5-2021

# Wwisdom in Action

Drew R. Tomasik
*Western Michigan University*, drew.tomasik@gmail.com

# WWISDOM IN ACTION

by

Drew R. Tomasik

A thesis submitted to the Graduate College
in partial fulfillment of requirements
for the degree of Master of Music
School of Music
Western Michigan University
May 2021

Thesis Committee:

Christopher Biggs, D.M.A., Chair
Carter J. Rice, D.A.
Keith Kothman, Ph.D.

# WWISDOM IN ACTION

Drew R. Tomasik, M.M.

Western Michigan University, 2021

Writing music to score for video games requires an extensive artistic vocabulary, but equally important is an understanding of game development and the technological integration required. This text is an overview and interpretation of a semester's work in Wwise, a software environment designed specifically for the integration of sound into video games. As Wwise is standard for the industry, facility with it increases marketability to a job candidate wanting a career in game development. The text stands alone to recount the process of learning Wwise from square one, as well as creating a soundtrack for a sample game; that said, it accompanies an unlisted YouTube video accessible by the link, youtu.be/Ozk7vLYjA4k.

ACKNOWLEDGEMENTS

I would like to express my thanks to Drs. Christopher Biggs, Carter Rice, and Keith Kothman for their assistance in my development as composer and student throughout my Master's education. I also greatly appreciate their willingness to join my thesis committee, as I strive to obtain my degree. I would also like to thank Jared Tubbs and Kris Bendrick for their friendship, support and musical insight throughout these past two years.

Drew R. Tomasik

## A FOREWORD

The following accompanies an unlisted YouTube video accessible by the link:

youtu.be/Ozk7vLYjA4k

This self-produced video is a practical demonstration of my work, serving as the chief component of this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

# CHAPTER I

## CONTEXT FOR MY CHOICE IN THESIS

Since my early teens, I took a keen interest in writing music for games. Franchises like *Kingdom Hearts*, *Final Fantasy*, and *Sonic the Hedgehog* impressed upon me the importance of a soundtrack in the efficacy of a story. It amazed me that even a piece of music playing within the limitations of 8-bit technology could carry immense emotional weight. In the scheme of a modern development cycle, however, this capability becomes an obligation. No technological advance of the 21st century has erased the notion of working within constraints, particularly concerning any field of game development. Hardware specifications, digital memory usage and processor demand all have a hand in the refinement of art in games, and music is no exception. This degree helped me dispel misconceptions that a game composer's job stops at sending an email with audio files attached. I realize now that a truly skilled game composer has a concept of more than just the music as an isolated entity. The music must properly interact with the game engine, art design, sound design and player input to present a seamless experience.

While the visual and mechanical aspects of game design take place in software development environments like Unity, Unreal and Amazon Lumberyard, sound and music integration happens in a type of software called audio middleware. This is similar to a development environment, but equipped with exhaustive tools for playing, editing and organizing audio files. Audio middleware includes programs like PortAudio, Fabric, FMOD and Wwise. I spent the past semester with Wwise, building an understanding of the software from a starting point of zero experience. It is specifically oriented toward games, and it integrates easily with Unity and Unreal, two of the most popular game engines in the industry. As such, I thought

it would be a smart career move to get it under my fingers as quickly as possible. It seems a fitting end to my degree, to be able to take the composition and production skills I gained over two years and apply them practically.

## CHAPTER II

## AN OVERVIEW OF THE PROJECT

The goal for my degree's end was to present a demo game outfitted with my music, which the game engine could utilize dynamically. I would produce newly composed tracks in tandem with my endeavors in Wwise. That said, I had to be realistic; learning software from square one and producing a myriad of tracks for an expansive game would result in either shaky or outright dysfunctional work. I resolved that the game need not be particularly long, nor the tracks I composed for it. I would work a small, polished project with a solid deliverable at the end. My main concern was gaining enough of a handle on Wwise in the meantime. I knew that there was certificate curriculum available, and that a demo game provided by that curriculum would serve as a decent canvas for this project. I spent the forthcoming weeks buried in tutorials and instructional videos, and while I did not purchase the prohibitively expensive $200 exam for the certificate, I made it through all the relevant lessons and quizzes without difficulty.

# CHAPTER III

## WWISE FOR THE LAYMAN

Wwise is a system of interlocking, dense hierarchies[1] directing a game engine to any number of various audio files needed for playback at any one time. These audio files can be played at random, in sequence, in combination, under certain conditions, and any other permutation needed in the game. Containers within containers house audio files, and multiple containers can easily (and non-destructively) access the same file. A Wwise Event relays any instruction from the game engine to do anything and everything to a referenced audio file or container. This can affect playback volume, reverb depth, pitch shifting duration, and pretty much anything else a developer can think of. The only stipulation is that the name of this Event must be the same as the message coming from the game engine. Wwise can interpret game data as parameters and map them to other parameters in Wwise. For instance, how healthy a player character is can determine how loud their heartbeat sounds in the mix. Wwise has numerous methods of taking game data and turning it into sound, but it also provides tools for mixing, schematics, and performance monitoring[2] in the game. When the configuration of hierarchies satisfies the developer, everything is packaged as a Soundbank and shelved with the rest of the game's source files.

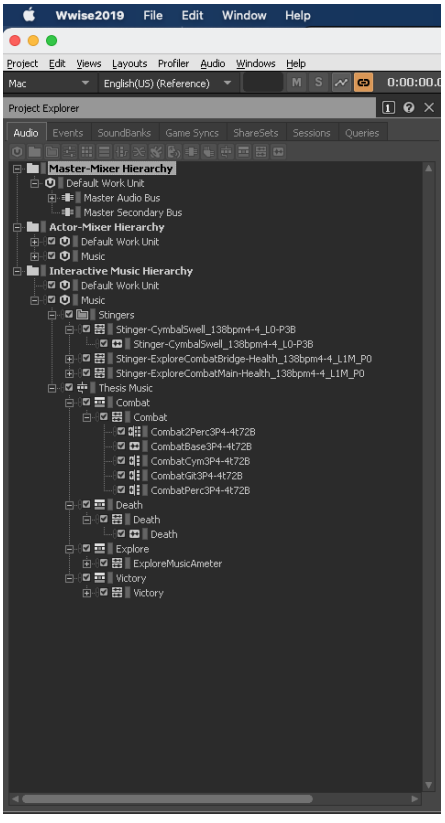

*Figure 1. Hierarchies in Wwise.*

[1] See Figure 1.
[2] See Figure 2.

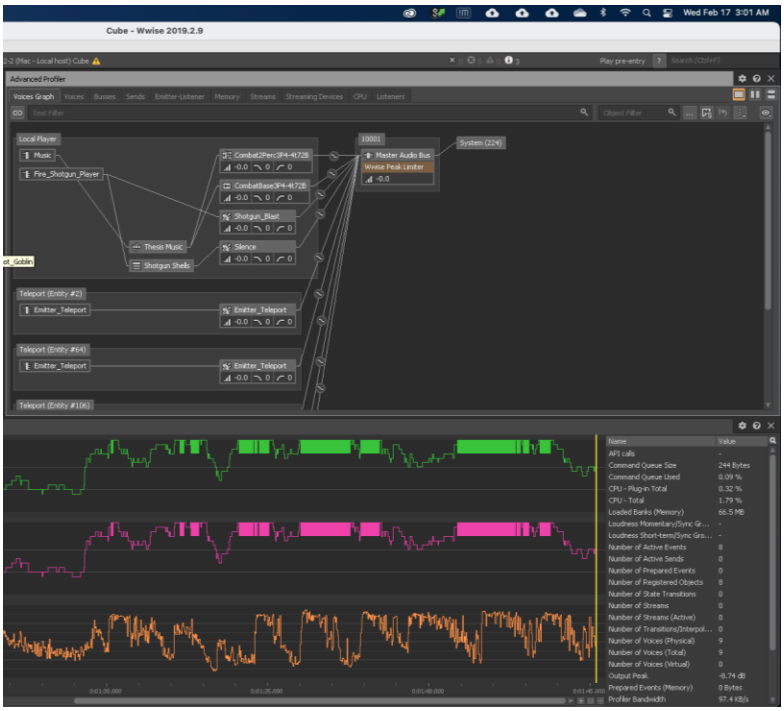

*Figure 2. Schematic View in Wwise.*

# CHAPTER IV

## MUSICAL CHOICES MADE

My first step after completing the curriculum provided by Audiokinetic (the publisher of Wwise) was to erase all the structures in the Interactive Music Hierarchy of the demo project. I had played through the demo game *Cube* with and without the demo soundtrack; it is a simple first-person shooter, and I play a soldier fending off monsters in an industrial compound. Using the curriculum, the demo Wwise project, and my own experience, I assessed which situations I could outfit with music and which parameters/events needed addressing to do so. The finer details of my integration procedure are demonstrated fully in the accompanying YouTube video, but my musical choices are worth expanding upon here.

I often draw inspiration from games whose soundscapes and character I am familiar with, as was the case with my take on the *Cube* soundtrack. The aesthetic of the rusting tread plates and arid atmosphere – particularly the sandstone walls and floor in certain parts of the map – reminded me of the industrial landscapes in first-person shooter *Team Fortress.* Meanwhile, the dim corridors with emergency lighting and electrical panels imitated the eerie steampunk mannerisms in *Bioshock.* Further, the grunts of the monsters and my character, as well as the blast of the shotgun, brought to mind the gritty dirges playing in the original *DOOM* for MS-DOS (1993). From these memories I synthesized what I hope is an appropriate accompaniment to the visuals and gameplay: clanking, brooding piano hits with lots of tinkling metal artifacts in the higher register; cymbal strokes destroyed with effects processing; and driving kick drum and bass guitar for when situations become more intense. I was immensely grateful for the ability to add layers of audio conditionally to a currently playing track.[3] This allowed me to give both regular and "Boss" combat encounters the same flavor but add suitable intensity at moments of high difficulty for the player.

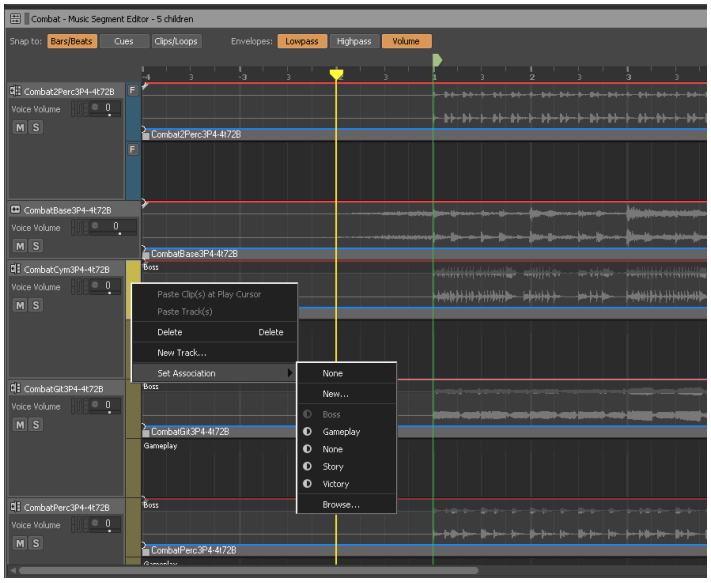

*Figure 3. Tracks and layers within the Interactive Music Hierarchy.*

[3] See Figure 3.

# CHAPTER V

## TAKEAWAYS OUTSIDE THE PRODUCT

Having a deliverable for the purpose of this thesis is wonderful, but my largest gains were experiential in nature. With as little as a semester's worth of tutorials in this curriculum, I feel wildly more prepared and abreast of the knowledge it takes to be a musician in the game industry. The concepts I came to understand as part of this project are applicable to software outside of Wwise, but also outside of this particular expression of the medium. The conditional programming available to me in Wwise gave me new scope on how music can be more immediately relevant to circumstance, and as a result I plan to become engaged in more interactive and multimedia work as part of my own musical growth. Audiokinetic's insistence on awareness of memory/CPU limitations forced me to use minimal material efficiently and effectively. Perhaps most importantly, this was the first time I was able to freely compose a soundtrack to interactive picture and have complete control over the process of integration.

This is only the first of many steps I will take to develop facility in Wwise and in game development. Should I earn myself a space in this industry during the coming years, I am excited by the prospect of using Wwise and other audio middleware to explore the potential of interactive music. The relationship between listener and composer scarcely seems so close, and I relish in the opportunity to foster dialogue between a player's listening experience and my audio workstation. Engagement in such an interdisciplinary, interactive craft promotes the advancement of the medium, and I am grateful that I got to experiment with it here.