GAME AUDIO DESIGN, COMPOSITION, AND INTEGRATION FOR MARINE ENVIRONMENTS

Tyler Chung
Northeastern University
Boston, MA
chung.ty@northeastern.edu

ABSTRACT

Marine soundscapes have been a continued interest for not just marine biologists and oceanographers, but also sound engineers. These soundscapes such as these are essential to creative audio works, particularly in video games such as *Subnautica* (2014) and *Abzû* (2016). This project aims to investigate the sound design, music composition, and game audio integration present in video games with an underwater setting, and how to apply those observations in a currently in-progress game, *Heqet*. This project entails an analysis of marine game audio works, as well as the use of a variety of audio tools, such as a DAW, music programming, an audio editor, and game audio middleware (Wwise) in a game engine (Unity). The audio created with these tools is intended to accompany multiple aspects of the game, including its narrative, setting, characters, gameplay, and game states.

1. INTRODUCTION

Game software has proved useful in not just video game development, but also cultural heritage, marine education, and underwater environment simulations (see Figure 1). Game engines are software frameworks that provide a variety of pre-built features to support the development of video games so that developers do not have to work from scratch. The increasing popularity of video games has created a high demand for experience and research in specific game engines, one of which is Unity. Unity allows game developers to use C# scripts, control animation and audio, render graphics, and other features. Unity also allows developers to integrate middleware for enhanced capabilities not present in Unity's tools. Middleware acts as the "middle" software between the engine and external systems. Wwise is a middleware solution for integrating sound between the game engine (Unity) and audio elements, which is necessary for creating immersive marine environments of video games. For reasons such as this, game development become an increasingly more interdisciplinary industry, as all aspects of a game now each play a larger role in its quality and success. Developing game audio, in many cases, touches on object-oriented programming, digital signal processing, and user interface design, in addition to music and sound design components.

Although many games with open worlds or a focus on nature have underwater elements, underwater video games where the marine environment is the core focus have become its own sub-genre. Music composition and sound design for underwater video games can be challenging due to a variety of factors. The audio must incorporate the difference in acoustics of the underwater setting, the interactive elements of the player with marine life, and the creative design and narrative of the game itself. As a result, music in video games has become its own subdiscipline and a studied field by game scholars [2]. Investigating the methods of creating and integrating current underwater video games is essential.

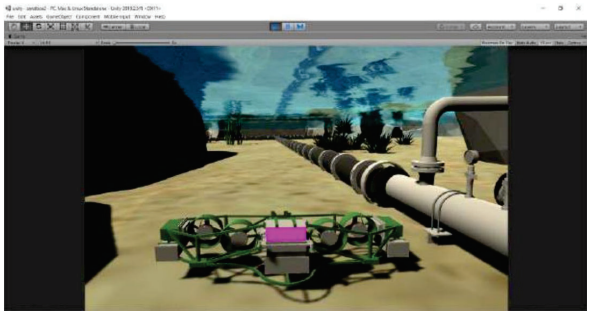

Figure 1. Unity is used to test the navigation of autonomous vehicles in an underwater simulator environment [1].

Heqet is an underwater adventure game currently in development that utilizes Unity and Wwise to explore, showcase, and put into practice techniques within music technology and other areas used in other underwater video games. Such techniques include crossfading, auxiliary sends, and spatialization. The audio of *Heqet* is inspired by the video games *Subnautica* and *Abzû*.

2. UNDERWATER VIDEO GAMES

Subnautica is an indie action-adventure game developed and published by Unknown Worlds Entertainment in 2014. Simon Chylinski is the sound designer and composer of the game. The original soundtrack is comprised of 56 songs. *Abzû* is an adventure game developed by Giant Squid and published by 505 Games. The soundtrack is made up of 18 songs composed by Austin Wintory. These two games are the face of the underwater game genre, and the aesthetics and materials of the soundtracks are worth investigating to understand how they are so effective and impactful to audiences.

Especially in video games, the term "song" is used loosely, mostly due to the impact of digital audio. Songs are music compositions with vocals, and in copyright the songwriting and sound recording processes are distinct. However, today generally the combination of the composition and the recording is referred to as a song, regardless of if a vocal part is present, as the piece can still have a great impact on audiences. Instrumental or electronic compositions, especially those of soundtracks, are considered "equal" to songs; music streaming services today make no distinction between these compositions as they are one in the same.[1]

Figure 2. A screenshot of *Subnautica* by the author.

Figure 3. A screenshot of *Abzû* by the author.

The music of *Subnautica* is somewhat different from *Abzû*, despite the similar environments. For one, the compositions are much shorter, with each piece being no shorter than 50 seconds but no longer than 3 minutes, compared to *Abzû*, whose pieces can expand from 1 to 10 full minutes. Also, while *Subnautica* seems to be mostly composed of electronic instruments, *Abzû*'s music is entirely orchestral [3]. Finally, the two pieces have different moods capturing different sides of the ocean, with *Subnautica*'s music feeling mysterious, adventurous, and eerie, while that of *Abzû* feels steady, mythological, and content. This is all due to the distinct narratives of the two games, as *Subnautica* requires the player to survive dangerous creatures and uncover the history of aliens, whereas *Abzû*'s gameplay focuses solely on the aquatic environments the player traverses, with no dialogue or demands.

However, both games' narratives entail ancient civilizations, which the ocean can be closely associated with. They therefore have similar elements. What is obvious is the reverberation and integration of the audio, but what is also worth noting is their occasional presence of a choir, which gives the sense of ancient origins and exploration. In addition, while aliens are not present in *Abzû*'s narrative, the term "alien" can be used to describe the ocean and the sea creatures within them. As discussed by K. Galloway in her discussion of *Abzû*, the alien nature of the ocean is not "by default hostile," but rather takes the player into an environment that is nonhuman [4]. While *Subnautica*'s composition has a more prominent alien-like feel due to its narrative and survival genre, *Abzû*'s composition focuses on an alien-like environment that is slow-paced and inconspicuous. This demonstrates an advantage nonhuman acoustic communication provides for marine environments in providing immersion for listeners. Technology allows for creative simulations of natural environments, especially those that are not intended to be hospitable, accessible, or heard by human beings.

Music plays a role in a video game's "cinematicability." Particularly in *Abzû*, the music allows the player to decipher and truly understand certain events in the game, which is based on fallen civilizations that used to exist (Babylonian and Sumerian) [5]. It allows the player to feel free in a game where the narrative may be linear. It also fills in narrative or visual gaps, by either providing a futuristic quality with synthesizers, or a familiar quality with established orchestral timbres. Filling in these gaps is the crucial function of music in film. Such games have almost created a genre of "interactive movies." Cinematic games that are limited in mechanics may struggle due to less gameplay time, however this can be compensated for by making music a mechanic of the game as well, and not just an accompaniment.

The goal of the audio of *Heqet* is to produce an environment that is as almost as effective if not just as effective as *Subnautica* and *Abzû* utilizing similar techniques, qualities, and thought processes. It is an attempt to integrate audio that fits into the narrative, gameplay, and style of the game as much as possible. The effectiveness of the audio is also based on how much it adds to the audience's feeling and experience, i.e. its ability to fill in the gaps and make the game feel complete despite limitations.

3. OVERVIEW OF *HEQET*

Heqet is a single player adventure puzzle game, originally created by Leo Norton, and developed by Devin Kansakar, Eli Silverstein, and myself. Development with the team started in early 2024. Both *Subnautica* and *Abzû* serve as inspirations for *Heqet* in both its game development and musical techniques.

[1] Sometimes these game "songs" are called "tracks," though "track" is used just as loosely to mean either an entire composition or a MIDI or audio part in a DAW.

3.1 Story

The game tells the story of a glowing frog named Anura, who is spawned in by "The Hands," a powerful entity of nature, at the beginning of the game. The Hands tells the Anura (the player) to destroy "The Orb," for reasons unknown to the player at the start. The player must solve puzzles to get to different areas and clear off objects on top of coral platforms. Doing so expands the influence of The Hands, as it can then emerge from these platforms (acting as checkpoints or fast travel for the player).

However, the truth is that The Hands are an evil entity bent on gaining power by expanding the area in which they can pop out of these coral platforms. They bleach coral nearby them as the player slowly leads The Hands towards the orb through the game. When the hands emerge from a coral platform, the coral bleaches. The Orb is a powerful golden object that gives life and color to the coral around it. Coral is a symbol for the health of the ocean. The Hands want to utilize and take the power from the coral and The Orb. It is sometimes referenced by the creatures the player gets to meet, as well as cave paintings. The evil and good aspects of The Hands and The Orb respectively are hinted at slightly as the game progresses.

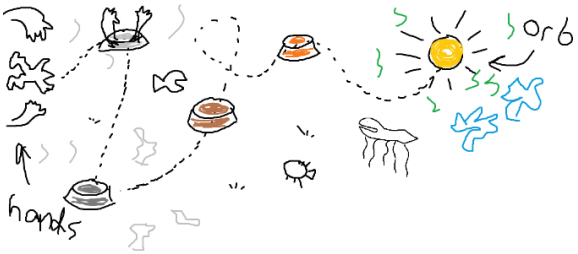

Figure 4. Drawing of game progression during stages of game development.

The Hands's true evil form is only revealed at the very end. However, Anura is a sea creature that does not belong to either side. They can either destroy The Orb, which the player had been pursuing the entire time, or save The Orb by getting The Hands eaten by a behemoth, a giant mindless but powerful fish.

Heqet is based on many real-life elements and events. Heqet, also spelled Heket, is the goddess of fertility in Egyptian mythology, which Anura can personify based on the player's end-game decision. Anura is the scientific classification for an order of amphibians, which we refer to as frogs. Coral bleaching is a current real-world issue that is exacerbated by human activity, corresponding to the anthropomorphic nature of The Hands. The Orb is loosely based on an actual golden orb found by scientists in September 2023 in the Gulf of Alaska.[2]

The game is an interactive movie in many ways. The world is meant to be explored by the player, but there is an explicit progression that the player must follow. The atmosphere and the story are what make the game feel alive. The player also traverses different environments. The music must fit each area and the sounds must be adjusted, e.g. reverb in a cave versus an open area. The audio needs to fulfill both the cinematicability and the interactive player-agency aspects of the game. In other words, the player must both feel like a story is unfolding before them and that they are creating the story themselves.

3.2 Development

Heqet is developed in Unity, utilizing Blender for its 3D graphics, Ableton Live 11 Standard for music composition, and Wwise for audio integration. The game has a variety of features and mechanics.

Unity provides essential features for the game development process. The game is built with multiple game objects in a scene. Many different components can be attached to these objects, such as animators, 3D renderers, collision detectors ("colliders"), a camera, visual effects, C# scripts, etc.

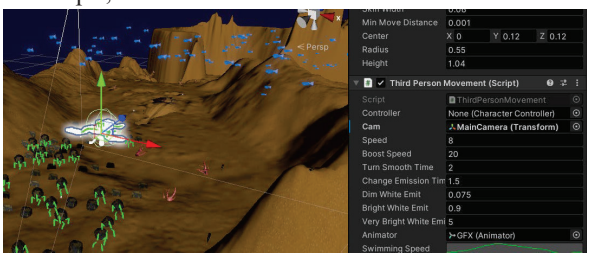

Figure 5. Anura, the frog, in a scene in the Unity editor. Attached to it is a C# script for third-person movement with multiple fields.

Blender is a 3D computer graphics software that has multiple features such as modeling, animation, rigging, and rendering.

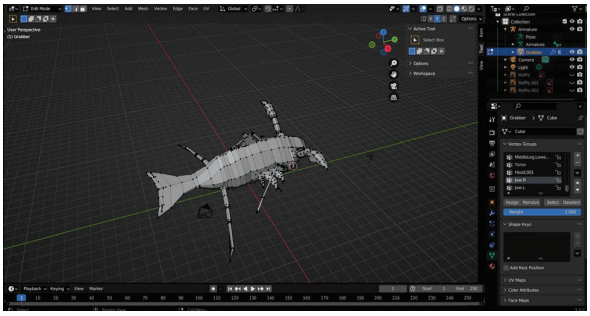

Figure 6. The Grabber sea creature in Blender's layout workspace with a variety of editors, such as the animation timeline at the very bottom.

Ableton Live is the digital audio workstation (DAW) that was used for music composition and some sound edits.

[2] https://www.smithsonianmag.com/smart-news/what-is-this-mysterious-golden-orb-scientists-found-on-the-ocean-floor-180982872/

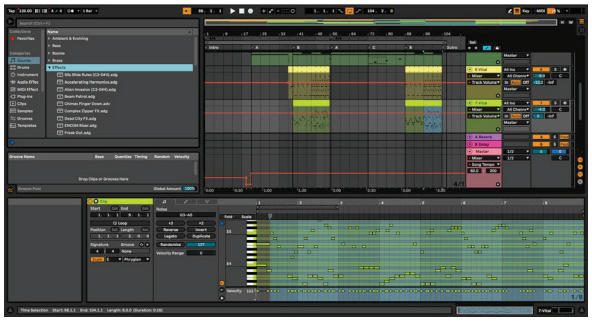

Figure 7. The Ableton Live Set of "Doorways" in automation mode.

Wwise has features that allow for more control over the audio of the game using events, switches, states, and soundbanks. *Abzû* uses Wwise for its audio integration [3].

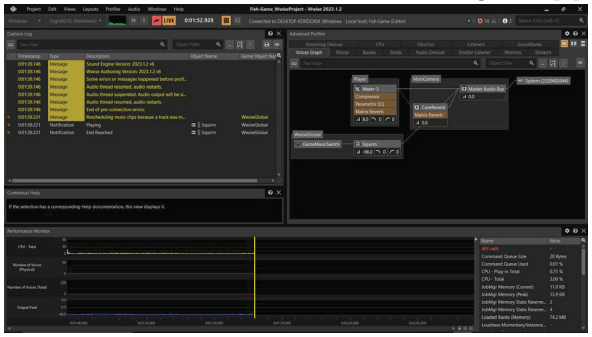

Figure 8. A view of Wwise's live profiler layout showing the voices graph while playing the game.

4. GAMEPLAY FEATURES

The game was built with the focus on a few core gameplay elements, with more planned in the future.

4.1 Puzzles

Each area of the game has various puzzles that the player must complete to continue onto the next one. The player traverses a tutorial cave, the shallows, the sandy dunes, and the tripod valley. The puzzles are (in order of appearance):

- Clearing boulders in front of a door to insert an object that matches the shape of the keyhole of the door.
- Help the hermit crabs move to another location.
- Feed a hungry flounder to obtain a key item underneath it.
- Navigate through seaweed and find a key item.

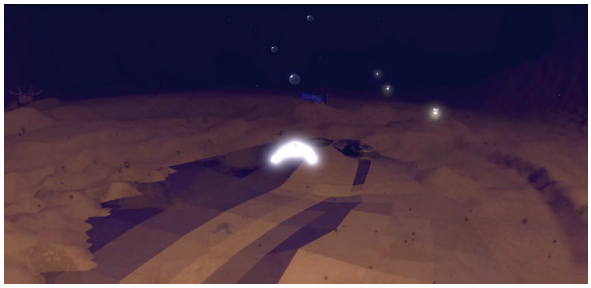

Figure 9. A fish approaches a flounder laying on the ground.

Figure 10. The flounder comes out of the ground to eat the fish.

4.2 Player Control

The core of the gameplay entails the player moving throughout the 3D world. The player's movement is standard WASD controls for forward, left turn, backward, and right turn, respectively, and the player can control the camera with the mouse to allow for movement in any direction with respect to it. The player can gain a small speed boost by holding the left shift key while moving, but it has a cooldown. The brightness of Anura is dim while the speed boost is on cooldown and becomes brighter when it is ready.

However, the more fun aspect of Anura's movement is their ability to dive down and jump from the ground, like a real-life frog, by holding down and releasing the spacebar. The player's other ability is telekinesis. When triggered with the left mouse button, the player can move certain objects anywhere to complete the puzzles. The player can also talk with certain characters within the game to reveal these puzzles.

Figure 11. Anura uses their telekinesis on a rock, which has a purple visual effect. Also present are walking hermit crabs.

4.3 Boundaries

The game handles boundaries slightly differently. Because the game takes place underwater, and the intent was for the game to feel free, it is hard to enforce puzzle boundaries without the environment feeling restricted. This also proves difficult since the player can swim upwards, unlike a traditional above water game where boundaries can be established through physical objects, such as mountains or fences. Many games simply use invisible walls (colliders). However, the game takes a different approach to enforcing boundaries.

After completing the tutorial area, the player can swim anywhere. However, if the player tries to cross into the next puzzle area (without completing the puzzles within the current one), they will be eaten by a behemoth. This is done through trigger colliders. If the player collides with this "wall," the eat animation is triggered. The screen fades to black and they respawn at the tutorial exit. This feels more in-line with the environment than a simple invisible wall that the player cannot physically go through.

Boundaries are also necessary to ensure that the player is somewhat guided. The player should not be able to swim straight in any direction too far outside of the area that the story takes place. 3D player areas are designated in the game engine, and should the player leave any one of those areas, the behemoth will swim next to and pass the player as a warning. If within a certain timeframe the player has not reentered a play area, the behemoth comes and eats the player.

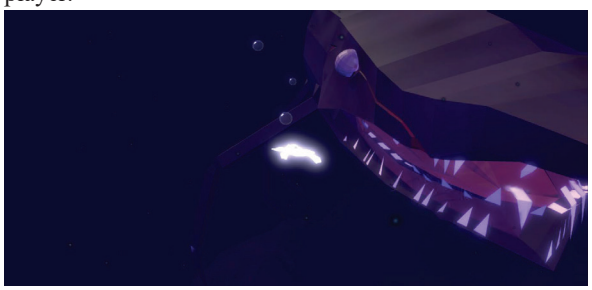

Figure 12. The behemoth travels to and eats the player.

4.4 Environmental Elements

The game is created with a similar idea to *Abzû*, where the player is quite literally a lantern in the ocean and the player's vision is restricted from seeing beyond a certain distance. The terrain material is relatively simple, and within the environment are rocks, plants, and a variety of moving sea creatures. In addition, there are specific acoustics in marine environments [6] that are considered:

- The speed of sound in water is almost five times faster than in air over long distances.
- Lower frequency sounds in deep waters experience less interference and therefore can travel long distances and are less attenuated.
- Ambient noise has distinct physical and biological sound sources.
- Anthropogenic sounds negatively affect sea species.[3]
- Continuous sounds dominate lower frequency sounds and travels long distances. They are "persuasive" and therefore increase background noise.
- Sea creatures rely on sound for communication and navigation.
- Sound is the best form of transmitting information, as light/sight is attenuated.

Some of these aspects are mimicked in the game.

5. MUSIC COMPOSITION

The game has 3 musical compositions, all produced in Ableton Live 11 Standard. Vital[4] was used for all the synthesizers used in the pieces.

5.1 "Doorways"

"Doorways," the first song, is named after the idea of underwater caves being doorways that the listener is exploring. This exploration, while mysterious, is also adventurous and alluring. The composition has the form of: Intro A B A′ C B′. The composition is in the key of E Phrygian. The tempo is at 100 bpm during the intro, though slows down during its transition to the A section, from which the tempo maintains 120 bpm throughout the rest of the piece. The entire piece heavily relies on synthesizers, taking inspiration from the songs of *Subnautica*. All instruments are sent into a reverb aux track.

The piece is meant to feel bright at the start, as if the listener has stumbled into a new but calming world. The synths create the eerie alien nature of the ocean. The bell is meant to feel like a voice welcoming the listener to this new place. The chords throughout the piece are meant to convey the harmonious aspect of marine life where there are no disturbances. The percussive instruments, though regular in rhythm, are meant to mimic the sounds one would hear in a cave. The plucky synth leads the piece in the B sections like it is a lead vocal (and a duet). All the parts of the piece act as a band of the ocean, with layers new layers being introduced throughout and all returning in the end and polyrhythm present in the hook.

5.2 "Squirm"

The second piece, "Squirm," is a more mysterious piece that is meant to convey the slight feelings of bizarreness and discomfort one would feel if they were sent to a new but fascinating world. This piece is in the key of C major and roughly two minutes in changes to C minor. The piece is not multi-sectional, but it is looped. It only contains eight parts, though some are modulated in certain ways.

[3] The Hands's sound design is inspired by these anthropogenic sounds, which have been a large focus in modern oceanography studies.

[4] https://vital.audio/

The foggy instrument that begins the piece is meant to make the listener feel like they are in a tunnel of some kind, corresponding to the cave the player first spawns at. "Foggy" refers to a noisy, muffled timbre that is prominent in the stereo image. The bass provides warmth, and the wood percussion provides uneasiness. The call and response is meant to symbolize the relationship between Anura and The Hands, or alternatively, mother nature. The timbres of the two instruments with delay are meant to contrast each other, with the synth naturally sounding consonant and the guitar sounding dissonant. One can imagine the plucky instrument being some little fish in the distance swimming around curiously.

5.3 "Free"

The third piece, "Free," is solely orchestral. The composition only has six parts and is in the form of: A B A′. Each section is in the key of G major, G minor, and F major, respectively. It maintains a steady tempo of 54 bpm. The orchestration consists of clarinet-like keys that play chords, a violin/horn, violins, a synth that mimics a full orchestra, flutes, and percussion.

"Free," naturally, tries to convey a feeling of freedom felt by the player after being released into this new world. It is heavily inspired by the soundtrack of *Abzû*. It is very much meant to feel like a dream, which can be both exciting and scary. This is symbolized by the key changes and the uncertainty of the piece's cadence. The inconspicuous polymeter of each instrument (which causes each instrument to change pitches at different times) that contributes to the chords also contributes to this feeling. The minimal instruments allow for each part to matter more and truly sing. Dreams take people into new worlds that can feel alien to what is known in the real world, and that is the intent of this piece.

6. SOUND DESIGN

Sounds are either created using Ableton Live and Vital or are recorded samples.

6.1 Water

Water sounds are essential to providing noise that is heard in the ocean. They are swimming sounds used for the player and other sea creatures that serve as background noise to be constantly heard. The sounds must synchronize with their movement. For example, the player moves by kicking and has speed boosts. The "slap" moment in the sound must occur at the same time Anura kicks, and the sound must be adjusted due to the speed. In particular, the swimming sounds heard from the player are edited recordings of someone swimming breaststroke, matching with how Anura kicks. All the water sounds, including bubbles and movement, are either sampled or recorded.

6.2 Sea Creatures

The sea creatures showcased in the game all have unique sounds. The behemoth has three sounds, a pass sound, a chase sound, and a chomp sound. The pass and chase sounds are made from a "flangered" harp sound, and the chomp was recorded and edited. The bobbit worm has a strong noisy sound when it soars from its current location. That sound is made from white noise (to mimic sand), with effects such as delay and granular synthesis. It is meant to be intense due to the worm's size and sudden movement. The flounder has a simple chomp sound for when it eats a fish. Finally, the hermit crabs have sounds for when they reveal themselves and their footsteps on the sand. The reveal sound is also made from white noise and the footsteps are made from percussion.

6.3 Other Effects

To further the realism and immersion of the game, there are other effects:

- Collisions, particularly between rocks and, for example, the ground. These collisions sounds are edited recordings of real-life rocks colliding with each other in the water.
- Telekinesis, made from a Vital synth with flanger-like waves, a phaser, and delay.
- The door opening and breaking apart, which is made from an actual edited recording of a door opening and closing. It is split into a head (for inserting an object key) and a tail (for the door opening).
- The Hands, which is a recording of the inside of the Boston Aquarium (with a lot of noise from human activity) with resonators and a compressor, with the resonators' base note and the track's panning each being modulated by a low-frequency oscillator (LFO).

All the sounds in the game have been edited in some way contribute to the game's alien-like nature. They do not perfectly feel like sounds one would hear in the ocean, and this is intentional.

7. AUDIO INTEGRATION

Both the music and sounds are integrated using Wwise. Using audio middleware is necessary due to the acoustics involved in marine environments and the project's complexity.

7.1 Sound Events

Sound events can be triggered in multiple ways, usually through a script and sometimes with an animation event. Sound events can be started as soon as the game starts, which is especially the case for continuously moving sea creatures which the player should be able to always hear (until they are far enough away). Wwise's AkEvent script can be used to do this easily. For timed effects such as the door opening, the audio is triggered by an animation event.

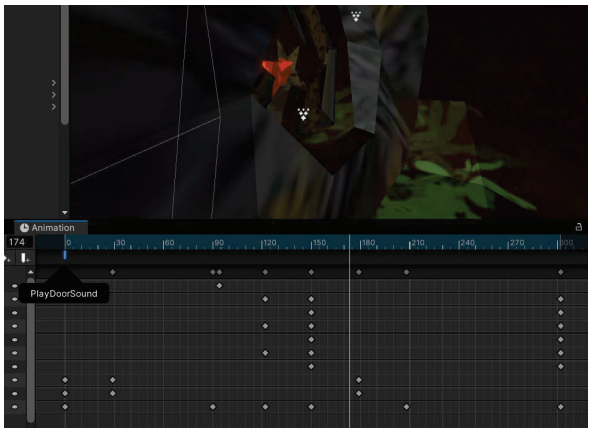

Figure 13. The PlayDoorSound() function of the C# script attached to the door is called at the start of the animation (but can be called at any point during the animation).

In addition, for the head and the tail of the door opening sound to be synchronized, there are two techniques that can be used. Another event can be called within the animation and both events can each play the head and the tail. What was used instead was a sequence container, which contains the head, silence, and the tail (in that order). The silence duration can be controlled to sync with the key insertion and the door breaking.

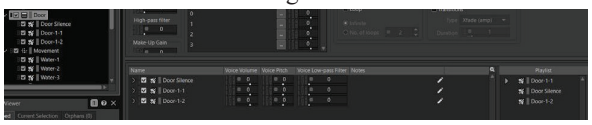

Figure 14. The sequence container, whose SFX are added to a playlist to be played in order with a single "play door" event.

For the Telekinesis effect, the sound is triggered with a C# script that already handles the player's telekinesis ability. For sounds of the fish particle system to occur, however, it is not sensible to add sounds to each individual fish, as this could cause performance issues. Instead, a few invisible game objects continuously travel in the direction of the fish (and restart from the beginning after a certain distance) which each act as a sound source. This allows the listener to feel like the fish are moving.

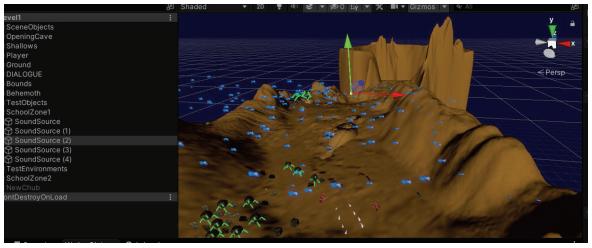

Figure 15. One of five sound source game objects moving in the direction of the fish.

For collisions, it is very easy for them to sound staticky, i.e. they all sound the same and therefore unrealistic. To conquer this issue, a random container is used with 7 different collision sounds. Each time a collision event occurs, a random one of the collision sounds are triggered. This contributes to the game's immersion.

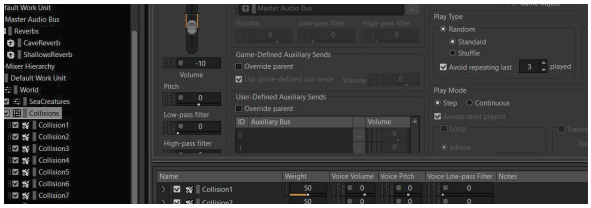

Figure 16. The Collisions random container.

7.2 Sound States

To integrate the music within the game, Wwise has an interactive music hierarchy that allows for a music switch. This switch allows for music transitions depending on the music state, which can be controlled by the game (Unity). These transitions can occur between playlists of tracks.

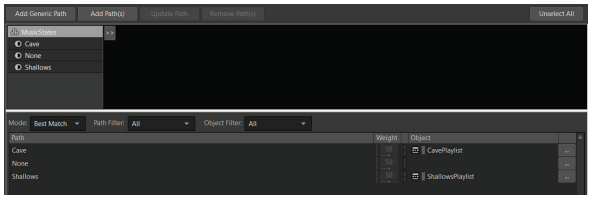

Figure 17. The music switch, connecting the MusicStates group to the playlists of tracks.

These states can be set using a mesh trigger collider for the cave to detect when the player enters and leaves the cave. A C# script can then be used to set the music states accordingly.

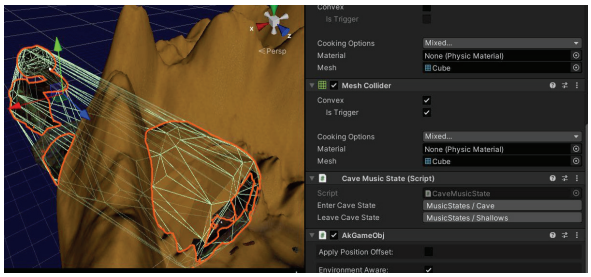

Figure 18. The cave with a regular mesh collider (so there is physical collision, i.e. objects do not go through the cave), a convex trigger collider to detect when the player enters and leaves it, and a C# script to set the music states.

A separate switch container outside of the music hierarchy is used for player movement (as the player is in one movement state at a time. Those states are set based on the animation being acted on Anura.

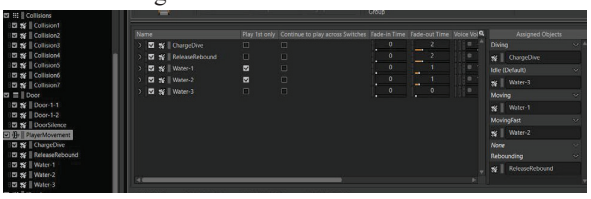

Figure 19. The PlayerMovement switch container, with each sound assigned to different player game states.

```
[SerializeField] private AK.Wwise.State EnterCaveState;
[SerializeField] private AK.Wwise.State LeaveCaveState;

Event function  tylerchung
void Awake()
{
    EnterCaveState.SetValue();
}

Event function  tylerchung
private void OnTriggerEnter(Collider other)
{
    if (other.CompareTag("Player"))
    {
        Debug.Log( message: "Entered Cave");
        EnterCaveState.SetValue();
    }
}

Event function  tylerchung
private void OnTriggerExit(Collider other)
{
    if (other.CompareTag("Player"))
    {
        Debug.Log( message: "Left Cave");
        LeaveCaveState.SetValue();
    }
}
```

Figure 20. A simple C# script to set the cave states: the Cave playlist for when the player enters the cave and the Shallows playlist for when they leave it.

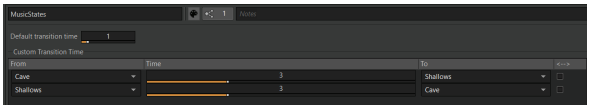

Figure 21. The MusicStates switch allows for control of state transitions (crossfading).

7.3 Soundscape

There is a clear soundscape change as the player progresses. The most notable thing is the change from being inside a cave to being in a wide-open area. In particular, the reverb must change as the player goes through this transition. The reverbs for each area can be done in Unity via reverb zones and auxiliary buses (which certain sounds can all be sent through) within the Master Audio Bus.

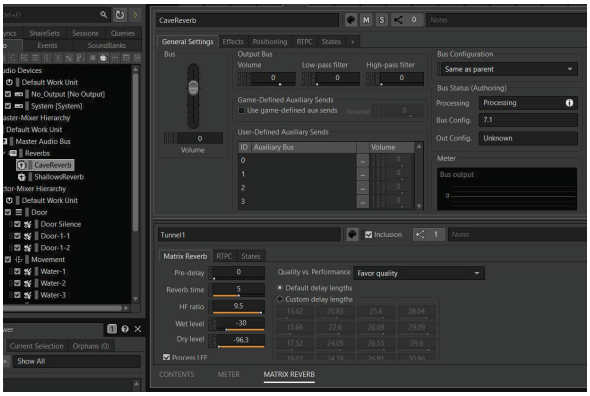

Figure 22. The CaveReverb aux track where sound effects are sent through (for sounds with “Use game-defined aux sends” checked).

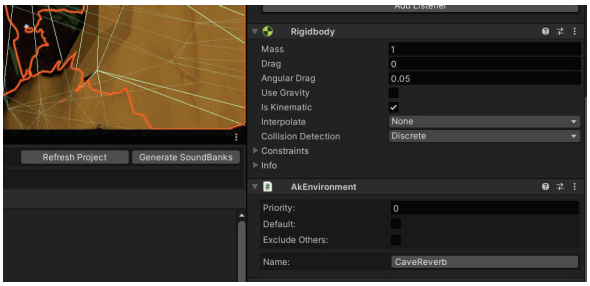

Figure 23. The AkEnvironment component of the cave attached with the CaveReverb. While in the trigger collider of the cave, audio is sent through that aux bus.

There exists an intersection between the AkEnvironment objects for the Cave and the Shallows area. A game object with the AkEnvironmentPortal component that is located in between the Cave and Shallows trigger colliders to allow for the transition and change in reverb.

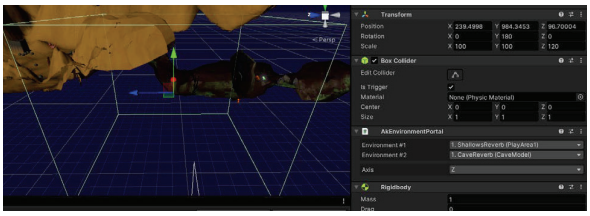

Figure 24. The ReverbZonePortal object with an AkEnvironmentPortal component. The z axis is selected because the reverb transition is in the z direction, and the component automatically detects the two aux buses associated with the AkEnvironments.

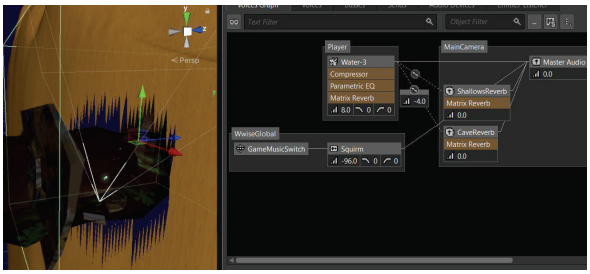

Figure 25. The Wwise Profiler demonstrates the audio being affected by the object with the AkEnvironmentPortal component. As the player travels through the cave (in the

z-direction), there is less CaveReverb and more Shallows-Reverb for a smooth transition.

In addition to reverb zones, to mimic the acoustics of the water as best as possible, attenuation can be created for all world sound effects. This marine attenuation is meant for sounds to travel large distances and high frequencies to be attenuated as sounds get farther. However, it does not perfectly mimic real-life marine acoustics, as the environment must respect how the game is designed (especially in terms of progression.

Figure 26. The marine attenuation applied to all sounds in the World actor-mixer. The volume and low-pass filter values change with respect to the distance between the listener and the sound source.

8. CONCLUSIONS

The goal was to research and analyze the game development of underwater games, particularly the audio aspects, and then put it into practice with an underwater video game. The bulk of my contributions to the game was the audio, with some programming and designing as well. There were a variety of obstacles that are customary in game development, such as the story progression and brainstorming puzzles, and in learning new software, such as the lack of documentation and issues that correspond to public questions asked by sound designers years ago that are left unanswered. In other words, was a lot of trouble-shooting, and it occurs often when using niche APIs in all fields of computer science.

There are a lot of limitations to note. For one, with the scale of the game development process and deadlines, a full soundtrack could not be made, nor a fully polished game. Only 3 core music pieces were created, and the game is developed enough to be a demo. Time was a limitation for everyone working on this project, with responsibilities present for other classes. This affects the quality and complexity of the sounds and the integration, as a professional setting would likely entail traveling across the world to obtain very specific high-quality sounds for a video game, as well as large amounts of material for Wwise that require fine-tuned mechanics.

There is a lot of opportunity for expansion. The game will continue to be developed to accomplish what we have planned during the development process and publicized via a distribution service. These skills are transferable to various fields, within and outside of game development. While much of development is specific to Unity and Wwise, the design process, software development principles, and audio techniques can be used in many fields of the industry. The project has opened the doors for further growth both personally and for those seeking to follow a similar path.

Acknowledgments

I thank my mentors Prof. Anthony De Ritis and Prof. Ronald Bruce Smith for their guidance and feedback on the audio aspects of the game. I also want to especially thank my fellow game developers, Leo, Devin, and Eli for working on *Heqet* with me. Finally, I thank Prof. Michael G. Wagner of Drexel University for his public educational videos on game audio with Unity and Wwise.

9. REFERENCES

[1] E. Osa and P. Orukpe, "Simulation of an Underwater Environment via Unity 3D Software," *1st International Conference of the Nigerian Institution of Professional Engineers and Scientists Book of Proceedings*, pp. 384–396, Oct. 2021. Accessed: Mar. 21, 2024. [Online]. Available: https://researchgate.net/publication/357406008

[2] A. Vékási, "Myth and Eco-messaging in ABZÛ," *Encounters of the Popular Kind: Traditions and Mythologies in Dialogue*, pp. 79–89, 2021. Accessed: Mar. 20, 2024. [Online]. Available: https://www.eltereader.hu/media/2022/02/Encounters_of_the_Popular_Kind__Traditions_and_Mythologies_in_Dialogue-web2.pdf#page=80

[3] M. Bradley and A. Wintory, "Crafting the ocean in the music of ABZÛ (feat. Austin Wintory)," YouTube, https://youtu.be/0S8WGtbYvSg (accessed Mar. 24, 2024).

[4] K. Galloway, "Climate Games, the Blue Humanities, and listening to the deep-sea ecosystems in games in a time of ecological crisis," *AMP: American Music Perspectives*, vol. 2, no. 2, pp. 139–157, Dec. 2021. doi:10.5325/ampamermusipers.2.2.0139

[5] J. Freitas, "Music, interaction and cinematicability: between Bound and Abzû," Videojogos 2018 — 10th Conference on Videogame Sciences and Arts, pp. 135–152, 2018. Accessed: 2018. [Online]. Available: https://core.ac.uk/reader/288869465

[6] N. Rako-Gospić and M. Picciulin, "Underwater noise: Sources and effects on Marine Life," *World Seas: An Environmental Evaluation*, vol. 3, pp. 367–389, Sep. 2018. doi:10.1016/b978-0-12-805052-1.00023-1