Sound, Space and Interaction: Treasure Hunt

by Nikki Rademaker (s2641887) and Yanna Smid (s2634163)

The Treasure Hunt

This project welcomes you to the invisible world of a treasure hunt. You need to find the hidden treasure not by the use of your eyes, but by the use of your ears. Listen closely to the frequency of the sound based on the location of the tracker while carrying it. The faster the beeps, the closer you are to the treasure. The beeping sound that can be heard, is based on the same sound a metal detector makes. This way the users already associate the rhythm of the beeping with how close they are to the treasure. The beeping sound is a synthesized noise made by an oscillator object in the PlugData patch, with an initial frequency of 380. To increase the timeframe within each beep, a delay is set. The closer to the coordinates of the treasure, the shorter the delay.

There is a little twist to the hunt. Zombies are watching over the treasure. Don't get too close to them, or otherwise they will capture you. You will hear the zombies better the closer you get to them, so make sure to stay far away from them. The zombie growls come from a .wav file from the Minecraft game. Certain game lovers might already be familiar with the sound. Others, who have not heard this sound before, will understand that this sound comes from monsters because of the ominous snarls. To make the sound louder, the signal of the .wav file gets multiplied by the fraction one divided by the distance between the tracker and the zombie.

While searching for the treasure, extra pressure is put on the player by the ticking of a timer. The goal is to find the treasure within a certain time limit between 30 and 60 seconds. The ticking noise gets louder as you near the time limit. This sound comes from a .wav file. The volume starts softly multiplied by 0.5 and increases until it is multiplied by 13 in the duration of the randomly selected time limit.

If the time is up you are game over, and a game over tune will be played. If the treasure has been found before the time limit, a winning sound is played. Lastly, if the player has found the zombies, an evil laughter sound is played, indicating that the monsters got you. All of these sounds come from a .wav file.

The game reacts to the user by adjusting the sounds based on their location and the user reacts to the game based on these sounds. To ensure some unpredictability in the treasure hunt game, the location of the treasure and the location of the zombies are randomly assigned. However, it is designed in a way that the two locations don't overlap and aren't too close to each other. Furthermore, the time limit that is set for the user to find the treasure is also randomly chosen each time you start the game.

The Setup

In regards to creating a treasure hunt atmosphere, we decided to allow users to interact with their surroundings in a confined space. This allowed them to freely navigate the area and utilize a tracking device to search for the treasure, enhancing the immersive experience of

the game. We used the HTC Vive VR set for this purpose, which includes a tracker that users hold in their hands to detect their position and orientation. For this project, the tracker was used to send the x and y position of the user to the computer system, so the code of the game could compare the live position to the generated random positions for the treasure and the monsters. This set also included two base stations that were needed to assign a room and capture the position of the tracking device in this room.

In each of the four corners of this room, a speaker was placed. The program was connected to all four channels to enhance the experience of this game. Sounds from the tracker, timer and monsters were all played simultaneously through all channels to increase the pressure of the game. We found that it was crucial for all speakers to play the same sounds, as you hear the volume of a sound increase as you approach a speaker. Additionally, pieces of furniture were positioned around the room as obstacles to increase the difficulty of the game.

The Creation Process & Reflections

The initial idea was the creation of an interactive game that combines physical movement and auditory feedback to immerse players. The objective was for players to locate a secret spot in a room within a set time limit, guided by sound cues. Initially, we considered using OpenCV for real-time video analysis but ultimately chose the HTC Vive with a tracker for better precision of position detection. An additional gameplay element was also added involving avoiding the location filled with monsters to further increase the challenge of the game.

During the development process we encountered several challenges. It started with difficulties in installing the necessary software due to a lack of internet access on the computer we were using. After overcoming this, we successfully set up a virtual environment via SteamVR and the Unity project. Once completed, the creation of the code to send information from Unity to PlugData went smoothly. Additionally, when setting up all the needed hardware, we faced a challenge in expanding the audio output to utilize all four speakers. In PlugData, we faced a significant challenge when using a float box to store and compare float values. The issue was that the float box would output the previously received float upon receiving a new input. This was resolved by using a "trigger bang float" object which first bangs the current float and then employs a spigot to allow the next float to be banged.

In the beginning, Yanna was working on our PlugData patches for the creation of our, while Nikki collaborated with Dominik and Emma in setting up the base stations and the Unity environment. Nikki also worked on linking Unity with Pure Data to transmit the tracker's coordinates. We both helped each other where needed. Once the setup was complete, we both worked on finishing the main PlugData patch and selected the necessary sound samples.

Reflecting back on this project, working on the game was an enjoyable experience. Initially, we were hesitant about using the HTC Vive set due to previous challenges with the technology, but it proved to be very reliable this time around. We also had concerns about how to connect Unity to PlugData, but using Open Sound Control for Unity 3D made the integration very doable. Another concern we had was that the game would be boring and

easy to play. However, the game turned out to be quite challenging for users at times, and we got a lot of positive responses from people who played it.

Sources/Credits:

- SteamVR Unity plugin for connecting the HTC Vive setup to Unity.
- UnityOSC (OSC.cs) for allowing to send messages from Unity to PlugData
- Winning sound (win.wav) https://pixabay.com/sound-effects/playful-casino-slot-machine-jackpot-3-183921/
- Game over, time is up (violin-lose.wav) https://pixabay.com/sound-effects/violin-lose-5-185126/
- Timer sound (timer.wav) https://pixabay.com/sound-effects/a-kitchen-timer-55420/
- Monster sound (MCZombie.wav) Minecraft.
- Game over, monsters (doomed.wav) https://pixabay.com/sound-effects/doomed-effect-37231/