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** 3](#_Toc529282603)

[1.1 Problem Statement 3](#_Toc529282604)

[1.2 Project Goal 3](#_Toc529282605)

[**2 Application / Product** 3](#_Toc529282606)

[2.1 Start 3](#_Toc529282607)

[**3 Background** 3](#_Toc529282608)

[3.1 Finish 3](#_Toc529282609)

[**4 Design** 3](#_Toc529282610)

[4.1 VR Structure in Unity 4](#_Toc529282611)

[4.2 Structure of Audio 4](#_Toc529282612)

[4.2.1 Background Ambient Music Compositions 4](#_Toc529282613)

[4.2.2 Sound Effects for Interactable Objects 4](#_Toc529282614)

[4.2.3 MIDI Arpeggiator 4](#_Toc529282615)

[4.3 Version Control 4](#_Toc529282616)

[**5 Implementation** 5](#_Toc529282617)

[5.1 Unity Development Kits 5](#_Toc529282618)

[5.2 Ableton Live 10 Music Editing 5](#_Toc529282619)

[5.3 Obstacles 5](#_Toc529282620)

[5.3.1 Platform Capabilities 5](#_Toc529282621)

[**6 Analysis and Verification** 5](#_Toc529282622)

[6.1 Playtesting 6](#_Toc529282623)

[6.1.1 Survey Results 6](#_Toc529282624)

[6.1.2 Reviews 6](#_Toc529282625)

[6.2 Industry Specialist Review 6](#_Toc529282626)

[**7 Interdisciplinary Connections** 6](#_Toc529282627)

[7.1 Start. 6](#_Toc529282628)

[**8 Related Work** 6](#_Toc529282629)

[8.1 VR: Music Visualizers / Interactive Applications 6](#_Toc529282630)

[8.1.1 Playthings VR 6](#_Toc529282631)

[8.1.2 Beat Saber 7](#_Toc529282632)

[8.1.3 Raybeem VR Music Visualizer 7](#_Toc529282633)

[8.2 Music Visualizers / Interactive Musical Applications 7](#_Toc529282634)

[8.3 Personal Previous Work 7](#_Toc529282635)

[8.3.1 Band Wagon 7](#_Toc529282636)

[**9 Future Work** 8](#_Toc529282637)

[9.1 Thesis Lead-In 8](#_Toc529282638)

[**10 Conclusion** 8](#_Toc529282639)

[**11 References** 8](#_Toc529282640)

# 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 further explore the capacity of audiovisual components of VR environments. The user will be able to interact with virtual instruments in the environment via controller input and microphone input, to generate correlated audiovisual output in the environment.

# Application / Product

## Usage

For those who are familiar with virtual reality environments, Soundscape is straightforward to play. 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.

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

## Finish

# Design

Before implementation in Unity [2]

## 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 [3]; these recommendations for structuring have served me well up through the close of development on this project.

In terms of porting this 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) [4] 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 [5]), 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 [6].

FMOD studio greatly aided in development, as this sound effects engine and authoring tool is primarily designed for video game scoring and sound design. [7]

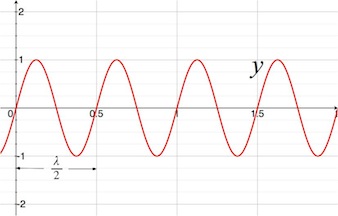
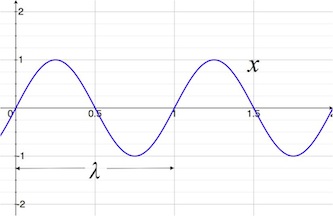
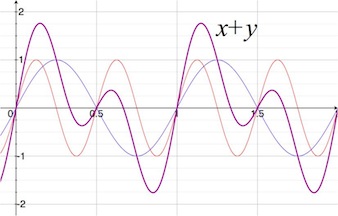
### Background Ambient Music Compositions

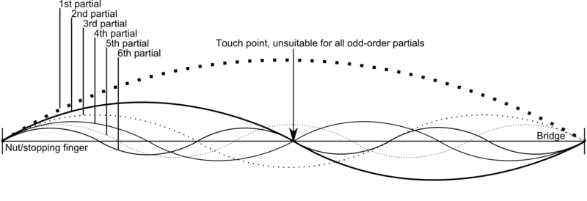
When I initially began production on Soundscape, I had compiled a few short compositions intended to be looped as background music for the player to use as scaffolding or as a guide when playing with the virtual musical objects. Though, after a few short rounds of playtesting, I determined that leaving 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 loose 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.

### 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. [9], [9] 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 the aforementioned overtones, see the following reference: [9].





### 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.[8] 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 [9] that allowed both freedom in creation for the user, yet substantial structure to yield sounds that would be pleasing to the Western ear. [10]

## Version Control

For ease of personal record keeping and the ability to revert back to prior editions in the case of fault, I used GitHub [11] to document my changes and have a reliable backup of version of my project.

# 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.

## Unity Development Kits

### 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.) [12] to create trees with advanced visual effects such as smooth LOD transition, fast billboarding, and natural wind animation.

## Audio Visualizer

### Fast Fourier Transform

Built into Unity’s SDK.

### Bandwidth Modifier

## Ableton Live 10 Music Editing

Bulleted list.

## 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 [4] 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

Bulleted list.

## Playtesting

### Survey Results

Link to Google Forms…

### Reviews

First-hand accounts.

## Industry Specialist Review

Thanks Dad.

# Interdisciplinary Connections

## Start.

Computer Science, Music, and Computer Graphics happily married. Polyamorous. Woo lol

# 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 form simple music visualizers to interactive music-based games, and 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

Link and speak of its awesomeness. [13]

### Beat Saber

Freakin awesome. So fun, cool, and relaxing. [14]

### Raybeem VR Music Visualizer

Rinse and repeat of above. [15]

## Music Visualizers / Interactive Musical Applications

## 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 [16]. 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 [17], Studio One [18], and Garage Band [19], 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. The previous work done on this tool was used in a game jam – a

# Conclusion

# References

|  |  |
| --- | --- |
| [1] | Facebook Technologies, LLC., Unity, Inc., "Oculus Support," Facebook Technologies, LLC., 2018. [Online]. Available: https://support.oculus.com/170128916778795/. |
| [2] | Unity Technologies, "Unity," Unity, San Francisco, 2011. |
| [3] | Unity Technologies, "Large Project Organisation," Unity Technologies, 2018. [Online]. Available: https://unity3d.com/learn/tutorials/topics/tips/large-project-organisation. [Accessed 6 November 2018]. |
| [4] | Oculus VR, LLC, "Oculus Utilities for Unity," Facebook Technologies, LLC, Menlo Park, 2017. |
| [5] | Oculus Rift, LLC, "Recommended System Specifications," Facebook Technologies, LLC, Menlo Park, 2017. |
| [6] | 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]. |
| [7] | Firelight Technologies, "FMOD," Firelight Technologies, Melbourne, Victoria, 2017. |
| [8] | WikiAuthor, "WikiAudio - Arpeggiator," 5 April 2018. [Online]. Available: https://www.wikiaudio.org/arpeggiator/. |
| [9] | J. Albaugh, "Musical Chord Progression Arpeggiator," CodePen.io, 2018. [Online]. Available: https://codepen.io/jakealbaugh/full/qNrZyw/. [Accessed 6 November 2018]. |
| [10] | 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]. |
| [11] | GitHub, Inc., "GitHub," GitHub, Inc., San Francisco, 2018. |
| [12] | IDV, Inc., Unity, Inc., "SpeedTree," IDV, Inc., Lexington, SC, 2017. |
| [13] | Always & Forever Computer Entertainment, "Playthings VR," Always & Forever Computer Entertainment, Brooklyn, 2016. |
| [14] | Hyperbolic Magnetism, LLC; Split; Lokiman;, "Beat Saber," Czech Republic. |
| [15] | Sokay, LLC; Whiteman, Bryson; Cryptic Circuitry; Roman, Ramiro; Estrada, Jennifer;, "Raybeem VR," Sokay, LLC, Los Angeles, 2018. |
| [16] | Epic Games, Inc., "Unreal Engine 4," Epic Games, Inc., Cary, 2018. |
| [17] | Ableton Live 10, "Ableton Live 10," Ableton Live, Berlin, 2018. |
| [18] | PreSonus, "Studio One," PreSonus Audio Electronics, Inc., Baton Rogue, 2018. |
| [19] | Apple, Inc., "GarageBand," Apple, Inc., Cupertino, 2010. |