ABSTRACT

Personifying Dynamic Video Game Music Through Live Performance: A Rescoring of RiME

Daniel Lujan, M.M.

Mentor: Scott McAllister, D.M.A.

My central argument for this thesis is that video game music loses its adaptive functionality when performed outside of a gameplay context. My goal is to use technology to incorporate live performance into dynamic game music and create an experience that facilitates collaborative human connection. My research will culminate in a performance of originally composed music that will accompany gameplay of the video game *RiME* by Tequila Works. I will give an account of the last year and a half, detailing the work that was required to realize this performance. First I will talk about my use of technology. Then I will discuss my approach to the music writing process. I will also explain how these two aspects work together to create the end result. Finally, I will talk about where I want this research to go in the future and how I want to expand upon it.

Personifying Dynamic Video Game Music Through Live Performance:
A Rescoring of RiME

by

Daniel Lujan, B.M.

A Thesis

Approved by the School of Music

Gary C. Mortenson, D.M.A., Dean

Timothy McKinney, Ph.D., Graduate Program Director

Submitted to the Graduate Faculty of
Baylor University in Partial Fulfillment of the
Requirements for the Degree
of

Master of Music

Approved by the Thesis Committee

Scott McAllister, D.M.A., Chairperson

Ben Johansen, Ph.D.

Terry York, D.M.A.

Matthew Fendt, Ph.D.

Enter Name, Degree

Accepted by the Graduate School

May 2020

J. Larry Lyon, Ph.D., Dean

Page bearing signatures is kept on file in the Graduate School.

Copyright © 2020 by Daniel Lujan

All rights reserved

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER ONE

Introduction

I have had an interest in video game music for as long as I have been composing music. In fact, video game music, specifically the music of *Halo 3* (Bungie, 2007), is what inspired me to start composing in the first place. Incidentally, I have always wanted to incorporate video games into my music writing. The obvious solution to this desire is to simply compose music for video games and call it a day. I definitely do intend to compose music for video games in the "traditional" sense, that is, by having music audio files integrated directly into game engines. However, I believe that my experience as a performer and a composer/collaborater is what has led me to approach video game music from a different angle and ask the question: Can live performance be brought to video game music? The biggest motivation behind this question is the fact that there are actual human musicians behind video game music. I see this as an incredible opportunity for collaboration, in this case between gamer and performer. Given the already hyper-interactive nature of video game music, this opportunity cannot be avoided.

The purpose of my thesis is not to explain what video game music is or talk about its characteristics. However, it will help if I give some background to the genre. Video games differ from most other types of visual media in that they are non-linear. A player is typically in control of the flow of the game, dictating things like elapsed amount of time, location, and actions taken inside the world. As a result of this player agency video game music cannot simply be a finite track that remains the same. No, the music must be ready to change in some way, be it by repeating a section, switching to another section, or adding/removing layers of the track. These changes certainly are not arbitrary. Game music has the ability to communicate a wealth of information, as well as encourage actions, to the player. An example of a game that does this is *The Legend of Zelda: Breath of the Wild* (Nintendo, 2017). There are different enemies in this game that each has their own musical theme. Whenever the player is spotted by any of these enemies the music will begin to play. The player can therefore use the music to determine what type of enemy he will be facing even if he has not yet spotted them. This providing of information will also determine how the player responds and what actions to take. When the music for the weak, goblin-like enemy is cued the player will know that they are not facing much of a threat and may therefore respond in a relaxed manner. On the other hand, the musical theme for the Guardian signifies that this very dangerous enemy is locking onto you and you either need to run or hide and do so very quickly. Despite being a game that is not particularly a "music game", *Breath of the Wild* makes music an integral part of its gameplay, so much so that a player experiencing it without audio might be disadvantaged if not just deprived of the full experience of the game. Another game that utilizes music seamlessly is *The Elder Scrolls V: Skyrim* (Bethesda, 2011). In the game, there are Word Walls from which the player can aquire new abilities. As the player approaches a Word Wall, Nordic music and chanting will be heard and increase in volume as the player gets closer to the wall, encouraging them to do so. A player experiencing this game without music might be more likely to miss one of these Word Walls and therefore miss out on part of the game, detracting from the experience. *Breath of the Wild* and *Skyrim* are examples of games that rely heavily on the interaction of the player and music in order to create an engaging experience. In summary, interactivity is a central characteristic of video game music and is what makes it unique.

My central argument for this paper is that video game music can only be experienced in its fullest capacity during gameplay. It loses its functionality when performed in a concert setting outside of a gameplay context. Essentially, video game music is labeled as such not only because it is played in conjunction with a game, but because it exhibits various non-linear traits that perform an observable role in the video game. With this viewpoint I can go as far as to say that "video game music" that is performed alone (without gameplay) is not really "video game music" at all. At first glance this claim sounds harsh but I want to clarify that I do not use the label "video game music" as a legitimizing term. My intention is not to confine video game music as an art form that *should only* be experienced within the context of gameplay. The ability of game music to break boundaries and be performed in out-of-gameplay contexts is a testament to its artistic value. A concert of linear music of Zelda, Halo, and Skyrim is an event I absolutely want to attend! Rather, I am intentional with the term "video game music" because it represents a genre of music that 1) is incredibly unique and 2) requires an entire set of distinctive techniques in order to create.

One of my goals is to highlight the qualities that make video game music unique and interesting and bring them into a performance setting to be experienced among multiple people, preferably of different backgrounds. My hope is that in the process, interesting interactions are fabricated, resulting in an elevated appreciation for both video game music and performers of music.

CHAPTER TWO

First Iterations

Before I dive into the details of my project in its current form, I want to give some insight into the initial workings of it. I did not immediately begin creating the music and technology that will be displayed in the final performance. There was experimentation, prototyping, and trial and error.

I got started on this project right away in the fall of 2018, the first semester of my Master's degree. I set out to create a simple video game and figure out if I could connect it to the application Max then display music in the program (I will talk more about Max later when I discuss technology more in-depth). I ended up succeeding in both of these tasks thanks to a handy OSC script I found online and the *bach* Max plugin.[1] I used the game engine Unity to create a simple exploration puzzle game.

[1] "bach: automated composer's helper," bachproject. https://www.bachproject.net (accessed March 1, 2020).

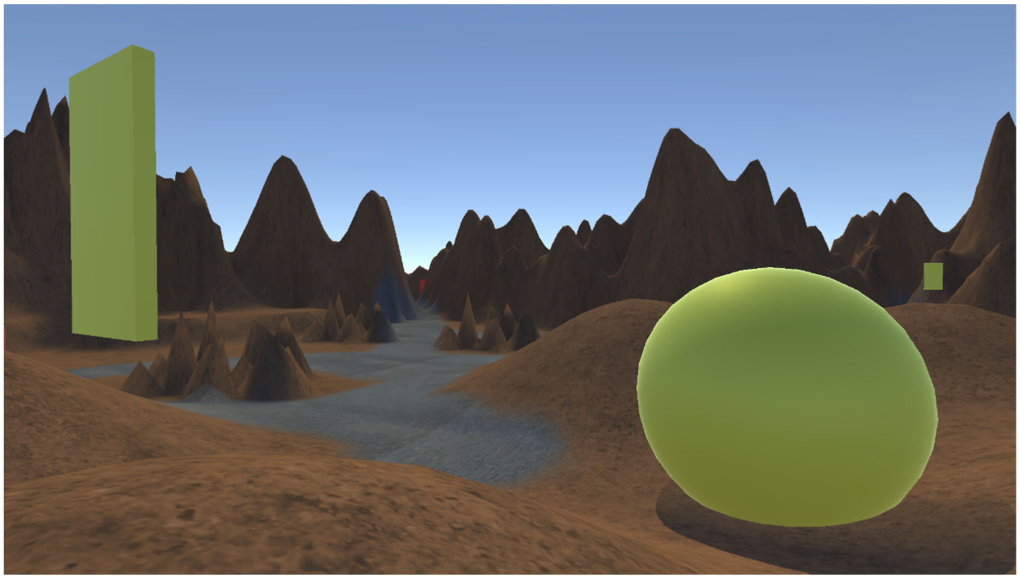


Figure 2.1 Main spawn area in the puzzle game I created

The game consists of four different areas that each has its own unique musical loop, consisting of solo piano music that I wrote. There is the main area where the player spawns then there are three smaller areas on the periphery of the map that each contains a puzzle. My main goal was to have each of these musics displayed in Max whenever the player ventured into the appropriate area, which is a very common game music technique. Staying in one area long enough would cause the music to loop as well, ensuring that the music was constant. I also attempted to get a bit more creative. The main spawn area begins with a musical loop that is fairly simple, but as the player completes puzzles, elements of each puzzle area’s music are incorporated into the musical loop of the main area, gradually increasing its complexity. This technique of additive layering is also very common in game music. In case there is any confusion of how this experiment actually played out, I had one person play the game on one computer while I played the piano, reading from the music that popped up on a second computer.

At the end of this project I was able to successfully incorporate two common game music techniques into live performance.

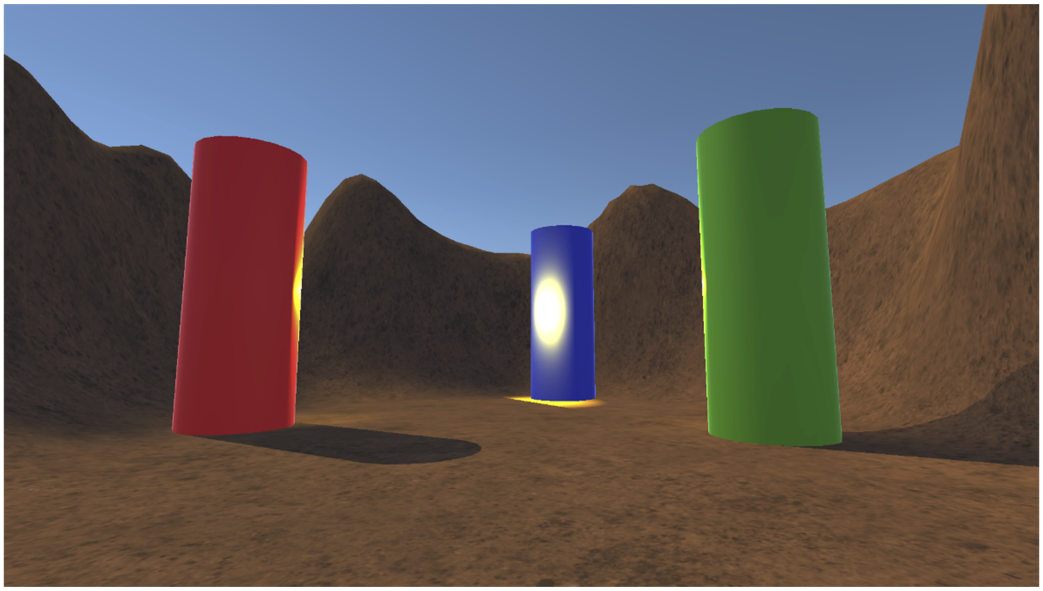


Figure 1.2 A puzzle area in the game

The following semester, spring of 2019, I saught to expand upon the work I had already done. My primary goal was to add more musicians into the equation, which I did, writing an additional violin and clarinet part. This not only gave me a more varied palette of sounds to work with but it most importantly allowed me to sit back and watch how 1) the musicians interacted with the music reading interface and 2) how the person playing the game interacted with the musicians and vice versa. I utilized the same game from the previous semester and arranged the score for violin, clarinet, and piano. The culminating performance can be viewed at the following link:

https://www.youtube.com/watch?v=m-UT41Gtn7c

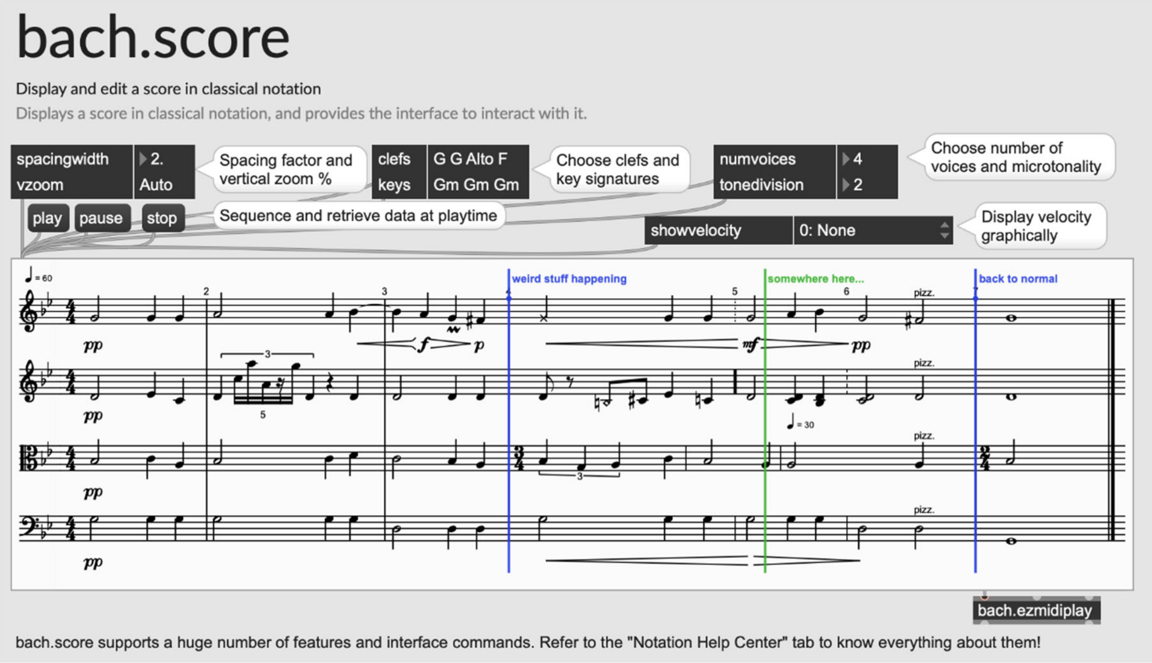
This performance was a pivotal moment for the ongoing development of this project. From it I learned some very important things that I would take into account for future iterations. The first obvious detail was that the game and music were simply too boring. If I intended to create an experience that was both captivating and engaging for player, musician, and audience alike, I needed to improve the quality of game, and perhaps more importantly for me, the quality of music. The second thing I learned was more logistical. In the puzzle game the music essentially depended upon the player moving between areas and spending a decent amount of time in each of them so that the music would have time to play out. However, what I found through player testing was that the player could enter into an area, activating the cue for its music, then quickly turn around and walk right back out. This essentially created the situation where the music was in a state of "catching up" with the player. Although this could be viewed as more a game design issue, I realized that I had to be more strategic with how I wrote the music and distrubuted it inside the game. I also needed to find a way to make the music more fluid and adaptable. Finally, the most important takeaway from that experience was the realization that the musicians needed to have more say in what to play and how to interact with the game. The entire time all they could do was read from the score verbatum. After all, my vision for this project is not only to incorporate live music in video games. It is to do so in a way that fosters collaboration between everyone involved. In the coming chapters I will explain the fruit of these lessons as it relates to preparing the final version of this video-game-classical-concert hybrid performance.

CHAPTER THREE

Approach to Technology

I'm going to explain the technology behind this entire feat first because I think having an understanding of it will help make my approach to writing the music make more sense. A little background on game audio: technology called "audio middleware" is a third party tool set that sits between the game engine and the audio hardware. It gives composers and sound designers the ability to create conditioned presets and variations of their audio. For instance, a sound designer can use audio middleware such as Wwise to build a car engine sound effect that randomizes certain elements of the sound each time it is played, such as pitch and volume. Composers, on the other hand, can use audio middleware to create conditional tracks of music, giving the ability to layer tracks in a certain way or shift from one track to another based on gameplay events. I give this background because I essentially created my own middleware application but for music notation.

I used Max by Cycling '74, which is a visual programming language and environment. This software has an unending amount of applications including, but not limited to, building custom software instruments, creating dynamic visuals, and networking with other computers. The list goes on. The biggest reason I chose to use Max is for a plugin called *bach*, which allows you to display music notation inside Max. Using *bach* I am able to import my pre-composed music as an xml file and load it into the bach.score object. Once there, I can decide how, when, and where I want to display my music to musicians.



bach.score

Display and edit a score in classical notation

Displays a score in classical notation, and provides the interface to interact with it.

spacingwidth 2.
vzoom Auto
clefs G G Alto F
keys Gm Gm Gm
numvoices 4
tonedivision 2
showvelocity 0: None

bach.score supports a huge number of features and interface commands. Refer to the "Notation Help Center" tab to know everything about them!

Spacing factor and vertical zoom %
Choose clefs and key signatures
Choose number of voices and microtonality
play
pause
stop
Sequence and retrieve data at playtime
Display velocity graphically
weird stuff happening
somewhere here...
back to normal
pizz.
pp
f
p
mf
bach.ezmidiplay

Figure 2.1 Bach.score object example

The first part of my middleware lets me create and load any number of these bach scores. This helps me organize my music. For instance, in this performance I utilize three scores, two for each piece of music and one for ambience music. From here I create presets that are individually activated when certain conditions are met. Within each of these presets, inside what I call my Music Navigator, I can determine which score I want to display as well as measure ranges within that score, thus creating a Music Queue. One slight downside of the bach.score object is that I can only show one system of music in what you would call "panorama view" in notation software. It does allow horizontal scrolling but that isn't viable for a live performance. Fortunately in Music Navigator I can decide by how many measures I want to split up each encompassing measure range. I can then cycle through each group on command. Once my presets are created and their Music Queues are set, they are ready to receive firing signals from the Conductor's Controls.

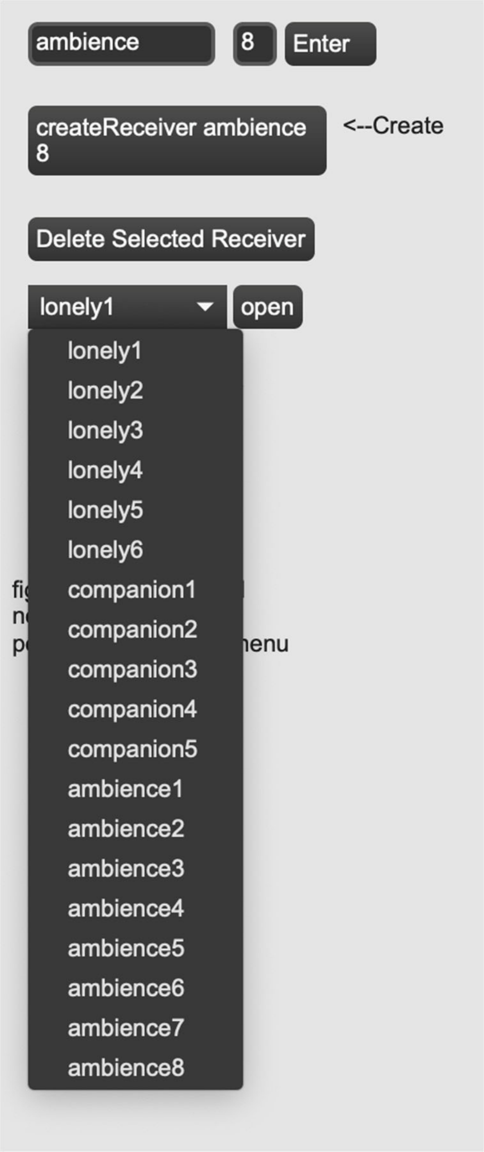
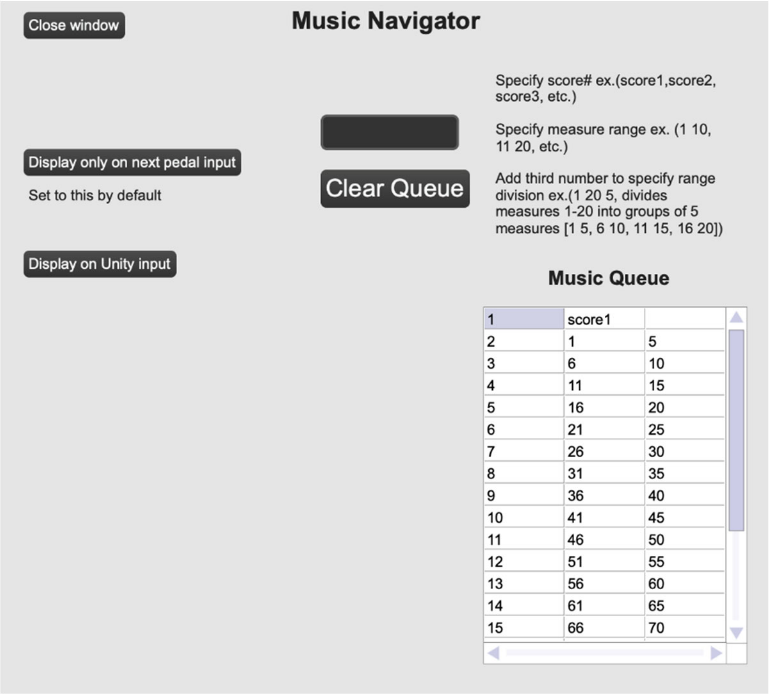


Figure 3.2 Preset maker and lister



Specify score# ex.(score1,score2, score3, etc.)

Specify measure range ex. (1 10, 11 20, etc.)

Add third number to specify range division ex.(1 20 5, divides measures 1-20 into groups of 5 measures [1 5, 6 10, 11 15, 16 20])

1	score1	
2	1	5
3	6	10
4	11	15
5	16	20
6	21	25
7	26	30
8	31	35
9	36	40
10	41	45
11	46	50
12	51	55
13	56	60
14	61	65
15	66	70

Figure 3.3 Music Navigator for defining the parameters for presets

During the performance I use the Conductor’s Controls to activate the presets. The sending signal is based on the name of the presets and the number attached to the end of it. For example, in the conductor’s controls there is a route called “lonely”. When a “1” is sent through this route, it activates the preset with the name “lonely1”. When a “2” is sent it activates preset “lonely2”, and so on. With this setup the Conductor’s Controls can be a simple numerical interface that lets me easily switch presets at command.

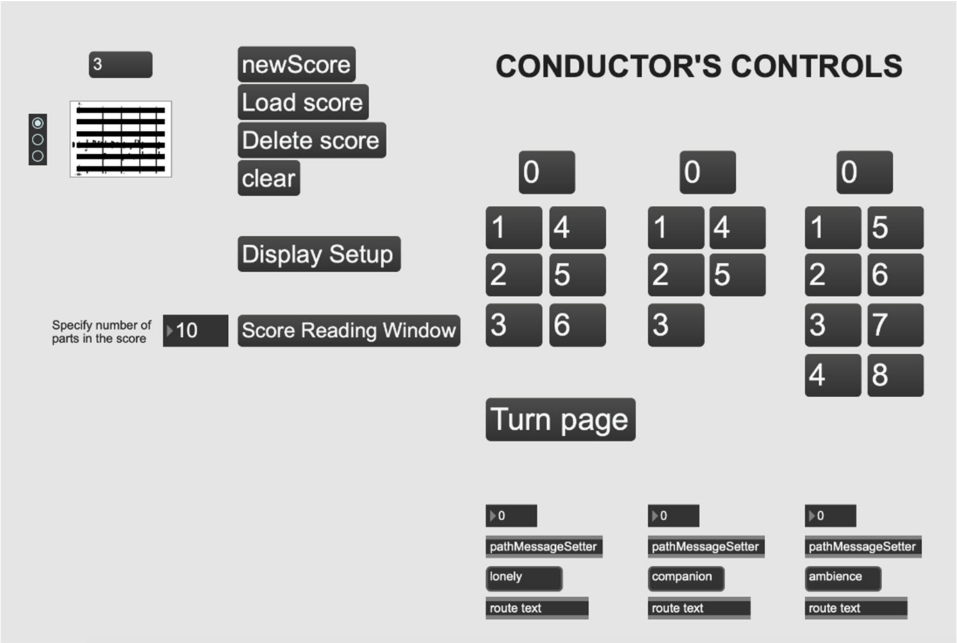


Figure 3.4 Conductor's Controls for use during performance

Now is where networking comes in and where I will explain what happens when a preset is activated. I use UDP (User Datagram Protocol) to send data to other computers that contain a client version of this application, which is the music display interface for the musicians. Once opened it loads the proper xml files into bach.scores then awaits measure range information incoming from the main application on my computer. Back on my machine, when I activate a preset with the Conductor's Controls, it prompts the score and measure number info (Music Queue) to be sent to the client computers via UDP. It does this iteratively. Upon initial activation the score number and first measure group are sent to the client, loading the appropriate music into its bach.score. Upon each consecutive trigger of the Turn Page button in the Conductor's Controls (binded to my space bar), the next measure group is sent to the client, triggering the display of those measures.

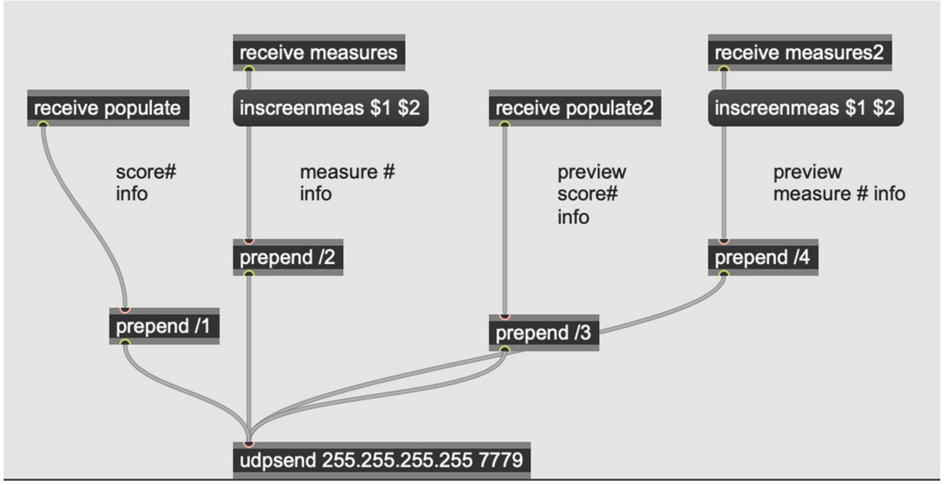


Figure 3.5 Setup inside the main application for sending out data via UDP

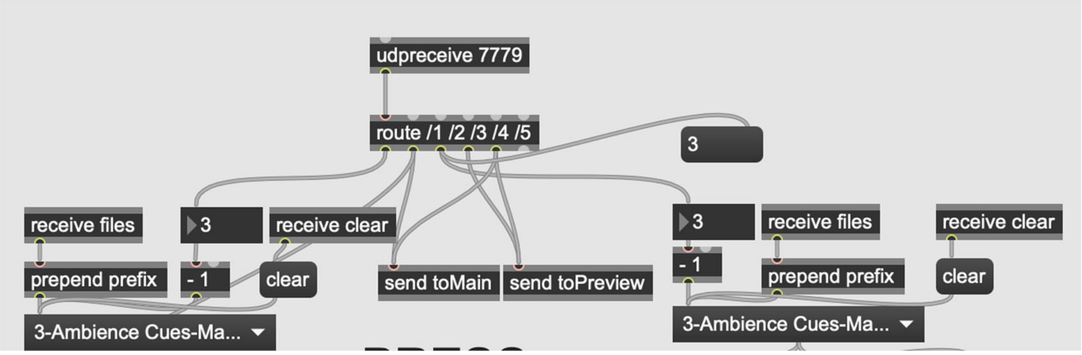


Figure 3.6 Setup inside the client application for receiving data incoming from the main application

This system gives me, as conductor, the ability to display to all of my musicians any piece of my score at any time during the performance. On the client end the musicians are provided with a main score display for reading the current music and a preview display that shows the next set of measures. The interface also allows them to pick what parts they want to view from the score. Therefore they can choose to display either their given part solely or a combination of parts, which can be helpful for maintaining ensemble cohesion. As the conductor I am also able to pull up the same interface and use it to view all parts of the score.

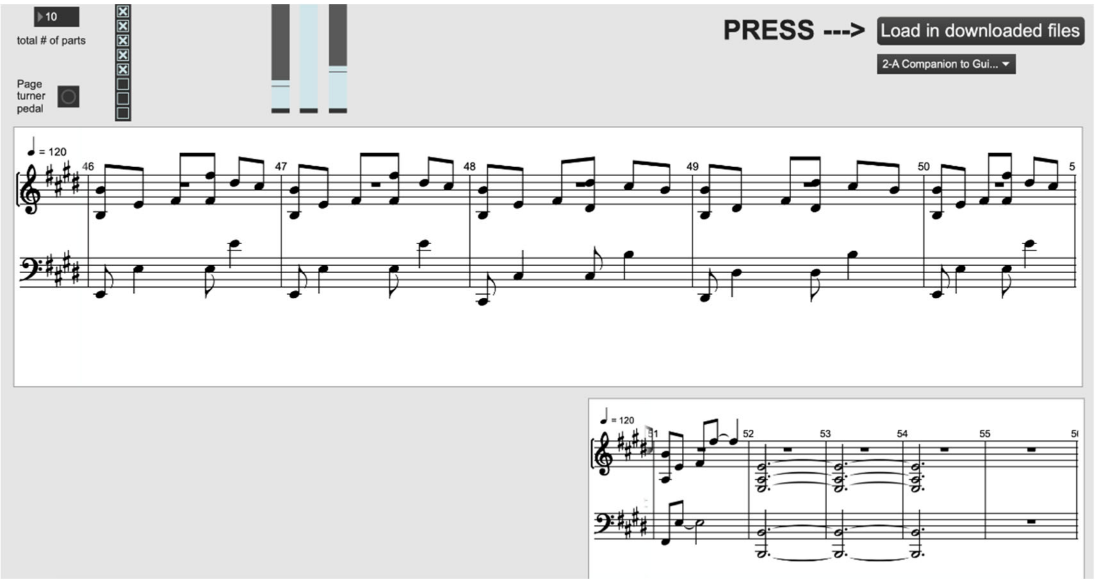


Figure 3.7 Example of the musician's reading interface (piano only)

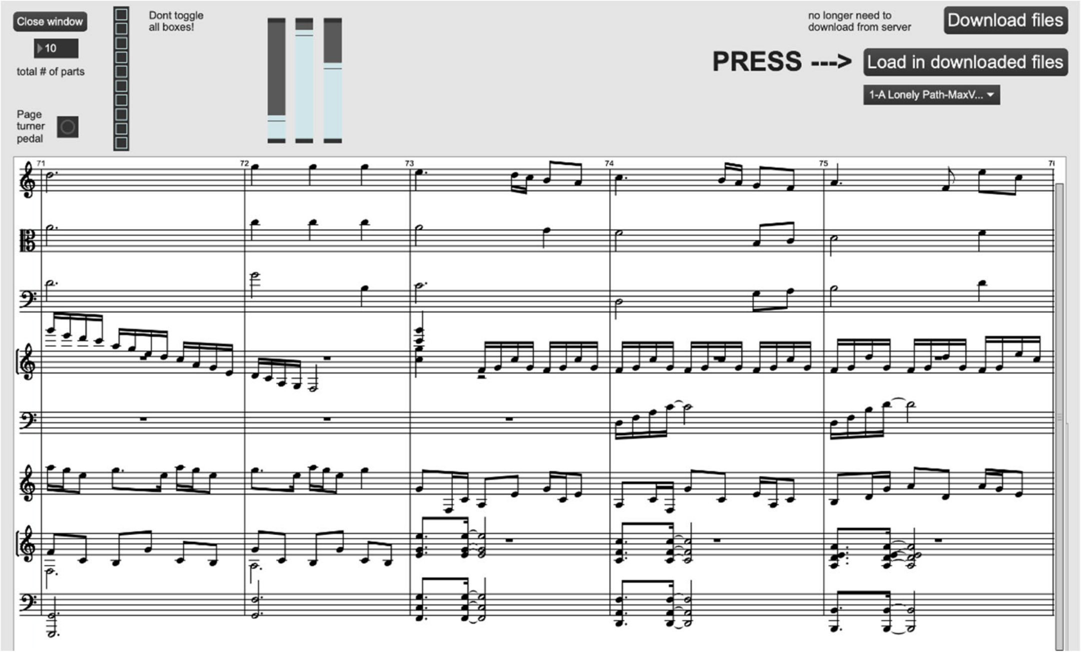


Figure 3.8 Full score interface for the conductor's use

This system of creating presets and being able to send them to my musicians at will works quite well. It is not without its drawbacks, however. One glaring issue is that not all of my musical markings are transferred to the bach.score object. As the barren figures might suggest, important details like dynamic, articulation, and phrasing markings are completely removed. This essentially removes all visual direction from the music, making it that much harder for musicians to interpret the music effectively. Another frustrating limitation is that 8va/8vb or octave-changing markings are not supported by the *bach* plugin. This means that including wide ranges of register, primarily for piano and harp, is not viable because it will require the musician to read from many leger lines, which is not reader-friendly for musicians. Another limitation is that the bach.score object does not give any formatting options. This leads to some spacing deficiencies in the music that can once again make it difficult or distracting to read. My remedy to these limitations has been to prepare the music with the musicians well in advance so that they are familiar with the music and all of its details.

CHAPTER FOUR

Approach to Music

For this undertaking I chose to rescore the video game *RiME* (Tequila Works, 2017). *RiME* is an adventure platformer game that follows a young boy as he explores mysterious ruins and lost palaces. Along the way you solve puzzles and uncover hidden secrets. The ending is very powerful as you learn that the game is a metaphor for grief. Consequently, *RiME* had a profound effect on me and this is partly why I chose to utilize it. However, the main reason is that the game provides a style of gameplay that makes extensive use of its soundtrack, which is used to narrate, underscore, and inform. I decided that I could create my own music and adapt it for the same purposes, using the gameplay as a canvas. The process of creating music for this game was very new for me as someone who has mainly composed linear concert music. I was tasked with writing music that served different roles, such as being narrative, informative, and background-focused. I also needed the music to be interesting and engaging but not overly technical. I also needed to ensure that I gave the musicians some amount of agency since interactivity was a critical component. In the following sections I will discuss how I addressed each of these challenges.

My first directive was to ensure that my music fit the aesthetic and pace of the game. *RiME* is very vibrant but slow paced. It gives the player time to exist and soak in their surroundings. I wanted to create a soundscape that was reverberant and varied so I chose to write music for violin, viola, cello, harp, piano, and percussion. This ensemble gives the possibility for a sound that can be large and strong, as well as subtle and delicate. I proceeded to compose music that was varied in character and texture, keeping in mind elements of the game and its general aesthetic. At this point I didn't know exactly where parts of my score would fit in the game but I knew that I needed contrast; low intensity music and high intensity music, sentimental music and light-hearted music. I also wanted my music to be strongly based on motives since I wanted to use it to represent various things in the game, such as characters and objects. I ended up creating two compositions that would serve as the source material for the game's new soundtrack, *A Lonely Path* and *A Companion to Guide You*.

4.1 Music as Narrative Force

For this demonstration I chose to utilize part of the first level of *RiME*. I then tasked myself with identifying key *narrative moments* of the level such as cutscenes and progressing on a puzzle. For example, there is an important cutscene in the level where the boy meets a fox companion that guides him through the rest of the game. This meeting is the first major event in the game so I wanted to put music over it that helped the scene propel the narrative. Another example of a moment like this is when the player activates statues that progress the current puzzle. Again, I utilize music in this moment that signifies progression to the player. How exactly does the music possess a narrative quality? In general, I use music that is rhythmically strong, melody-focused, and capable of standing on its own. Incidentally, that is exactly how you would describe the two pieces *A Lonely Path* and *A Companion to Guide You*. More practically speaking, whenever a narrative moment in the game occurs I prompt the musicians to read directly from the score of either piece. However, each narrative moment is only 20-30 seconds long so only part of the score can be read. This is where I went into the score and essentially cut it into pieces to be used in specific parts of the game, taking into account its intensity level as well. Some narrative moments in the game are stronger than others so I utilized sections of music with varying levels of intensity in order to represent them. For example, when the player wakes up on the beach I see that as a narrative moment but not a particularly strong one. Therefore I use a snippet from *A Lonely Path* that features the cello playing a drawn out melody while the harp and vibraphone support it with a soft and consistent ostinato. An example of a stronger narrative moment is when the player activates the last statue, thus solving the first puzzle of the game. During this moment I want the music to evoke the feeling of accomplishment and of “unlocking the door” to the next step in the adventure. Therefore I use a snippet of *A Lonely Path* that is thick in texture, dynamically strong, rhythmically active, and melody-centric. In summary, whenever narrative moments occur in game, I have the musicians play a snippet of music, usually 20-30 seconds long, directly from the score. It is during these moments that the music is in its most active state.

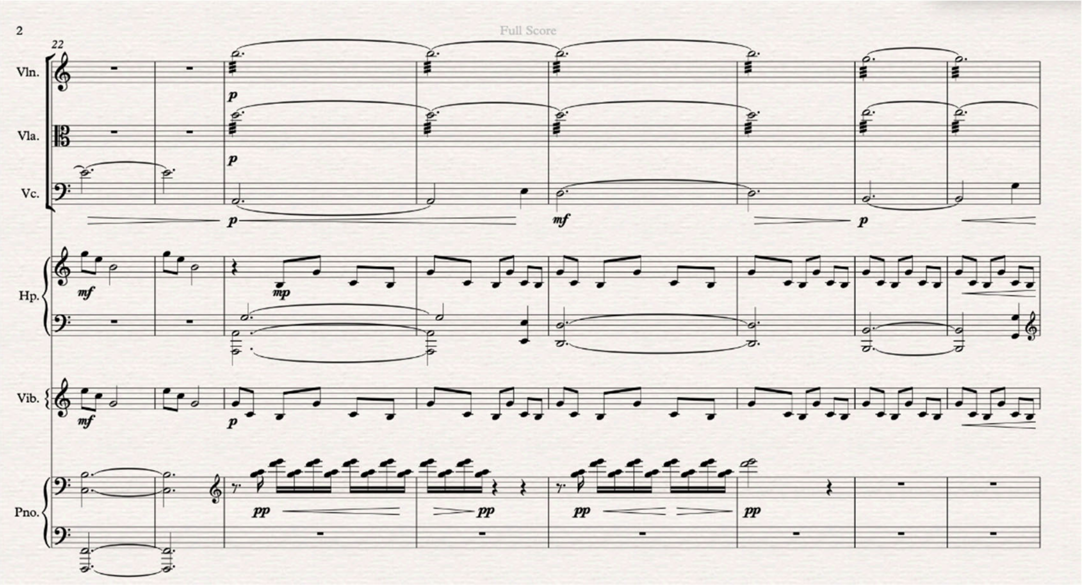


Figure 4.1 Music snippet that occurs during beach scene

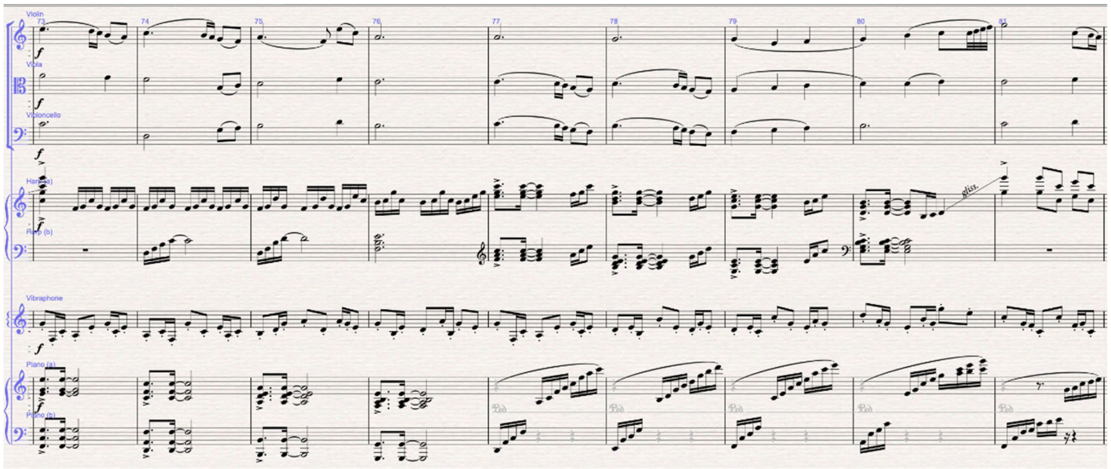


Figure 4.2 Music snippet that occurs after activation of last statue

4.2 Music as Background

The majority of the gameplay does not consist of narrative moments, however. The player spends most of their time walking around the level, exploring the environment and trying to figure out the puzzles. For this type of gameplay I didn't want the music to draw much attention to itself. Therefore its function is primarily background. I refer to these sections of ambience as Aleatoric Glue, something that essentially binds together the scattered narrative moments. This hints at the opportunity I saw to incorporate aleatory and improvisation, thus giving the musicians a chance to be creative and interact with the gameplay. During these sections of ambience I display to the musicians only mimimal guidelines for what to play. Each guideline is in the form of two displayed measures. The first measure consists of either a scale or chord, which communicates to the players the harmonies and key in which they should be improvising. The key is either A aeolian or B mixolydian, corresponding to the keys in the primary score. The second measure consists of thematic material, usually a melody that appears somewhere in the primary score. These ambience guidelines give the musicians room for creativity while maintaining the musical aesthetic I have created. The soundscape therefore remains homogenous across every moment of gameplay. I have also encouraged the musicians to pay close attention to the gameplay during these sections so that they may best contribute ideas that are contingent upon what the player is doing.

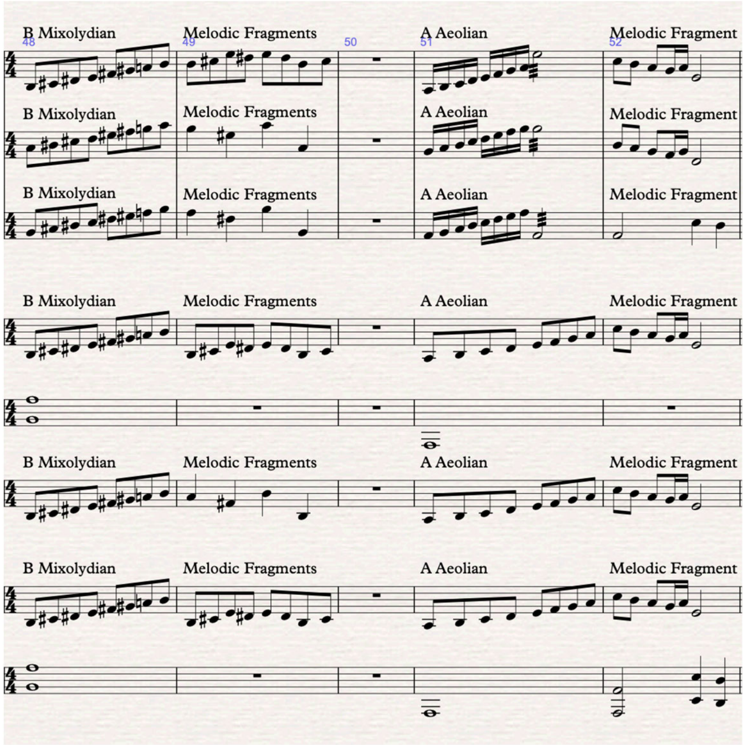


Figure 4.3 Example of two aleatoric guidelines for ambience sections

4.3 Musical Interjections

The last type of music I prepared is what I call *microcosms*. These 15-30 second long spans of music function as sort of a middleman between narrative and ambient music. They appear momentarily at times during the Aleatoric Glue, usually in order to signify some object or entity that the player has come across. In this way they exemplify music that communicates to the player. I composed out each microcosm, utilizing material from the primary score. They are essentially mini representations of the score (hence the name) and are thus heavily thematic and motive-driven. This is an important detail because the purpose of each microcosm is to represent a specific entity in the game. There are four microcosms that are each characterized by a specific theme: the Fox, the Hooded Figure, and Boy, and the Obelisk. One example of how I use a microcosm during gameplay is when the boy encounters the Hooded Figure. When this encounter occurs I prompt the musicians to play the microcosm associated with the Hooded Figure. The music is meant to rise in intensity briefly but not to the level of narrative music. Another example is when the player gets close to the Fox statue I prompt the musicians to play the Fox microcosm. By using the microcosms in this fashion I allow the music to narrate small moments in gameplay in a way that subtly communicates to the player that what they are seeing is important.

4.4 Approach to Performance

The music is constantly in a fluid state, flowing between each of these types of music: narrative, ambient, and communicative. My role as the conductor is to watch the gameplay, determine what music should be played, and communicate that to the musicians. I also have to ensure that the switches occur seamlessly, which is a challenge that requires me to be very familiar with my musical presets and fluent with the Conductor's Controls. It also requires that I communicate with the musicians in a clear and effective way. For this demonstration I have adapted techniques used in *soundpainting*, which is "the universal multidisciplinary live composing sign language for musicians, actors, dancers, and visual Artists."[2] Basically, I use certain gestures to indicate to the performers when I want them to improvise, how I want them to play entrances, and whom I want to play and not play. This last detail relates to the way I can choose to layer the music, namely the Aleatoric Glue and microcosms. During these sections I don't necessarily want every performer to be playing all the time. For instance,

[2]. "Soundpainting," Soundpainting. www.soundpainting.com/soundpainting/ (accessed March 5, 2020).

the aleatoric ambience music will always be made up of a combination of piano, harp, and percussion. I can play with texture in that combination alone, indicating if I want certain players to drop out or play out. However, there may be moments when I cue in the strings if I feel like the sound is needed. This decision-making is all in the moment and very much a part of how I as conductor get to improvise as well. Another important example of layering occurs during the microcosms. I wrote each microcosm in a "vertical manner", meaning that the music works with all of the parts as well as with only a few of them. I did this so that I can vary the way I actualize each microcosm in game. A good example is when the player encounters the Hooded Figure, which happens 2-3 times in the section of gameplay I am using. The idea is that when the player sees him the first time I will include the appropriate microcosm but with only cello and harp, for instance. Each consecutive encounter I will add in more instruments, adding a progressive element to the music. This ability to layer the music, another common game music technique, lets me do more with less music.

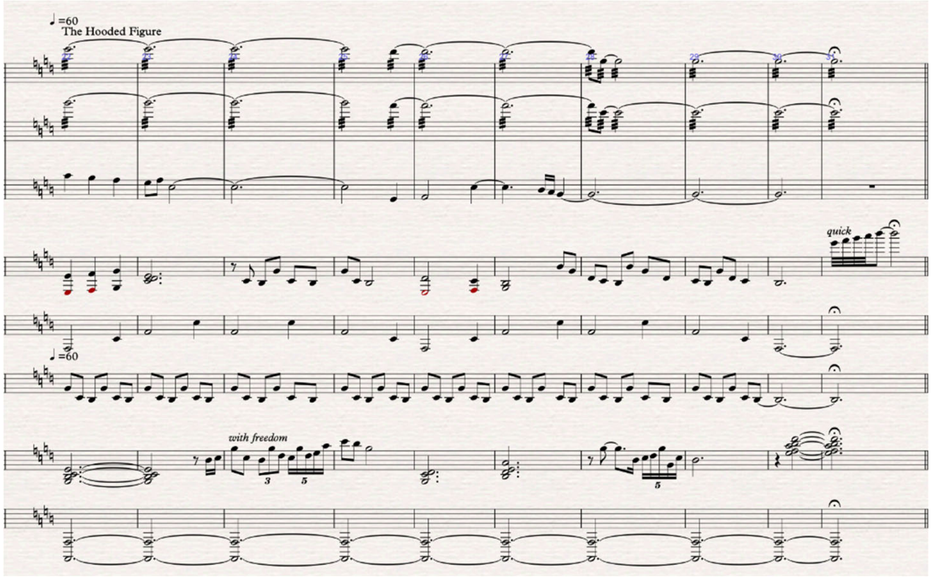


Figure 4.4 Microcosm for the hooded figure

CHAPTER FIVE

Conclusion

Unfortunately, the recent Coronavirus outbreak made having a final performance impossible. Therefore, this paper merely represents a detailed plan of how the performance would be realized. However, this is an ongoing project that I will continue to see through to completion after my academic studies have concluded. My work up until now has proved that technology can be used to incorporate dynamic game music into live performance, thus accomplishing one of my central goals. However, in future performances what I hope to discover is that this type of performance/interaction is engaging and compelling. Given that that is true, my plan is to improve upon the technology that supports it. My next goal is to create an accessible application that simplifies the process of scripting sheet music and displaying it to performers. My vision is that anyone will be able to use this technology to create a performance/interaction like I have, using any type of music and game. I believe that hybridizing interactive video game music and live performance has practical potential as well, from providing musicians with more performance opportunties to creating new methods of music education. Whatever the application, my aim is to collide worlds and create unique opportunities for collaboration. My hope is that as a result, there will be an increased appreciation for both video games and the arts.

REFERENCES

"bach: automated composer's helper." Bachproject. March 1, 2020. https://www.bachproject.net.

"Soundpainting." Soundpainting. March 5, 2020. http://www.soundpainting.com.