>

The Cone is used to control sounds that have extended lengths. Setting the cone in motion will allow it to spin in either direction. Spinning the cone will scrub through the timeindex of the sound, forwards or backwards depending on the direction of spin (see 3:12 in the video). This allows the performer to catch the Cone and hold it in place, or reverse its direction. There are a few times during this work when I hold the tip of the Cone in my left hand while controlling the spinning of the interface with my right hand (see 5:00 in the video). This technique allows me to focus on a selected location in the timeindex of an audio file to manipulate at a finer resolution. The act of manipulating the Cone in this way provides another unique, performative approach to generating data from this interface.

My interactions with the iPad during performance are focused on two types of controls. First, I use the controls in Mira, provided by Max/MSP to control the relative amplitudes of the sounds heard at any given point. Second, I direct Kyma to start and stop each different sound in the MultiGrid. During setup and debugging, I am able to open and close different windows for calibration, setup, etc., from the user interface I created using Mira in Max/MSP.

## SUMMARY

Though I spent only four years pursuing this degree, this dissertation represents a lifelong pursuit to compose and perform my own musical and artistic visions. I am honored to share my work with an audience. My hope is that my work illicit a sense of magic in another. The same sense of awe and wonder I feel when I see and hear captivating artistic performances. The element of magic is a common thread through this dissertation.

The seven compositions in this dissertation employ seven different DDIs. In some cases, I am repurposing video game controllers to generate the data needed in my compositions, while other times I am using creations of my own. These creations use light, movement, and above all — interaction to generate data I map to aspects of a musical performance.

Each of these seven compositions feature a different approach to sound synthesis, though many overlap and borrow processes and audio files from one another. Using a sound in multiple compositions is challenging because you risk over exposing your audience to the same content. I am happy that I was able to share some of the same audio recordings between several of the seven compositions without repeating the same process or combinations of modifications.

As a drummer, my body understands the muscle memory needed to perform a task in front of an audience. However, performing with a DDI is so very different from one interface to the next because no two interfaces are alike. Even though they mostly are the same types of sensors, the embodiment of these sensors affect how I interact with

them. Every data-driven instrument requires its own set of movements and interactions for performance.

Each of these seven compositions represents a milestone for me as an artist, composer, performer, designer, maker, and dreamer. Every one of these milestones required me to learn something new, or take a chance on an inspiration.

# APPENDICES

## APPENDIX A

### CLARA VOCE - ENGLISH TRANSLATION OF TEXTS

#### English Translation - Green

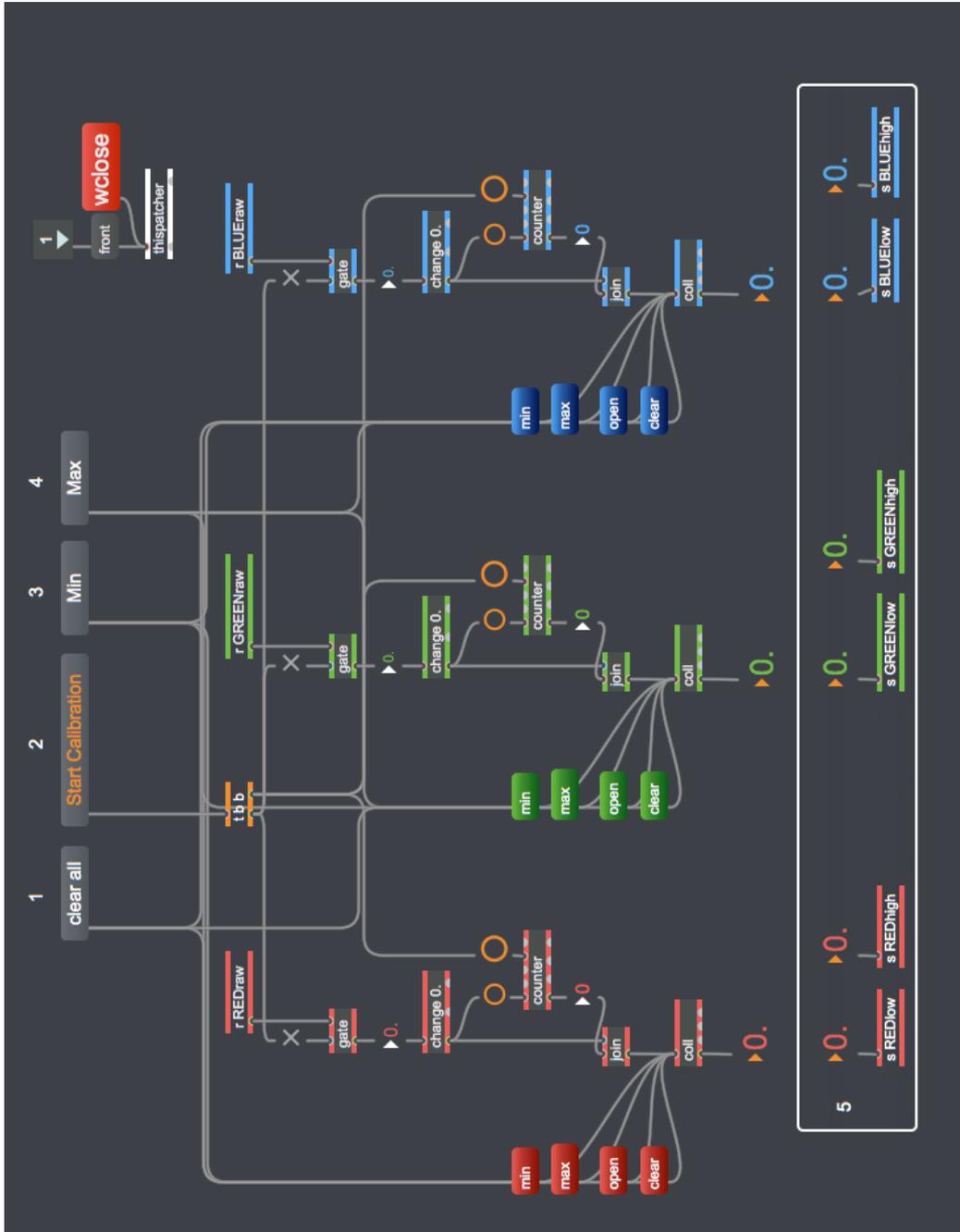
Finances  
Fertility  
Luck  
Charity  
Ambition  
Rejuvenation  
Jealousy

#### English Translation - Blue

Tranquility  
Comprehension  
Honor  
Passion  
Sincerity  
Equality

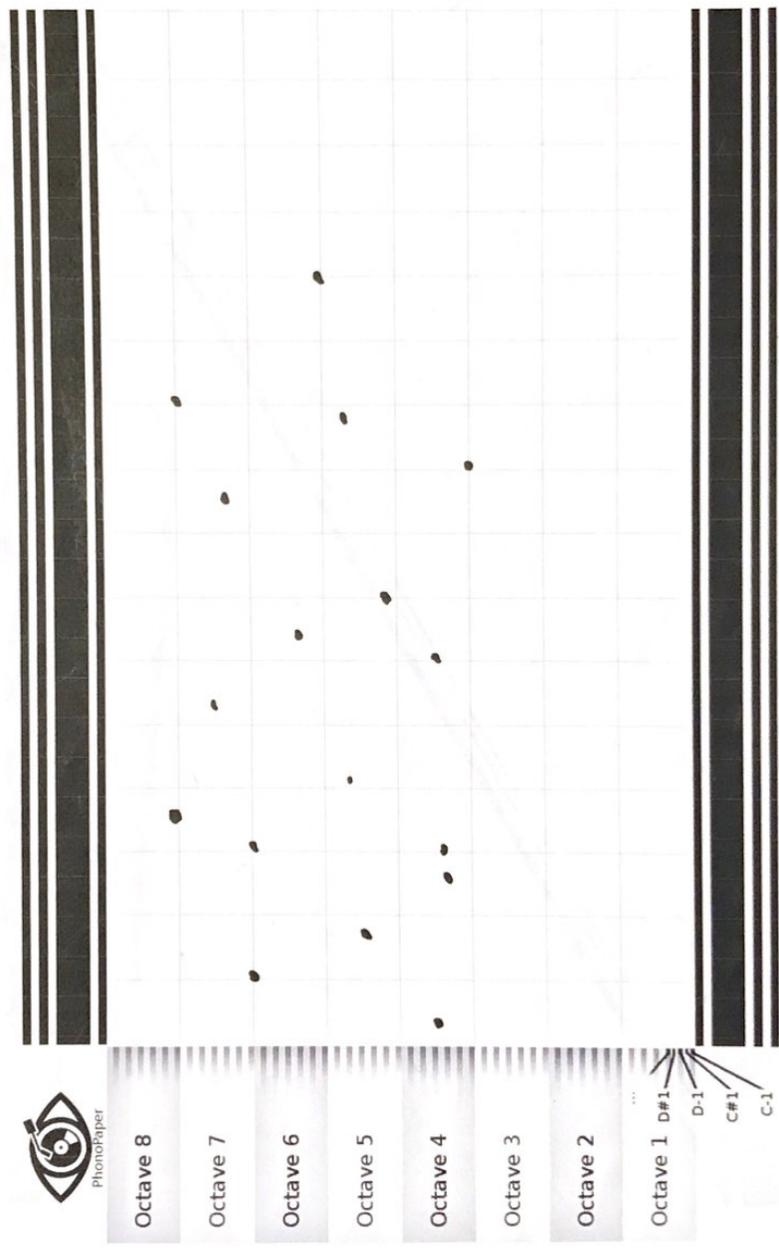
# APPENDIX B

## CHANNELING RACHMANINOFF CALIBRATION SOFTWARE

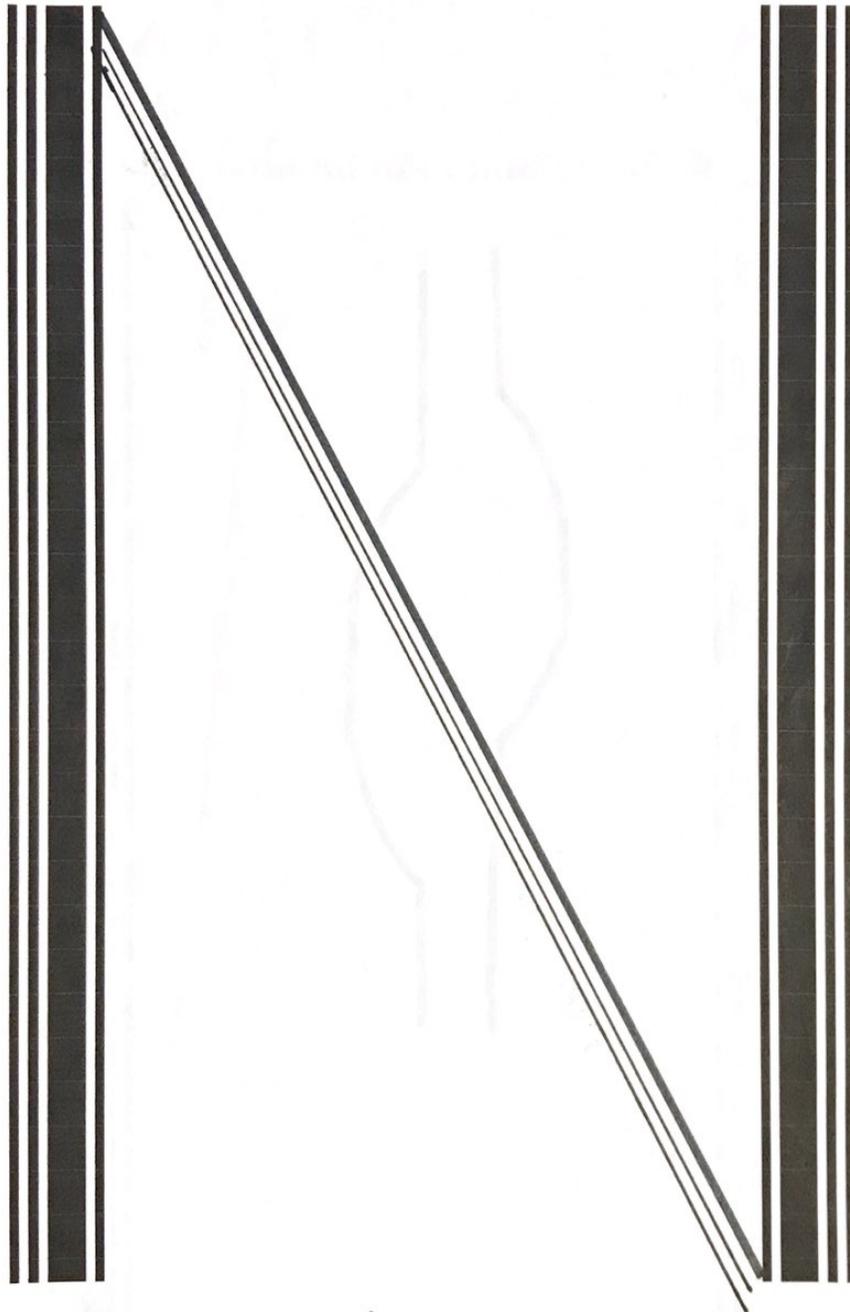


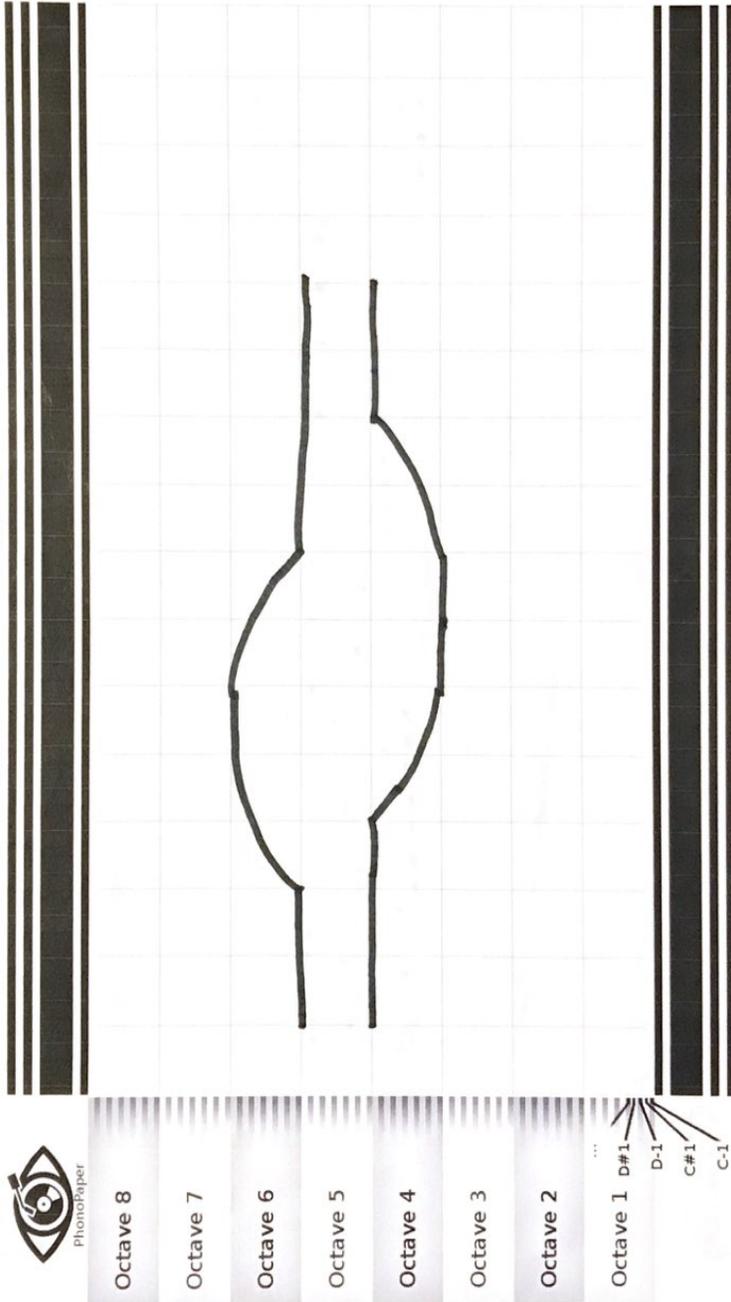
# APPENDIX C

## CURVES & LINES - DRAWINGS









Octave 8

Octave 7

Octave 6

Octave 5

Octave 4

Octave 3

Octave 2

Octave 1

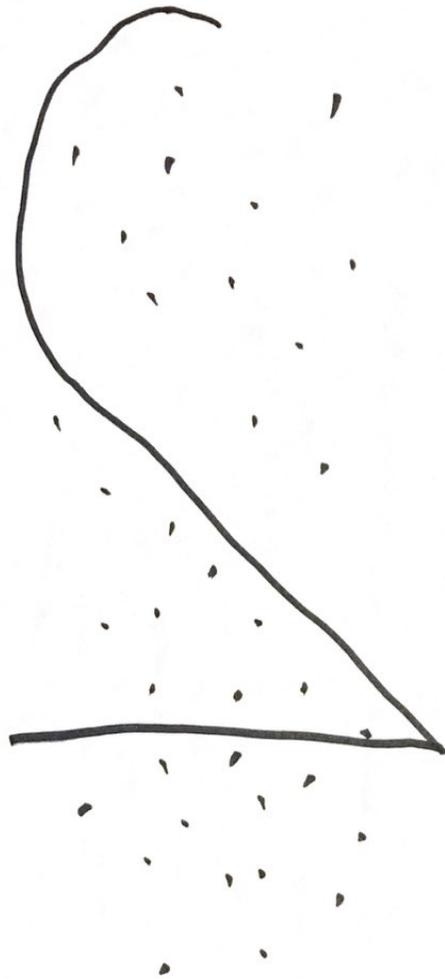
...

D#1

D-1

C#1

C-1





# APPENDIX D

## CURVES & LINES - BLANK SCORE SHEET

Phonopaper

Octave 8  
Octave 7  
Octave 6  
Octave 5  
Octave 4  
Octave 3  
Octave 2  
Octave 1  
D#1  
D-1  
C#1  
C-1

**APPENDIX E**  
**EUPHONY - ALL TEXTS**

From *Winterreise* (English)

I'm used to Going Astray  
And Every Sorrow to its Grave  
I have Found Shelter

I Dreamed of many Colored Flowers  
The Way They Bloom in May  
I Dream of Green Meadows  
Of Merry Bird Calls

From *Pussy Willow*

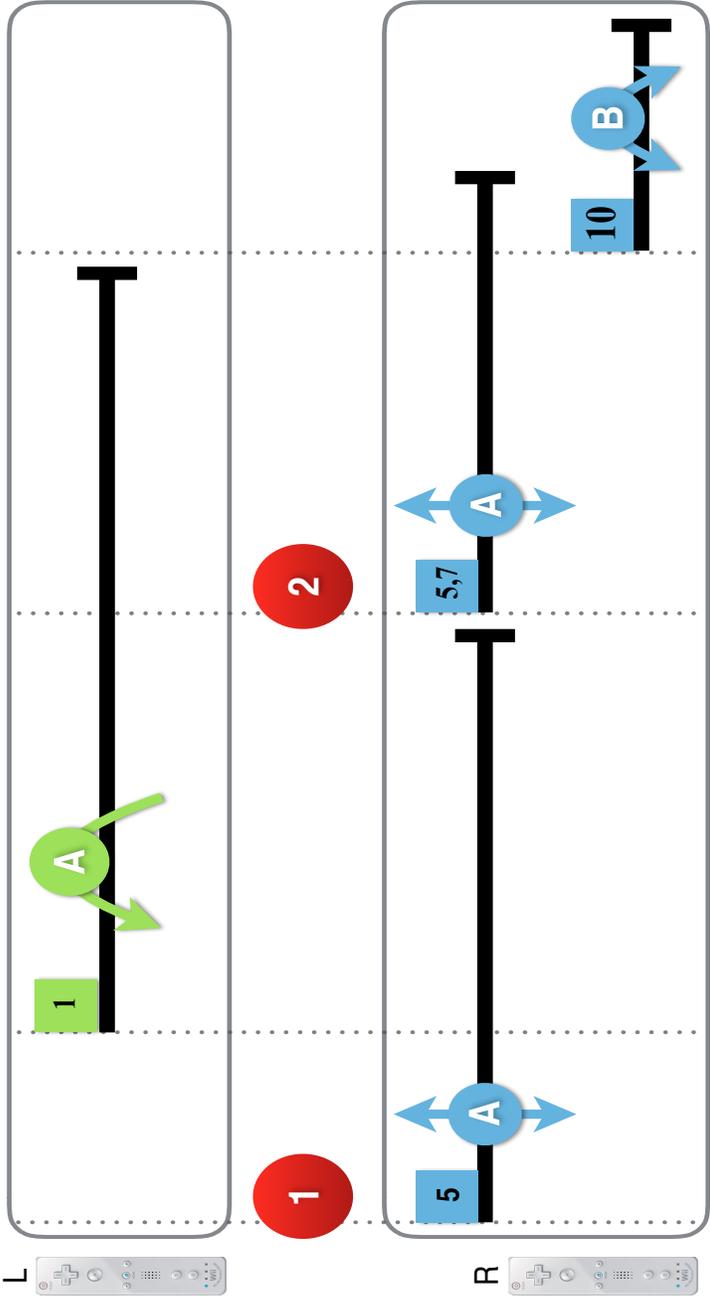
And All The Birds Began to Sing

From *Summer Midnite*

Warm Midnite Falling  
Stars Shining  
Dancing Brightly  
Peaceful All at Once

APPENDIX F  
(IS) - FULL SCORE

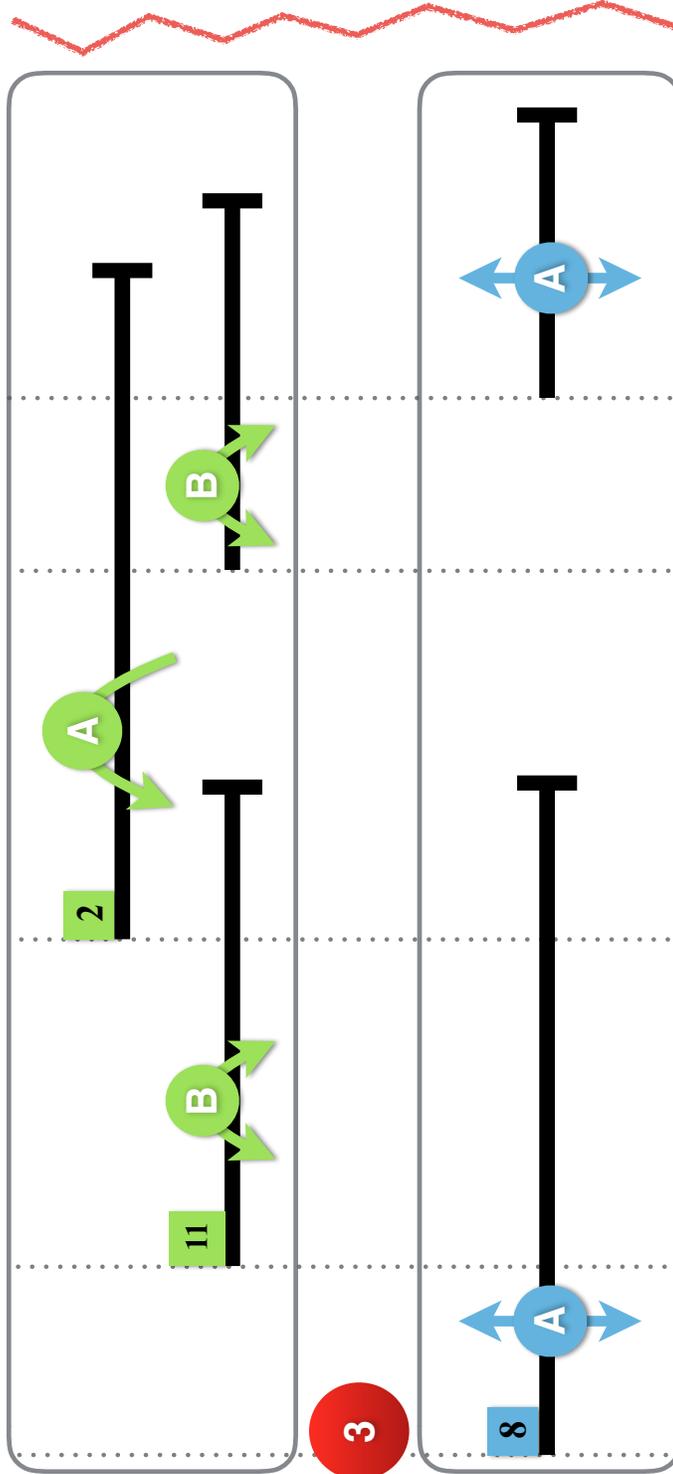
The Insistence of Sameness



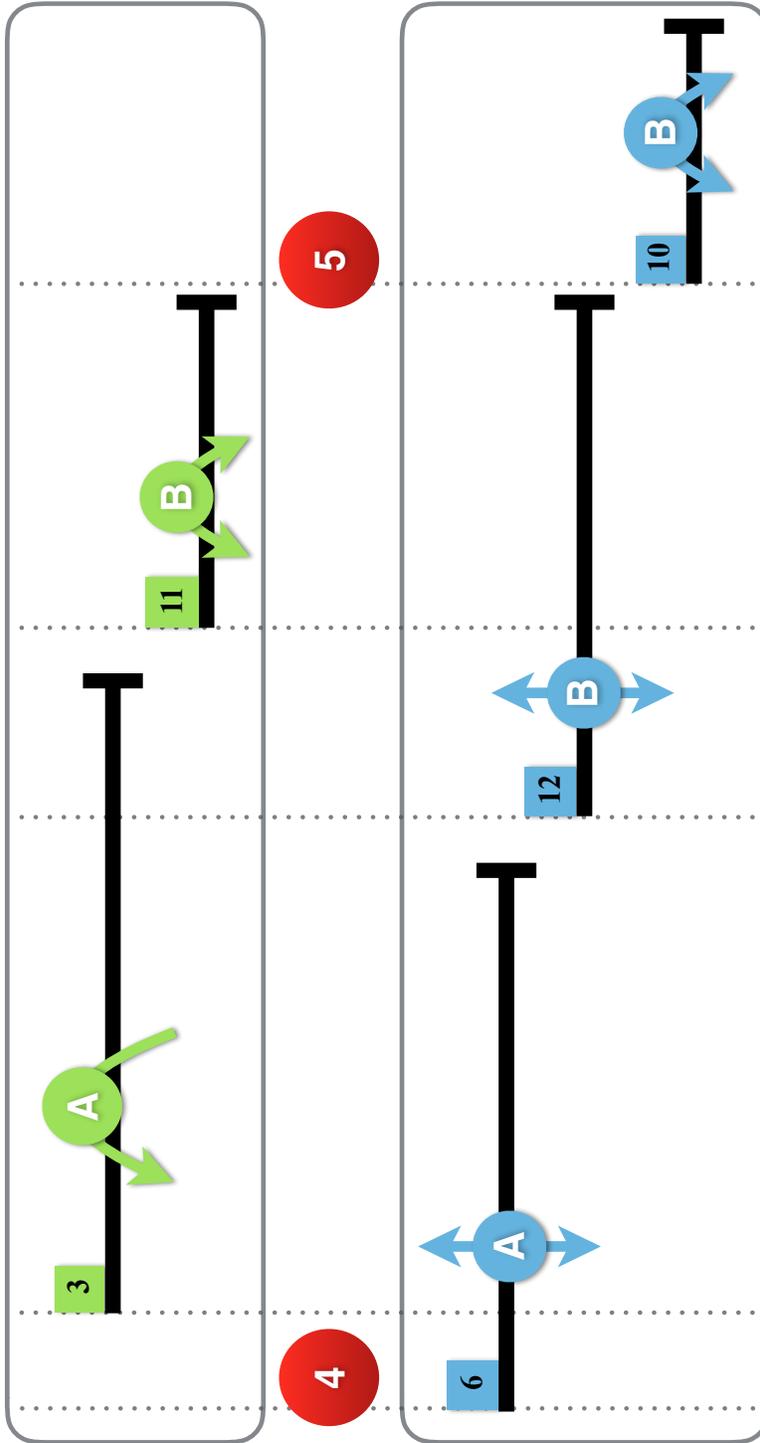
2017 Steve Joslin

1 of 5

# The Insistence of Sameness



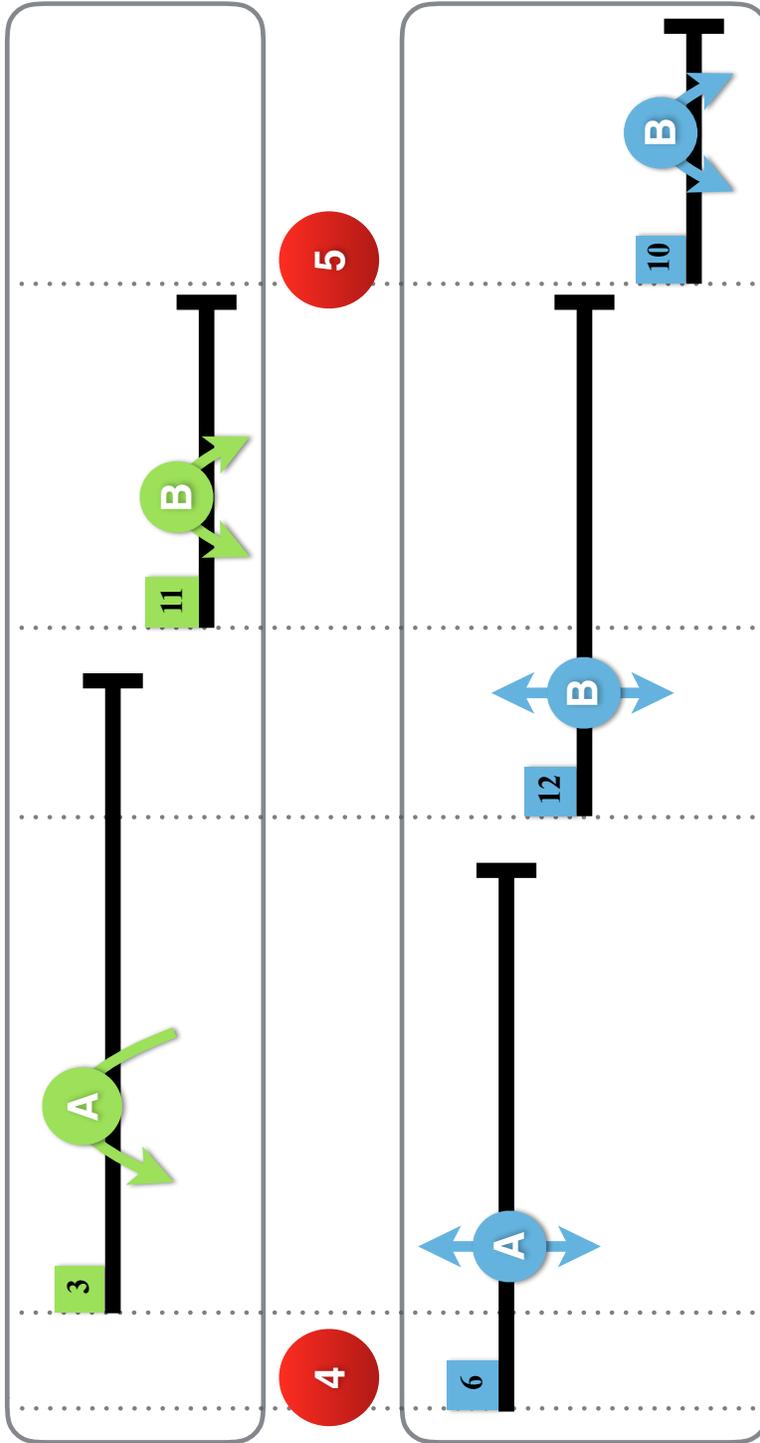
# The Insistence of Sameness



3 of 5

2017 Steve Joslin

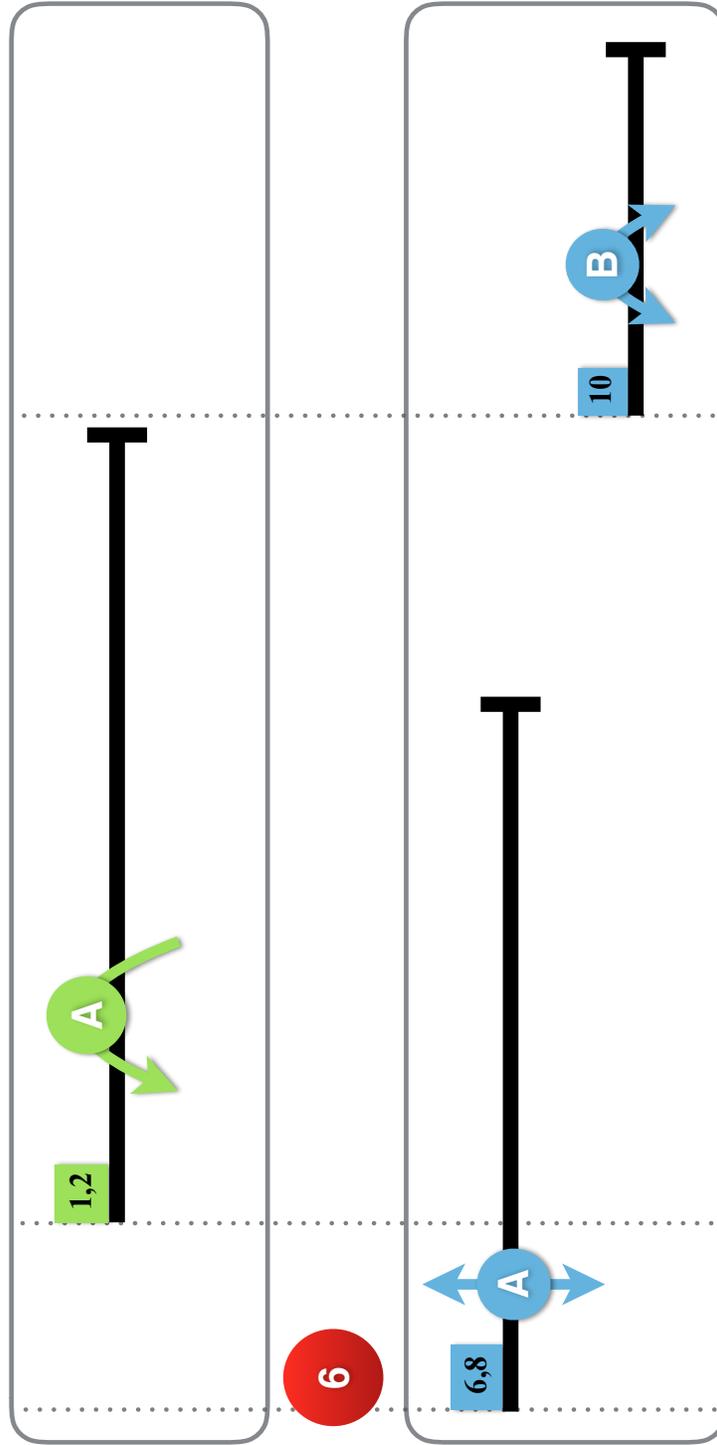
# The Insistence of Sameness



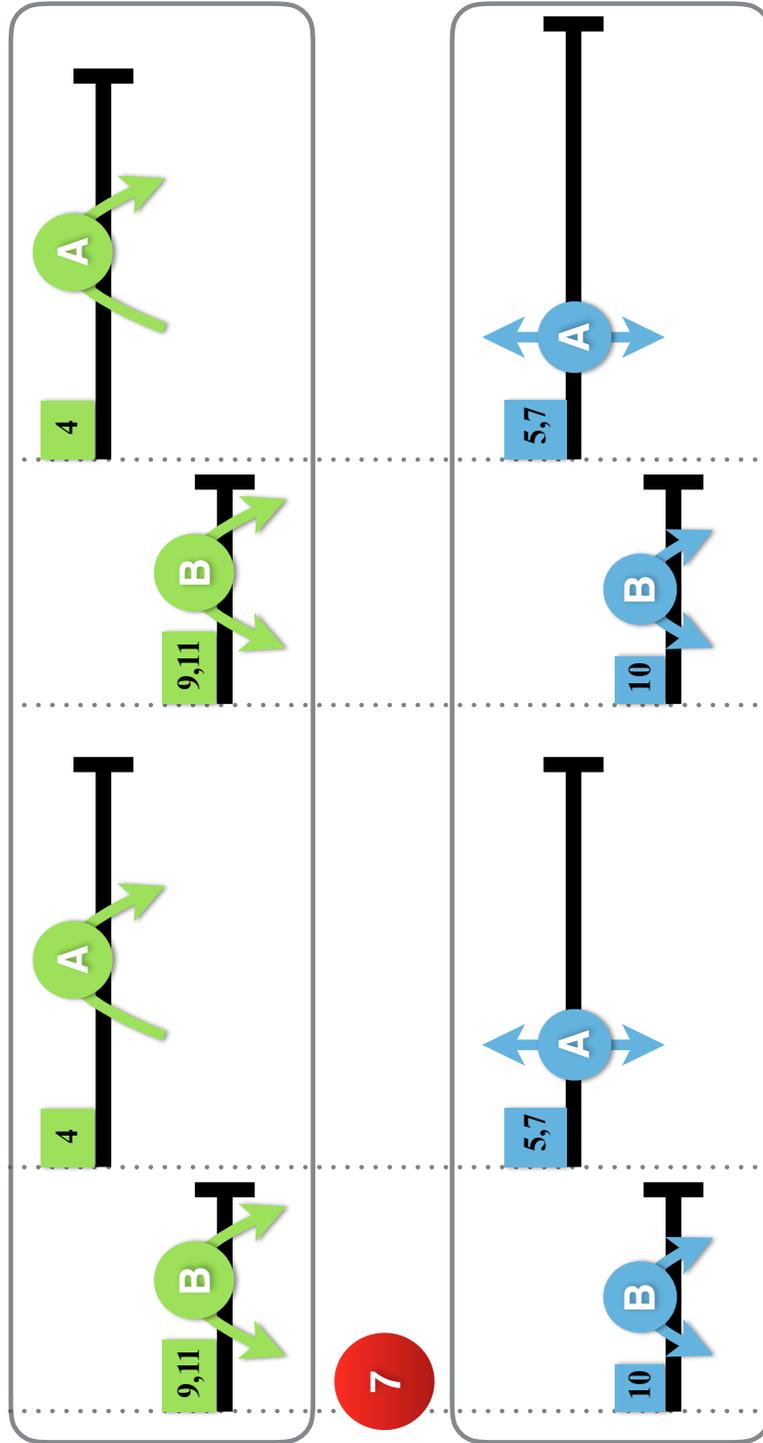
3 of 5

2017 Steve Joslin

# The Insistence of Sameness



# The Insistence of Sameness



# APPENDIX G

## (IS) - MIRA

Insistence\_of\_Sameness  
The Insistence of Sameness

Section **0**  
Scale **C-2**  
Note **C-2**

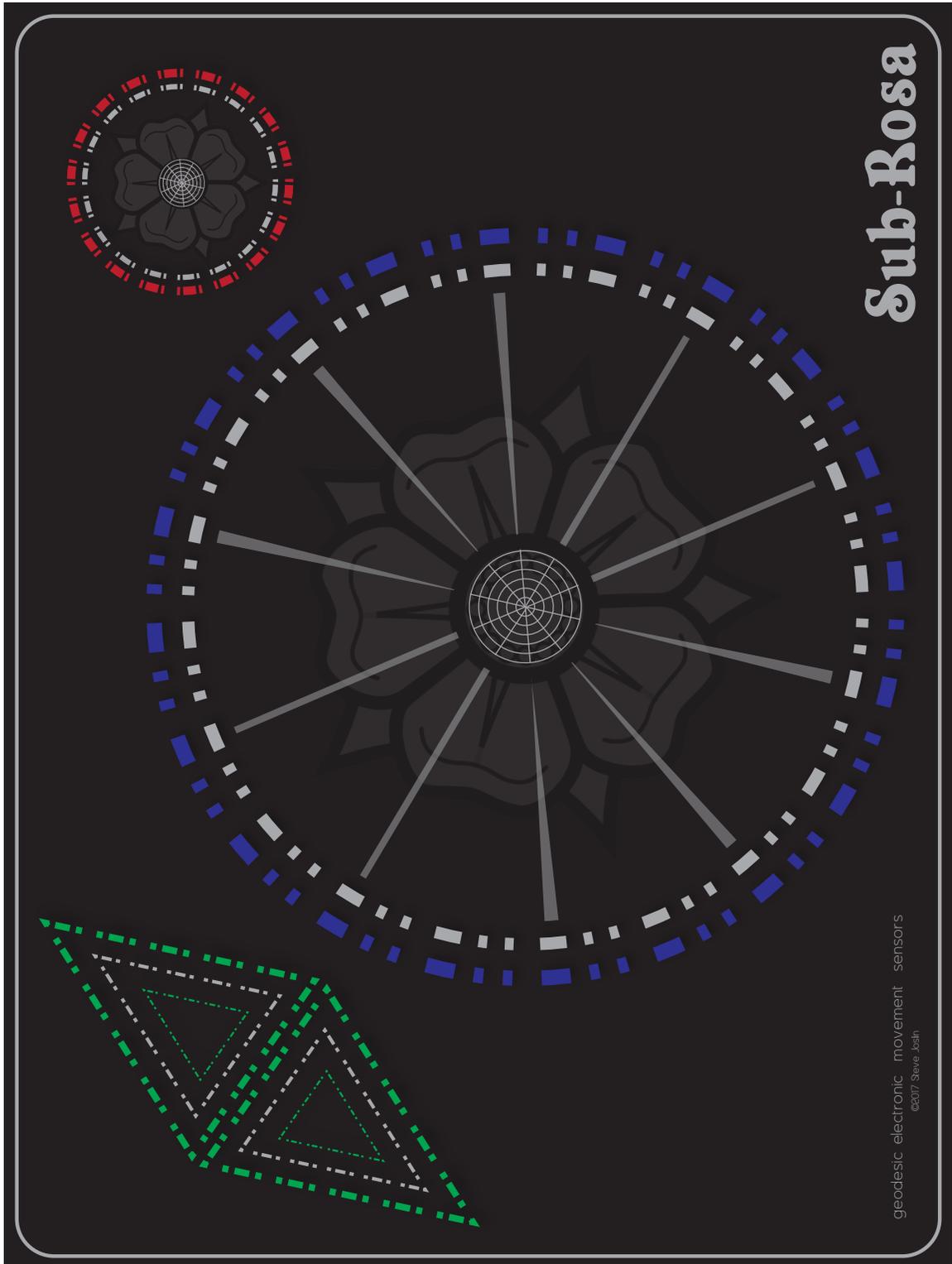
35 L  
A B

49 R  
A B

Elapsed Time **0 : 0**  
Section Time **0 : 0**

# APPENDIX H

## SUB-ROSA - VINYL SURFACE



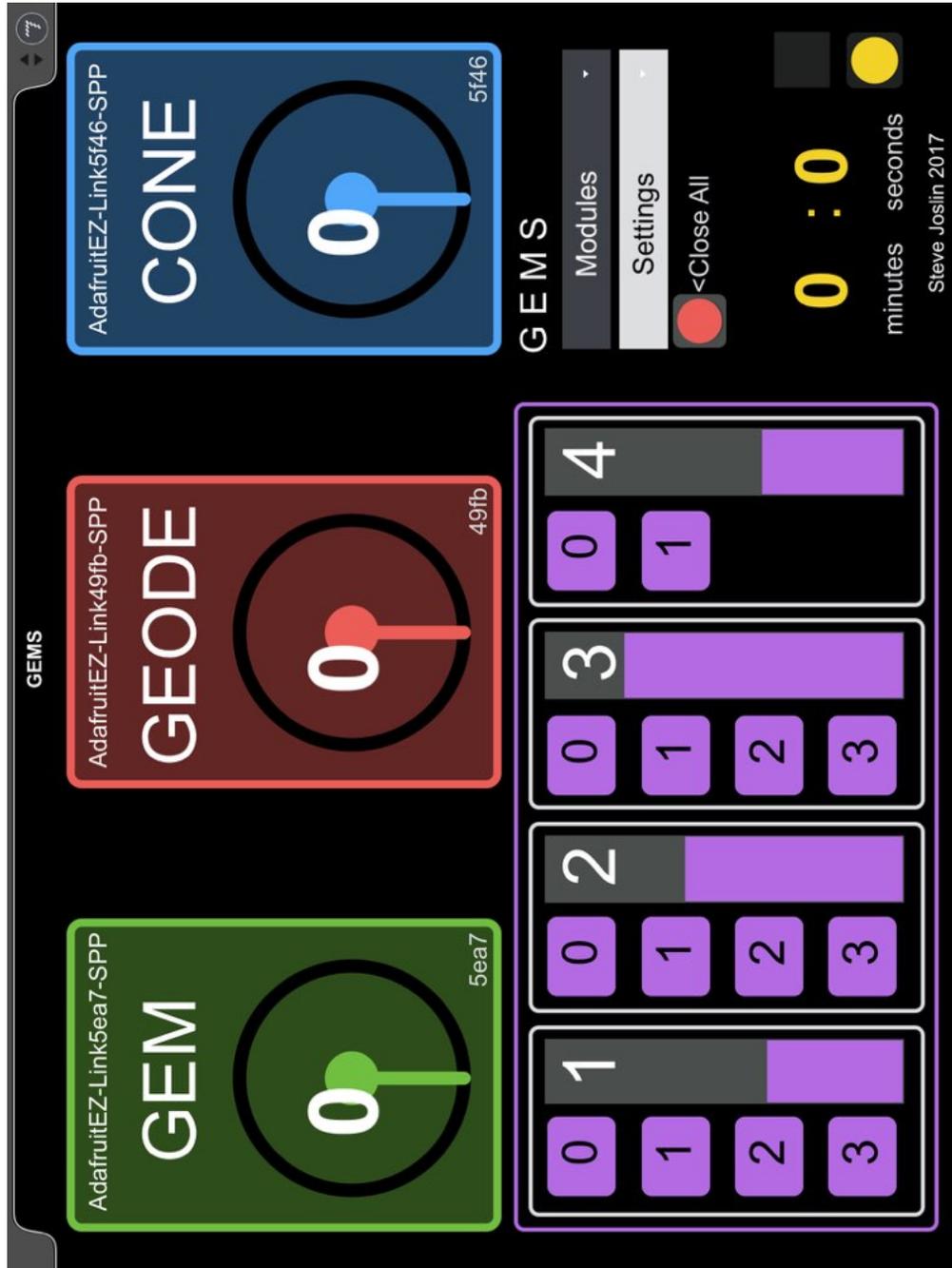
APPENDIX I  
SUB-ROSA - GEM CALIBRATION

The screenshot displays the 'GEM Calibration' interface. At the top left is a 'Close' button. Below the title, there are several purple buttons with values: 0, 19, 37, 56, 73, 95, 113, and 127. A large black circle with a white '0' is positioned at the bottom center. The interface is divided into two main sections: 'Gem Triggers' and 'Filter Noisy Notes'. The 'Gem Triggers' section has a subtitle '(set trigger points for GEM)' and a table with 10 columns. The 'Filter Noisy Notes' section has a subtitle '(filter noisy notes from trigger points)' and a table with 10 columns. A '999' button is located at the bottom left of the 'Filter Noisy Notes' section.

GEM Calibration									
0	19	37	56	73	95	113	127		
Gem Triggers (set trigger points for GEM)									
1	2	3	4	5	6	7	8	9	10
▶0	▶19	▶37	▶56	▶73	▶95	▶113	▶127	▶0	▶0
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Filter Noisy Notes (filter noisy notes from trigger points)									
1	2	3	4	5	6	7	8	9	10
▶999	▶999	▶999	▶999	▶999	▶999	▶999	▶999	▶999	▶0

# APPENDIX J

## SUB-ROSA - MIRA



## APPENDIX K

### SUPPLEMENTAL RESEARCH SOURCES

Cage, John. "Williams Mix," 1952. [http://johncage.org/pp/John-Cage-Work-Detail.cfm?work\\_ID=246](http://johncage.org/pp/John-Cage-Work-Detail.cfm?work_ID=246).

Cook, Perry. "Miranda Eduardo R. and Wanderley Marcelo M., New Digital Musical Instruments: Control and Interaction Beyond the Keyboard. Volume 21 of the Computer Music and Digital Audio Series. A-R Editions, Inc., Middleton, WI, 2006. ISBN 0-895790585-X." Organised Sound 12, no. 2 (2007): 184–185. <https://www.cambridge.org/core/article/miranda-eduardo-r-and-wanderley-marcelo-m-new-digital-musical-instruments-control-and-interaction-beyond-the-keyboard-volume-21-of-the-computer-music-and-digital-audio-series-ar-editions-inc-middleton-wi-2006-isbn-08>.

Moog Foundation. "Bob Moog: A Timeline | Moog Music Inc." Ewebsite. Accessed November 12, 2017. <https://www.moogmusic.com/legacy/bob-moog-timeline>.

Vigliensoni, Gabriel, and Marcelo M Wanderley. "A Quantitative Comparison of Position Trackers for the Development of a Touch-Less Musical Interface" (n.d.). Accessed November 13, 2017. [http://www-personal.umich.edu/~yangqi/NIME2012/papers/155\\_Final\\_Manuscript.pdf](http://www-personal.umich.edu/~yangqi/NIME2012/papers/155_Final_Manuscript.pdf).

## REFERENCES CITED

- Ableton. "Music Production with Live and Push | Ableton." Last modified 2018. Accessed March 19, 2018. <https://www.ableton.com/>.
- Ableton. "Max for Live | Ableton." Accessed March 19, 2018. <https://www.ableton.com/en/live/max-for-live/>.
- Adafruit. "Adafruit Pro Trinket - 5V 16MHz ID: 2000 - Adafruit Industries, Unique Fun DIY Electronics and Kits." Last modified 2017. Accessed March 19, 2018. <https://www.adafruit.com/product/2000>.
- Adafruit. "NeoPixels : Adafruit Industries, Unique Fun DIY Electronics and Kits." Last modified 2017. Accessed March 19, 2018. <https://www.adafruit.com/category/168>.
- Adobe. "Buy Adobe Illustrator CC | Vector Graphic Design Software." Website. Accessed May 7, 2017. <http://www.adobe.com/products/illustrator.html>.
- AudioBus. "Audiobus Developer Center." Accessed March 19, 2018. <https://developer.audiob.us/>.
- Autism Speaks. "Neurobiological Mechanisms of Insistence on Sameness in Autism | Autism Speaks." Autism Speaks Website. Accessed May 1, 2017. <https://www.autismspeaks.org/science/grants/neurobiological-mechanisms-insistence-sameness-autism>.
- Autism Wiki. "Stimming | Autism Wiki | FANDOM Powered by Wikia." Autism Wiki. Accessed April 1, 2017. <http://autism.wikia.com/wiki/Stimming>.
- Blender. "Blender.Org - Home of the Blender Project - Free and Open 3D Creation Software." Accessed March 19, 2018. <https://www.blender.org/>.
- CNMAT. "Introduction to OSC | Opensoundcontrol.Org." Website. Last modified 2018. Accessed April 25, 2018. <http://opensoundcontrol.org/introduction-osc>.
- Cycling74. "Cycling '74." Website. Last modified 2018. Accessed May 1, 2017. <https://cycling74.com/>.
- Dictionary.com. "QR Code | Define QR Code at Dictionary.Com." Accessed March 19, 2018. <http://www.dictionary.com/browse/qr-code?s=t>.

- Dictionary.com. "Data Communication | Define Data Communication at Dictionary.Com." Accessed March 19, 2018. <http://www.dictionary.com/browse/data-communication>.
- Expedia. "Visit Temple of the Sinawava in Springdale | Expedia." Accessed March 19, 2018. <https://www.expedia.com/Temple-Of-The-Sinawava-Springdale.d6298730.Vacation-Attraction>.
- Focusrite. "Focusrite." Website. Accessed May 2, 2017. <https://itrack.focusrite.com/products/itrack-dock>.
- Formlabs. "Form 2 | Formlabs." Accessed March 20, 2018. [https://formlabs.com/store/us/form-2/?utm\\_source=google&utm\\_campaign=GP\\_US-NA-Purchase-Search\\_Brand-Trademark-EN-Exact-Paid-Adwords&utm\\_term=formlabs\\_3d\\_printers&utm\\_medium=cpc&gclid=CjwKCAjw4sLVBRAIEiwASbIR-y5I3VqW3mmx0gvLbO9xDUdiigQd7CLcxrt01GNKkqbS-0ii](https://formlabs.com/store/us/form-2/?utm_source=google&utm_campaign=GP_US-NA-Purchase-Search_Brand-Trademark-EN-Exact-Paid-Adwords&utm_term=formlabs_3d_printers&utm_medium=cpc&gclid=CjwKCAjw4sLVBRAIEiwASbIR-y5I3VqW3mmx0gvLbO9xDUdiigQd7CLcxrt01GNKkqbS-0ii).
- Freed, Adrian. "Musical Applications and Design Techniques for the Gametrak Tethered Spatial Position Controller." Proceedings of the SMC 2009 - 6th Sound and Music Computing Conference (2009).
- Greek Legends and Myths. "Ouriana in Greek Mythology - Greek Legends and Myths." Accessed March 19, 2018. <http://www.greeklegendsandmyths.com/ourania.html>.
- Henke, Robert (Monolake). "Granulator II | Ableton." Accessed March 19, 2018. <https://www.ableton.com/en/packs/granulator-ii/>.
- Hydro Flask. "21 Oz Standard Mouth Bottle." Last modified 2017. Accessed February 2, 2018. <https://www.hydroflask.com/21-oz-standard-mouth-flex/color,black,a,92,o,20>.
- Jimbo. "Serial {Communication} - Learn.Sparkfun.Com," n.d. Accessed March 19, 2018. <https://learn.sparkfun.com/tutorials/serial-communication>.
- McMillen, Keith. "SoftStep2 USB MIDI Foot Switch & Expression Pedal Controller | Keith McMillen Instruments." Last modified 2018. Accessed March 19, 2018. <https://www.keithmcmillen.com/products/softstep/>.
- Mental Floss. "A Brief History of the Devil's Tritone | Mental Floss." Last modified 2018. Accessed March 19, 2018. <http://mentalfloss.com/article/77321/brief-history-devils-tritone>.

- National Park Service. "2015 Artist in Residence - Crater Lake Science and Learning Center (U.S. National Park Service)." Last modified 2015. Accessed March 19, 2018. <https://www.nps.gov/rlc/craterlake/artist-in-residence.htm>.
- Native Instruments. "Komplete : Synths : Absynth 5 | Products." Last modified 2018. Accessed March 19, 2018. <https://www.native-instruments.com/en/products/komplete/synths/absynth-5/>.
- Nintendo. "Nintendo WiiU." Website. Accessed May 2, 2017. <https://www.nintendo.com/wiiu/accessories>.
- Oram, Daphne. "Oramics." Website. Accessed April 30, 2017. <https://daphneoram.bandcamp.com/releases>.
- Paiste. "Paiste: Products-&gt; Sounds-&gt; Percussive Sounds." Website. Accessed May 1, 2017. [http://www.paiste.com/e/product\\_any\\_det.php?menuid=67&modelid=37](http://www.paiste.com/e/product_any_det.php?menuid=67&modelid=37).
- Processing Foundation. "Processing.Org." Website. Accessed June 1, 2017. <https://processing.org/>.
- Processing Foundation. "Setup and Draw \ Examples \ Processing.Org." Last modified 2018. Accessed March 19, 2018. <https://processing.org/examples/setupdraw.html>.
- Quarles, Sazaria. "Haiku About Summer Nights, Summer Midnight, Summer Poem." Last modified 2015. Accessed November 1, 2017. <https://www.familyfriendpoems.com/poem/summer-midnight>.
- Rachmaninoff, Sergei. "Prelude in C Sharp Minor, Op. 3, No. 2." Decca, 1990.
- Remo. "REMO Rototom®." Last modified 2018. Accessed March 19, 2018. <http://remo.com/products/product/rototom/>.
- Soviet Cams. "Soviet and Russian Cameras - Lubitel-166." Accessed March 19, 2018. <http://www.sovietcams.com/index.php?2022299833>.
- Spangler, David B. "Dipping into the Fibonacci Numbers." *Mathematics Teaching in the Middle School* 19, no. 5 (2014): 266–268. <http://www.jstor.org/stable/10.5951/mathteachmidscho.19.5.0266>.
- SparkFun. "Analog to Digital Conversion - Learn.Sparkfun.Com." Last modified 2017. Accessed March 19, 2018. <https://learn.sparkfun.com/tutorials/analog-to-digital-conversion>.

- Stolet, Jeffrey. *Kyma and the SumOfSines Disco Club*. Edited by Adam Shanley. Morrisville: Lulu Press, 2012.
- Symbolic Sound. "Kyma•tweaky: Learn Capytalk." Accessed March 19, 2018. <http://www.symbolicsound.com/cgi-bin/bin/view/Learn/Capytalk>.
- Travel Belize. "Altun Ha | Belize | Travel To Belize | Ruins Of Belize | Jade Head —." Accessed March 19, 2018. <http://altunha.com/>.
- USB.org. "Device Class Definition for Human Interface Devices (HID) Universal Serial Bus (USB)" (2001). Accessed March 19, 2018. [http://www.usb.org/developers/hidpage/HID1\\_11.pdf](http://www.usb.org/developers/hidpage/HID1_11.pdf).
- Wacom. "Wacom Intuos Pro." Website. Accessed May 3, 2017. <http://www.wacom.com/en-co/products/pen-tablets/wacom-intuos-pro>.
- Webster. "Magic | Definition of Magic by Merriam-Webster." Accessed May 1, 2017. [https://www.merriam-webster.com/dictionary/magic?utm\\_campaign=sd&utm\\_medium=serp&utm\\_source=jsonld](https://www.merriam-webster.com/dictionary/magic?utm_campaign=sd&utm_medium=serp&utm_source=jsonld).
- Wildora. "Home | OSCulator." Website. Last modified 2018. <https://oscillator.net>.
- Wise-Brown, Margaret. *Pussy Willow*. Random House Children's Books, 1997. [https://books.google.com/books/about/Pussy\\_Willow.html?id=eoMz96LCR4EC](https://books.google.com/books/about/Pussy_Willow.html?id=eoMz96LCR4EC).
- Youtube. "Wolfgang Amadeus Mozart - Wiegenlied (Lullaby), K. 350 - YouTube." Accessed March 19, 2018. [https://www.youtube.com/watch?v=vXRzq\\_8bJ3c](https://www.youtube.com/watch?v=vXRzq_8bJ3c).
- Zolotov, Alexander. "PhonoPaper on the App Store." IOS APP Store. Last modified 2015. Accessed November 13, 2017. <https://itunes.apple.com/us/app/phonopaper/id865947553?mt=8>.

## **SUPPLEMENTAL FILES**

VIDEO: CLARA VOCE

VIDEO: OURIANA'S MEDITATION

VIDEO: CHANNELING RACHMANINOFF

VIDEO: CURVES & LINES

VIDEO: EUPHONY

VIDEO: INSISTENCE OF SAMENESS (IS)

VIDEO: SUB-ROSA

SOFTWARE: CLARA VOCE

SOFTWARE: OURIANA'S MEDITATION

SOFTWARE: CHANNELING RACHMANINOFF

SOFTWARE: CURVES & LINES

SOFTWARE: EUPHONY

SOFTWARE: INSISTENCE OF SAMENESS (IS)

SOFTWARE: SUB-ROSA