Soundscape

Virtual Reality Audiovisual Experience

Megan E. M. Washburn

Liberal Arts and Engineering Studies:

Computer Graphics and Music

California Polytechnic State University

12 December 2018

**Soundscape:  
Virtual Reality Audiovisual Experience**

Megan Washburn

Liberal Arts and Engineering Studies:

Computer Graphics and Music

Senior Project Advisors: Michael Haungs and David Gillette

California Polytechnic State University

San Luis Obispo, CA USA

12 December 2018

Contents

[**1 Introduction** 4](#_Toc532547226)

[1.1 Problem Statement 4](#_Toc532547227)

[1.2 Project Goal 4](#_Toc532547228)

[**2 Application Specifications 4**](#_Toc532547229)

[2.1 Usage 4](#_Toc532547230)

[2.2 Required Hardware 5](#_Toc532547231)

[**3 Background** 5](#_Toc532547232)

[3.1 Technical Information 5](#_Toc532547233)

[3.2 Musical Information 6](#_Toc532547234)

[**4 Design** 6](#_Toc532547235)

[4.1 VR Structure in Unity 6](#_Toc532547236)

[4.2 Structure of Audio 6](#_Toc532547237)

[4.2.1 Background Ambient Music Compositions 6](#_Toc532547238)

[4.2.2 Sound Effects for Interactable Objects 7](#_Toc532547239)

[4.2.3 MIDI Arpeggiator 8](#_Toc532547240)

[4.3 Modeling Software for Assets 8](#_Toc532547241)

[4.4 Version Control 9](#_Toc532547242)

[**5 Implementation** 9](#_Toc532547243)

[5.1 Unity Development Kits 9](#_Toc532547244)

[5.1.1 Oculus Avatar SDK 10](#_Toc532547245)

[5.1.2 Unity Terrain Engine 10](#_Toc532547246)

[5.2 Audio Visualizer 10](#_Toc532547247)

[5.2.1 Fast Fourier Transform 10](#_Toc532547248)

[5.2.2 Bandwidth Modifier 11](#_Toc532547249)

[5.3 Ableton Live 10 Music Editing 11](#_Toc532547250)

[5.4 Obstacles 11](#_Toc532547251)

[5.4.1 Platform Capabilities 12](#_Toc532547252)

[**6 Analysis and Verification** 12](#_Toc532547253)

[6.1 Playtesting 12](#_Toc532547254)

[6.1.1 Reviews 12](#_Toc532547255)

[6.2 Industry Specialist Review 12](#_Toc532547256)

[**7 Interdisciplinary Connections** 12](#_Toc532547257)

[**8 Related Work** 13](#_Toc532547258)

[8.1 VR: Music Visualizers / Interactive Applications 13](#_Toc532547259)

[8.1.1 Playthings VR 13](#_Toc532547260)

[8.1.2 Beat Saber 13](#_Toc532547261)

[8.1.3 Raybeem VR Music Visualizer 13](#_Toc532547262)

[8.2 Music Visualizers / Interactive Musical Applications 14](#_Toc532547263)

[8.2.1 Crypt of the NecroDancer 14](#_Toc532547264)

[8.2.2 Guitar Hero / Rock Band 14](#_Toc532547265)

[8.2.3 Patapon 14](#_Toc532547266)

[8.3 Personal Previous Work 14](#_Toc532547267)

[8.3.1 Band Wagon 14](#_Toc532547268)

[**9 Future Work** 15](#_Toc532547269)

[9.1 Thesis Lead-In 15](#_Toc532547270)

[**10 Conclusion** 15](#_Toc532547271)

[**11 References** 15](#_Toc532547272)

# Introduction

## Problem Statement

Current audio implementation in most VR experiences is underutilized. This project aims to demonstrate the power of audio when developing a VR environment or experience. By setting up an environment for a user to interact and experiment with, this project aims to achieve a deeper impact on the user via interactive visual and audio cues.

## Project Goal

To develop a visually and auditorily immersive virtual reality experience to explore the capacity of audiovisual components of VR environments. The user is able to interact with virtual instruments in the environment via controller input and microphone input, to generate correlated audiovisual output in the environment.

# Application Specifications

## Usage

For those who are familiar with virtual reality environments, Soundscape is straightforward to play during its initial Daytime stage. The objects are placed in accessible locations in the scene, to ensure that each musical functionality can be explored with ease by the user. The similarity to keyboard instruments and simple layout invites the player to move their Oculus hand controllers to interact with the objects in the scene, which serves as the main mechanic to use this application.

During Soundscape’s Nighttime stage, players will strike floating lotuses in time with the music; they emanate from crystal objects that react in time with the selected song, serving as audio visualizers (using FFT calculations). To move forward in the list of pre-programed songs, players can either press the “A” button on the Oculus controller or “A” key on the keyboard. To move backwards in the list, players can similarly press the “X” button or key on the Oculus controller or keyboard respectively.

##### Skip Song Forward

##### Skip Song Backward

Figure : Illustration of where the "A" and "X" buttons are on the Oculus controllers to swap songs during Nighttime setting.

When equipping the Oculus hardware, there are a few adjustments to keep in mind to achieve comfortability while wearing and operating the head mounted display (HMD). Along the sides of the HMD lying over the temples and running along the crown of the player’s head are three Velcro straps that can be adjusted to comfortably secure the headset to the wearer’s head. The on-board headphones can be flipped up and back if the wearer so desires. As Soundscape potentially lends itself to rapid motions with the hand controllers, it is recommended to fasten the controller straps around the player’s wrists for the duration of play.

## Required Hardware

To experience Soundscape smoothly, recommended hardware specifications are as follows: [1]

* Graphics Card: NVIDIA GTX 1060 / AMD Radeon RX 480 or greater
  + Alt GPU: NVIDIA GTX 970 / AMD Radeon R9 290 or greater
* CPU: Intel i5-4590 / AMD Ryzen 5 1500X or greater
* Memory: 8GB+ RAM
* Video Output: Compatible HDMI 1.3 video output
* USB Ports: 3x USB 3.0 ports, plus 1x USB 2.0 port
* OS: Windows 10 or newer

Oculus provides a tool to verify your machine’s compatibility; see references for the link to this tool. [2] Once the requirements listed above are satisfied, Soundscape is ready for the player to explore and enjoy.

# Background

Current audio implementation in a majority of VR applications is underutilized; this project aims to demonstrate the power of audio when developing a VR experience. By setting up an environment for users to experiment and play with, this project explores means to achieve a deeper impact on the user via interactive visual and audio cues.

## Technical Information

In my creation of Soundscape, I call upon a variety of technical concepts to achieve the effects I intended. For example, I required Fast Fourier Transforms (FFTs) [3] to implement an audiovisualizer that responds in real-time to given audio input. FFTs essentially are various algorithms that compute discrete Fourier transform (DFT) of a sequence or its inverse (IDFT). Fourier analysis converts a signal (e.g. a sound wave) from it’s original domain (e.g. time or space) to a representation in the frequency domain, and vice versa. [4] This allows the numerical representation

Virtual reality (VR) is an interactive computer-generated experience within an entirely simulated environment (not to be confused with augmented or mixed reality (AR and MR), which include captured elements of the real world in which the experience is taking place). Current VR technology uses head-mounted displays (HMDs) or multi-projected environments, occasionally in combination with physical props, to better improve immersion to the virtual environment. [5]

## Musical Information

The theory behind my choices of which tones to include in the scaffolding of the virtual instruments for the user to play is based in both Western and Japanese music theory. While resolution of chord progressions is heavily relied on in Western music theory, the lessened reliance on chordal resolution in Japanese music theory lent well to the continuous creative mode in Soundscape, to allow users to flow while creating their music. Pentatonic scales, given that they consist of only five notes, lend well to improvisation, therefore mesh well in most combination situations. I selected scales with major notes for brighter scenes (such as daytime), and scales with minor notes for darker scenes (nighttime). [6]

# Design

## VR Structure in Unity

Given the multi-component nature of this project, I wanted to be sure to start with a solid organizational structure to maintain sustainable development. Previously, I’ve worked on game projects that too often end up with a bloated file structure and outdated asset libraries, that ran into issues with versioning during concurrent development. Therefore, I kept in mind some recommended best practices for organizing Unity project directories going into this project [7]; these recommendations for structuring have served me well up through the close of development on this project.

In terms of porting the project to VR, there was not much else needed in the backend other than to implement the joint documentation for Oculus and Unity developers’ Oculus Utilities for Unity. Using the OculusVR Plugin (OVRPlugin) [8] made the conversion from development in a single-screen format to a virtual reality environment seamless, when followed correctly. Again, I must reiterate the importance of access to the correct tools and hardware; after transferring development to a machine better suited for VR development (as noted by Oculus’s specification requirements [9]), the rate of productivity and creation greatly increased.

## Structure of Audio

While arranging in Ableton Live 10, I wanted to keep my compositions organized not only for ease of development, but to smoothly integrate changes when I ported over the compositions to Unity. Of the variety of recommended song structures, I decided to take inspiration from pieces of Ableton’s guide on arrangements [10].

FMOD studio greatly aided in development, as this sound effects engine and authoring tool is primarily designed for video game scoring and sound design. [11]

### Background Ambient Music Compositions

When I initially began production on Soundscape, I had compiled a few short compositions intended to be looped as background music for the player to use as scaffolding or as a guide when playing with the virtual musical objects. Though, after a few short rounds of playtesting, I determined that leaving more space for the player to inject their own creative direction – subtracting the structured musical direction I was suggesting in the scene – allowed the player to have more agency over the audio that they were generating, and therefore have a more enjoyable experience with the application.

With that, I reduced the background music to an ambient repeating loop. This loop still employs a basic chordal structure, but has no set rhythmic pattern, to allow the user to play with any cadence they prefer. This also lent better to the arpeggiated musical objects I implemented, so there was more pause in the generated audio yet still without being too sparse so not to lose the player’s interest. For this background music, I used audio clips of singing bowls: deep metallic bowls that produce calming, and in some variations, eerie, overtones when the rim is traced with a wooden mallet, in a similar manner to tracing a crystal wine glass with your fingertip or to how an armonica is played.

For added spacey effect, I applied a panning audio script to the audio sources emitting the singing bowl clips. Again, Unity ensures that this implementation is smooth, as panning the audio from one ear to the other is controlled by a public variable within the AudioSource class. I made simple sine calculations within the range [-1, 1] based on time elapsed to get a smooth, sinusoidal passing of the audio from one ear to the other.



Figure : Location of audio source for singing bowl clips. Panning of audio is handled via script rather than object transform.

### Sound Effects for Interactable Objects

When designing the sound effects to be emitted by the interactable objects, I needed to ensure that each audio clip blended well amongst each of the interactable classes, as any order and frequency of activation is allowed during gameplay. With this design consideration in mind, I decided to use mellow audio clips from acoustic instruments such as classical guitar, harp, and hang drums. Each instrument I selected have a specific set of associated overtones that create pleasant harmonies when combined. Overtones are frequencies other than the dominant frequency of the note; the following diagram illustrates harmonic overtones that are generated on the strings of an instrument. [12], [13] Though I could not simulate in real-time the physical properties that naturally generate these overtones in Unity, I instead procured various recordings that incorporated subtle variations in these overtones that achieve a similar effect to the actual physical generation of these overtones. For diagrams and explanations of overtones, see the following reference: [14].

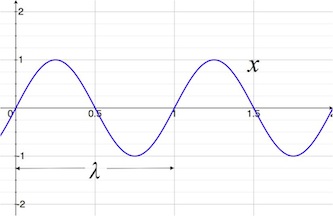
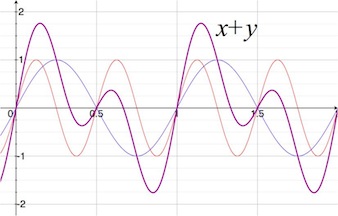
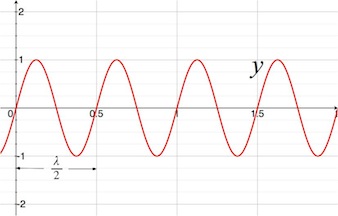
 

Figure 3: Illustration of two sinusoidal waves, x and y, followed by their combination with harmonic overtones illustrated as overlying waves to show how the resulting wave is influenced by the component waves.

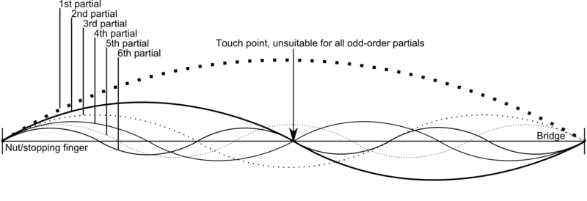


Figure 4: Illustration of physical effect of dividing a string on any instrument to elicit the corresponding harmonics. Each subdivision corresponds to a new, specific partial, each generating unique overtones for the respective partial.

### MIDI Arpeggiator

I composed audio to feed into Unity scripts that allows the player to automatically step through a sequence of notes based on their input, thus creating an arpeggio.[15] By converting the floating stones to triggers, the players can layer arpeggiations to build their musical work.  
  
Allowing this kind of leniency for the player in triggering sounds required implementing musical structure, to ensure that most combinations would still sound pleasing to the ear. To do so, I composed audio clips using standard arpeggiated patterns [16] that allowed both freedom in creation for the user, yet substantial structure to yield sounds that would be pleasing to the Western ear. [17]

## Modeling Software for Assets

I primarily used both free assets available on TurboSquid [18] and CGTrader [19], as well as modifying assets on my own to suit my needs for the game. I primarily used Blender [20] to modify assets and generate normal for meshes (for improved lighting calculation capabilities), though occasionally used Maya [21] to make smaller edits to the more organic shapes I used.

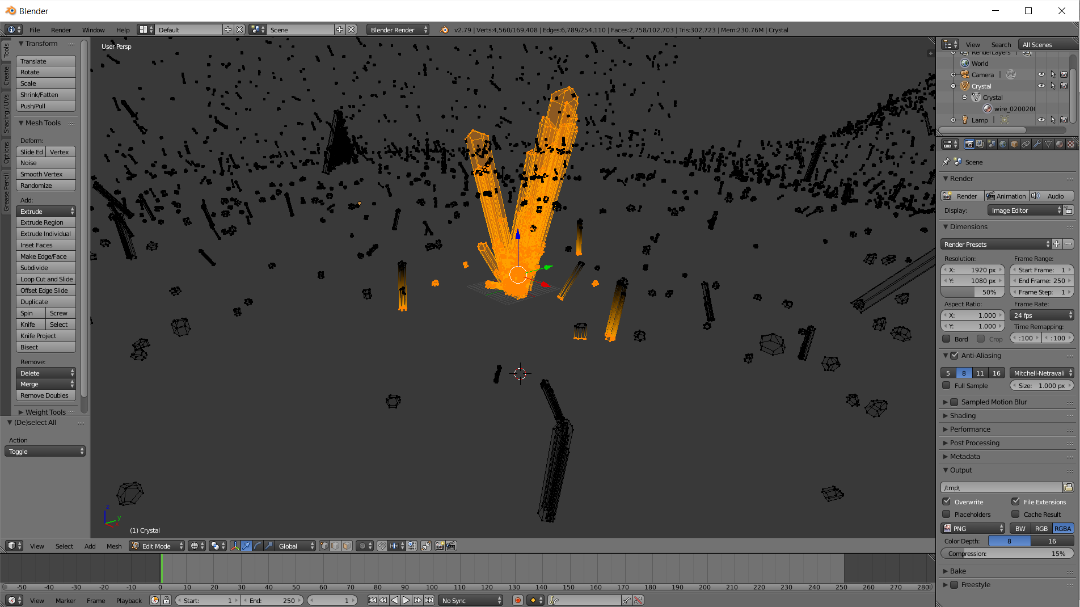


Figure : Example usage of Blender; using the vertex selection tool to edit a free 3D crystal asset pack.

## Version Control

For ease of personal record keeping and the ability to revert to prior editions in the case of fault, I used GitHub [22] to document my changes and have a reliable backup of version of my project. With a properly configured .gitignore file, I was able to develop from multiple machines as needed. As I was

# Implementation

Overview of the development process, including technologies used and the nuances of working with these technologies. Structure of this section is largely chronological in terms of my work process.

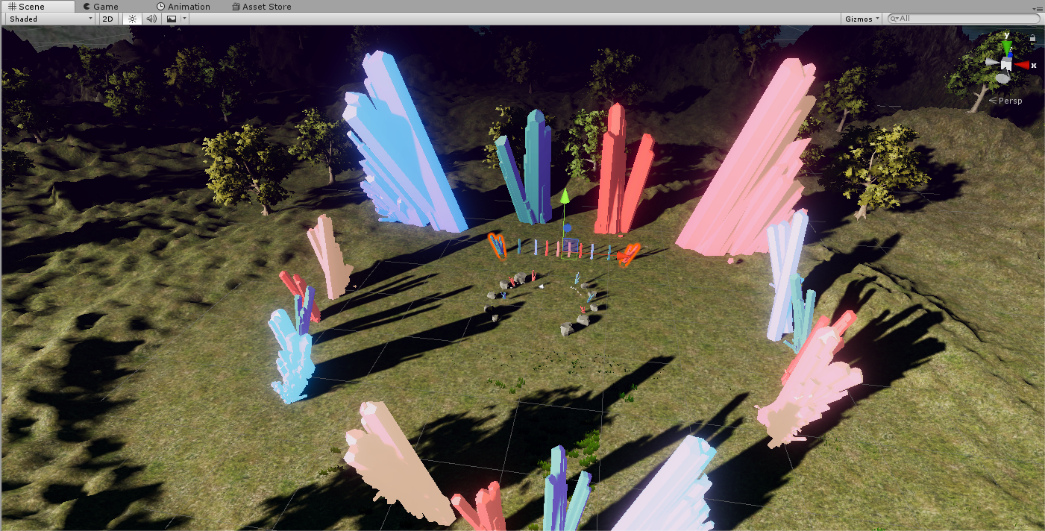


Figure : Pictures of editing view of Nighttime scene in Soundscape.

## Unity Development Kits

Unity is a robust engine that provides much functionality ranging from collision detection to post-processing effects in their included libraries. I used these libraries for both previously mentioned functionalities, as well as for event listeners and response, audio listeners and management, as well as for modelling and management of the objects in the virtual environment.

### Oculus Avatar SDK

This software development kit was a great assist in developing this virtual reality environment. Collision detection with the OVRAvatar did not come easily. Though this development kit handled communication with the Oculus hardware cleanly, basic operations such as collision detection were not built in. I therefore modified the provided scripts to behave how I needed them to: I programmed the included bone meshes within the OVRAvatar’s hand to each collide with the objects I marked as musical objects.

### Unity Terrain Engine

To model the surrounding environment, I used Unity’s robust built-in terrain engine. The editor includes tools to raise and lower the terrain mesh, level the mesh, add water in local minimum vertex locations, paint textures, and add and remove trees or grass. The editor includes a variety of brush presets to paint the vertex locations, tree or grass locations, and textures. Custom brush patterns can be loaded in via 2D grayscale images, a functionality similar to height mapping but on a more granular level. Height mapping takes in a grayscale image representing heights (lighter spots corresponding to higher locations, darker to lower),

The engine driving creation of trees for the terrain is equally robust; developers also are given the option of modeling their own trees using the built in SpeedTree Modeler (form IDV, Inc.) [23] to create trees with advanced visual effects such as smooth LOD transition, fast billboarding, and natural wind animation.

## Audio Visualizer

### Fast Fourier Transform

Analysis of a provided audio source using FFT is built into Unity’s SDK [24]; as these calculations can be difficult to execute manually, this functionality greatly speeds up my process for implementing audiovisualization based on varying frequency bands in real-time. This allowed me to seamlessly integrate real-time audiovisualzation into Soundscape, without the hardware and mathematical overhead of grabbing data from an input audio clip by hand.

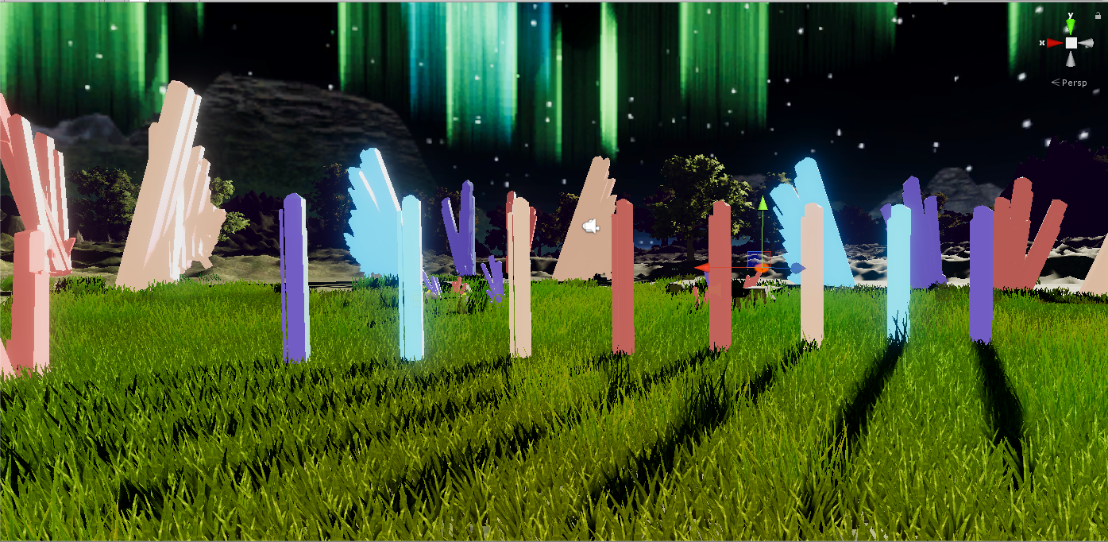


Figure : Picture of the eight main objects that FFT band with masks were applied to modify height based on real-time audio.

### Bandwidth Modifier

To achieve a less granular representation of frequencies, I applied masks for specific bandwidths. Taking all the measured frequencies within a given range and aggregating the data, I was able to represent a smooth range of high and low frequencies. To further smooth the representation of frequencies, I implemented a height buffer to slowly decay from the last local max height for the crystal bar visualizers.

## Ableton Live 10 Music Editing

I used Ableton Live 10 for both sketchups of musical ideas I had for the project, as well as for more involved development of the sound design for this game. Ranging from simple sound effects to fuller, multi-track compositions, Ableton served as an invaluable tool for smoothly developing the auditory landscape I had envisioned for this project. I plan to further refine my composition and sound design skills with this tool to build off of the techniques I developed for the purpose of this project.

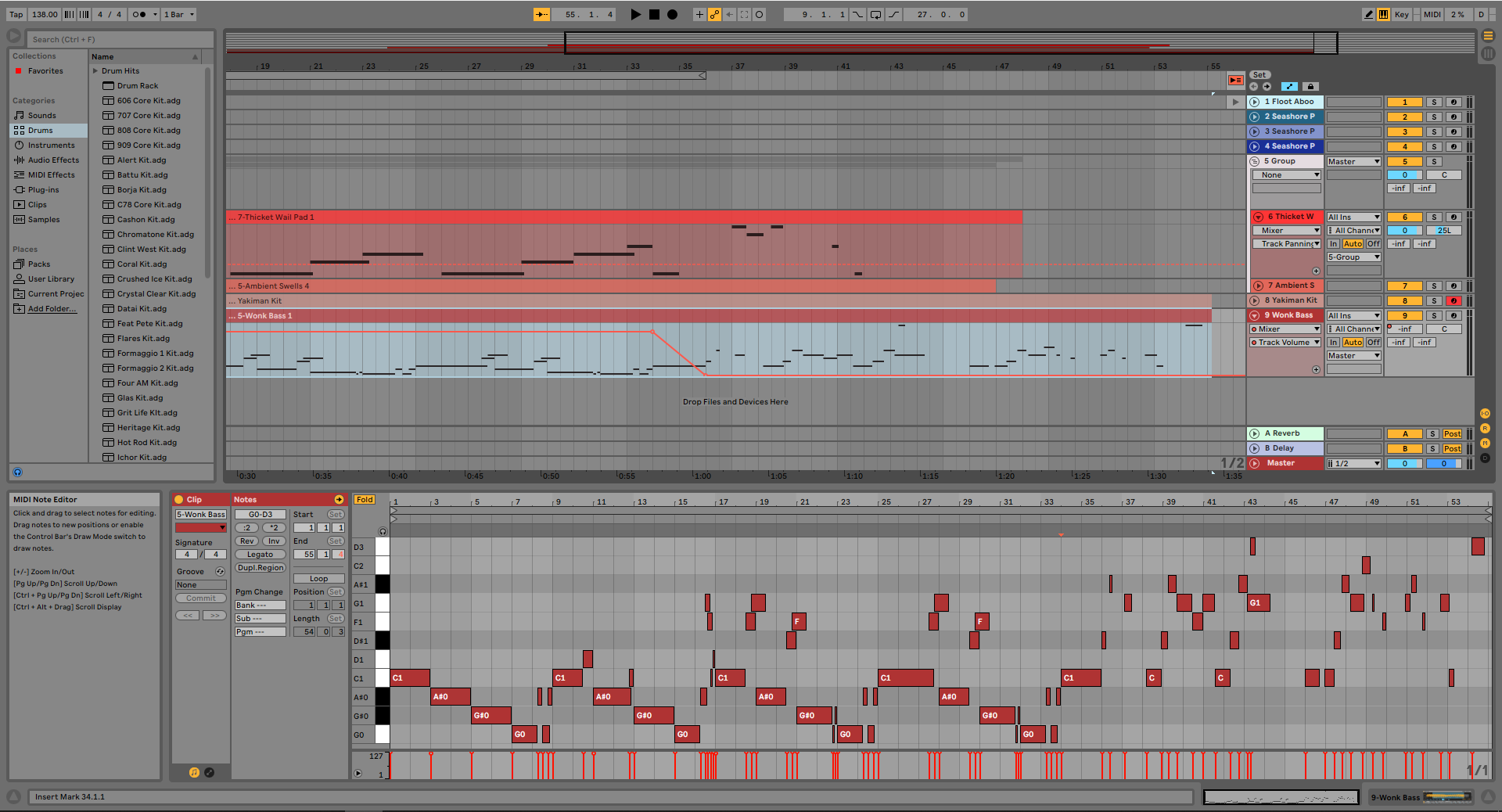


Figure : Example of my layout for composing songs in Ableton Live 10.

## Obstacles

This section documents the most notable impediments that arose during development, from platform issues to nuances of designing for virtual reality environments.

### Platform Capabilities

Initial development was conducted on a MacBook Pro, Mid 2012; though a versatile machine, due to the architecture and design The Oculus VR development kits [8] were able to be installed and developed with in the Unity version of MonoDevelop, but this machine has no support for running the actual headset. Developing in this manner was a tedious process, as testing could only occur at a much slower rate.

For example, the standard application post-processing effects – such as chromatic aberration and motion blur included in Unity’s post-processing plugin – immediately caused discomfort once displayed in the Oculus headset. When testing on-screen, these effects seemed harmless; it is important to note that discrepancies in quality can be easily overlooked if development is not concurrently checked in the VR environment.

Though with further research, I found that a more specific implementation of post processing allowed for the aesthetic effects that Unity’s Post-Processing Stack can achieve without the discomfort of the normal application. Within my OVRCameraRig (child of OVRPlayerController), I attached a post-processing profile to the CenterEyeAnchor. With this, I could use post-processing effects such as bloom and color grading without causing as much of a nauseating sensation when viewed via the Oculus headset.

# Analysis and Verification

Playtesting and qualitative comparison are the main means of analysis of the final version of Soundscape. Verification of design and code continued concurrently throughout development, with the help of Visual Studio’s debugger tool and various online and in-person sources.

## Playtesting

I intend to conduct user surveys in the future, when my application is better polished. In this section, I document a few first-hand reviews given after testing Soundscape by volunteers.

### Reviews

First-hand accounts.

## Industry Specialist Review

David Washburn et al.

# Interdisciplinary Connections

The design of this application required a frequent connection between programmatical structure and audio and visual design. To create the system, audio, and visual assets for a smooth, well-functioning game, in-depth knowledge of computer science, music, and computer graphics is absolutely necessary. Development required nimble switches in mental paradigms from musical structure, to visual design in modeling software, to software architecture considerations. A majority of the content for my game was developed using creative software, and the most obvious interdisciplinary connection I made was in the scripting I did to direct the software I used to perform the custom tasks I specified them to do.

This project afforded me the opportunity to strengthen the interdisciplinary skills I have crafted over my career in Liberal Arts and Engineering Studies, a practice I absolutely intend to continue and afford myself more chances post-graduation.

# Related Work

In this section I discuss previous work that inspired this project and served as examples of what virtual reality musical experiences can achieve. I describe applications ranging from simple music visualizers to interactive music-based games; I include discussion on the specific features of each that inspired the work behind Soundscape, as well as ideas for future work.

## VR: Music Visualizers / Interactive Applications

The following sections catalog previous work done in virtual reality for music visualization, or in interactive musical experiences. To varying degrees of complexity, interactivity, and visual and auditory intensities, the following examples proved to be great sources of inspiration.

### Playthings VR

This application served as the main inspiration behind Soundscape, in the way it presents a whimsical musical VR experience for users to bring their own creativity to life. Set up like a virtual MIDI keyboard in some levels, others like a sequenced game similar to Guitar Hero, Playthings VR serves as yet another strong example of effective, impactful music in a virtual reality environment. The manner of interaction in which players have more agency over the state of the game is one of the biggest motivating factors behind the creation of Soundscape; I intended to have a similar mechanic, but to place it in a mellower environment whose tone would be closer to meditative rather than playful. [25]

### Beat Saber

A sensational application released in early 2018, this VR game did a great deal for audio in virtual reality by means of showing its impactful potential for users. It is a VR rhythm game, in which the player’s goal is to slash the beats which fit perfectly into precisely handcrafted music. Inspired heavily by Star Wars (hence the light sabers) as well as modern EDM music, this game culminates to a well-done, easily replayable, interactive musical VR experience. [26]

### Raybeem VR Music Visualizer

This application is the first VR project from the company Sokay. Though not necessarily a game (the traditional output of this company), this application can be interactive or non-interactive, as well as realistic or abstract in design. This project inspired both ideas for visual design, but more so the idea that I wanted my project to similarly be a relaxed environment for the user to primarily decide how it is to be experienced. Raybeem initially was intended to be a VR music visualizer, just to be a neat way to listen to music. It eventually evolved to include much more robust modes referenced as “Themes” by the developers. [27]

## Music Visualizers / Interactive Musical Applications

### Crypt of the NecroDancer

Developed by Brace Yourself Games, Crypt of the NecroDancer is an “award winning hardcore roguelike rhythm game.” Players must move to the beat to help your character (a funky tomb raider, in essence) to success in the various dungeons of the game. The electronic soundtrack by Danny Baranowsky is exceptional, specifically designed for an interactive musical experience. The game also allows for use of the player’s own personal MP3 collection, taking the replayability of the game to an entirely new level. Both the enjoyable mechanic and strong audiovisual design served as solid inspirations for my own project throughout development. [28]

### Guitar Hero / Rock Band

These games both were trailblazers in audio-centric rhythm-based games, and therefore both inspired much of the gameplay ideas I included and attempted during development of Soundscape. These games were careful to both showcase the selected repertoire of music, as well as leave players feeling substantially challenged yet still enjoying the gameplay. They also give players the opportunity to develop their gameplay to more of a skill rather than mere play, while still maintaining interest for casual gamers in the format of lower difficulty levels. [29], [30]

### Patapon

A fresh take on the mechanics necessary for a rhythm-based game, this format of call-and-response called for a novel take on the sound design for the game. Resulting in a uniquely enjoyable and replayable game, Patapon served as an abundant source of inspiration while developing for Soundscape. It is an “innovative 2D platform / action game in which players are tasked with leading the colorful Patapons – a brave and noble tribe – through a series of epic battles against opposing armies and gigantic monsters.” Players queue actions via drumbeats to march, attack, and defend with the Patapon tribe in this distinctly unique rhythm game. [31]

## Personal Previous Work

### Band Wagon

Prior (and concurrently) to developing Soundscape, I contributed in development to a rhythm-based game developed entirely in Unreal Engine 4 [32]. Programming rhythm-based mechanics helped develop familiarity with game programming with audio as the premier attribute to the application. This project also greatly helped familiarize with FMOD studio, a tool that proved to be very valuable to the functionality of Soundscape’s interactive musical aspects.

#### Sound Design and Soundtrack

Between Ableton Live 10 [33], Studio One [34], and Garage Band [35], I had accumulated ample experience designing audio for an interactive experience. Viewed too often as independent entities in the game development process, I concurrently crafted both the soundtrack and audio cues to mesh with one another.  
  
In terms of soundtrack, creating a background track that was repeatable without being distracting yet while avoiding becoming overly repetitive was key.  
  
Sound effects (SFX) were integral to this game as it was one of the main cues the player got that indicated their performance. Player performance and enjoyment was significantly improved with refined SFX rather than solely visual cues. Adding subtle cues and balancing with the current cues, rather than increasing – and thereby risking overdoing – different cues to the player yielded better results during playtesting.

# Future Work

## Thesis Lead-In

In my work on my thesis in pursuit of my Master’s degree in computer science, I plan to implement (and improve upon an existing) tool that procedurally generates music based on any given game, and to inject this tool into Soundscape. The previous work done on this tool was used in a game jam – a design sprint-like event in which participants intensively collaborate on creating games – to generate music on the fly for the games being created, given a few parameters of tone and energy as specified by the gam jammers.

I plan to develop on this previous work to generate music with varying tonalities of audio, and to incorporate deeper scaffolding of music theory, aiming to generate music that is more lyrically and harmonically pleasant.

# Conclusion

Soundscape served as a strong platform for me to develop my programming, composing, and artistic and game design skills; with that, I plan to continue development of such projects towards the goal of conveying messages that are strengthened by being created in this interdisciplinary format. This application successfully demonstrates the potential of audio in interactive virtual environments, and hopefully serves as an environment for users to retire to for either meditative relaxation, or creative enjoyment.

# References

|  |  |
| --- | --- |
| [1] | Facebook Technologies, LLC., Unity, Inc., "Oculus Support," Facebook Technologies, LLC., 2018. [Online]. Available: https://support.oculus.com/170128916778795/. |
| [2] | Facebook Technologies, LLC., Unity, Inc., "Oculus Rift Minimum Requirements and System Specifications," Unity, Inc., 2018. [Online]. Available: https://support.oculus.com/248749509016567/. |
| [3] | Wolfram Research, Inc, "Fourier Series," Wolfram Alpha, 29 November 2018. [Online]. Available: http://mathworld.wolfram.com/FourierSeries.html. [Accessed 2018]. |
| [4] | Wikipedia, "Fast Fourier transform," Wikipedia, 6 December 2018. [Online]. Available: https://en.wikipedia.org/wiki/Fast\_Fourier\_transform. [Accessed 2018]. |
| [5] | Wikipedia, "Virtual Reality," Wikipedia, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Virtual\_reality. [Accessed 2018]. |
| [6] | Wikipedia, "Japanese Musical Scales," Wikipedia, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Japanese\_musical\_scales. [Accessed 2018]. |
| [7] | Unity Technologies, "Large Project Organisation," Unity Technologies, 2018. [Online]. Available: https://unity3d.com/learn/tutorials/topics/tips/large-project-organisation. [Accessed 6 November 2018]. |
| [8] | Oculus VR, LLC, "Oculus Utilities for Unity," Facebook Technologies, LLC, Menlo Park, 2017. |
| [9] | Oculus Rift, LLC, "Recommended System Specifications," Facebook Technologies, LLC, Menlo Park, 2017. |
| [10] | Ableton Live, "Play with song structures," Ableton Live, January 2018. [Online]. Available: https://learningmusic.ableton.com/song-structure/song-structure.html. [Accessed 6 November 2018]. |
| [11] | Firelight Technologies, "FMOD," Firelight Technologies, Melbourne, Victoria, 2017. |
| [12] | E. O. Musikforschung Basel, "Cello Map," Fachhochschule Nordwestschweiz Hochschule fur Musik, [Online]. Available: http://www.cellomap.com/index/the-string/harmonics.html. |
| [13] | M. Read, "Music Stack Exchange," 28 August 2011. [Online]. Available: https://music.stackexchange.com/questions/3849/how-do-harmonics-work. [Accessed 2018]. |
| [14] | M. Paschko, "www.hangblog.org: A Documentation Project about PANArt, the Hang, and other Pang instruments," 4 September 2008. [Online]. Available: http://www.hangblog.org/discover-the-hangs-overtones/. [Accessed December 2018]. |
| [15] | WikiAuthor, "WikiAudio - Arpeggiator," 5 April 2018. [Online]. Available: https://www.wikiaudio.org/arpeggiator/. |
| [16] | J. Albaugh, "Musical Chord Progression Arpeggiator," CodePen.io, 2018. [Online]. Available: https://codepen.io/jakealbaugh/full/qNrZyw/. [Accessed 6 November 2018]. |
| [17] | J. Smith, "Dissonant tones only unpleasant to a Western ear," CMuse, 21 July 2016. [Online]. Available: https://www.cmuse.org/dissonant-tones-only-unpleasant-to-a-western-ear/. [Accessed 6 November 2018]. |
| [18] | TurboSquid, "TurboSquid," Louisiana Entertainment.gov, 2018. [Online]. Available: https://www.turbosquid.com/. [Accessed 2018]. |
| [19] | CGTrader, "CGTrader Free 3D Models," 2018. [Online]. Available: https://www.cgtrader.com/free-3d-models. [Accessed 2018]. |
| [20] | Blender, "The Free and Open Source 3D Creation Suite," 2018. [Online]. Available: https://www.blender.org/. [Accessed 2018]. |
| [21] | Maya, "Maya Computer Animation Software," Autodesk, Inc., 2018. [Online]. Available: https://www.autodesk.com/products/maya/overview. [Accessed 2018]. |
| [22] | GitHub, Inc., "GitHub," GitHub, Inc., San Francisco, 2018. |
| [23] | IDV, Inc., Unity, Inc., "SpeedTree," IDV, Inc., Lexington, SC, 2017. |
| [24] | U. Technologies, "FFTWindow Documentation," Unity, Inc., San Francisco, 2018. |
| [25] | Always & Forever Computer Entertainment, "Playthings VR," Always & Forever Computer Entertainment, Brooklyn, 2016. |
| [26] | Hyperbolic Magnetism, LLC; Split; Lokiman;, "Beat Saber," Czech Republic. |
| [27] | Sokay, LLC; Whiteman, Bryson; Cryptic Circuitry; Roman, Ramiro; Estrada, Jennifer;, "Raybeem VR," Sokay, LLC, Los Angeles, 2018. |
| [28] | Brace Yourself Games., "Crypt of the NecroDancer," Brace Yourself Games., 2018. [Online]. Available: https://braceyourselfgames.com/crypt-of-the-necrodancer/. [Accessed 2018]. |
| [29] | Wikipedia, "Guitar Hero (video game)," Harmonix, 2006. [Online]. Available: https://en.wikipedia.org/wiki/Guitar\_Hero\_(video\_game). [Accessed 2018]. |
| [30] | Wikipedia, "Rock Band (video game)," Harmonix, 2007. [Online]. Available: https://en.wikipedia.org/wiki/Rock\_Band\_(video\_game). [Accessed 2018]. |
| [31] | PlayStation, "Patapon Remastered," Sony Interactive Entertainment LLC, 2018. [Online]. Available: https://www.playstation.com/en-us/games/patapon-remastered-ps4/. [Accessed 2018]. |
| [32] | Epic Games, Inc., "Unreal Engine 4," Epic Games, Inc., Cary, 2018. |
| [33] | Ableton Live 10, "Ableton Live 10," Ableton Live, Berlin, 2018. |
| [34] | PreSonus, "Studio One," PreSonus Audio Electronics, Inc., Baton Rogue, 2018. |
| [35] | Apple, Inc., "GarageBand," Apple, Inc., Cupertino, 2010. |
| [36] | Unity Technologies, "Unity," Unity, San Francisco, 2011. |