Forest Mole – An Audio-Based VR Game

Exploring spatial audio-simulation for navigation and spatial awareness in games

Master's Thesis in Computer Science and Engineering

Ida Altenstedt
Rebecka Willig

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2026

Master's Thesis 2026

Forest Mole – An Audio-Based VR Game

Exploring spatial audio-simulation for navigation
and spatial awareness in games

Ida Altenstedt
Rebecka Willig

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2026

Forest Mole – An Audio-Based VR Game
Exploring spatial audio-simulation for navigation
and spatial awareness in games
Ida Altenstedt
Rebecka Willig

Supervisor: Staffan Björk, Department of Computer Science and Engineering
Examiner: Michael Heron, Department of Computer Science and Engineering

Master's Thesis 2026
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Description of the picture on the cover page (if applicable)

Typeset in LaTeX
Gothenburg, Sweden 2026

Forest Mole – An Audio-Based VR Game
Exploring spatial audio-simulation for navigation
and spatial awareness in games
Ida Altenstedt
Rebecka Willig
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

This thesis follows the design and implementation of an audio-based VR game. The game was created as part of a process using the *Research Through Design* method, in order to answer the research question: What are important aspects to consider when designing audio games for VR? The answer is precented in the form of a SWOT analysis, containing 25 observations spanning four categories. The report contains the first two development cycles for the audio-based VR game *Forest Mole*. During development, various issues arose, the most important one being related to the functionality of the spatial audio library used. While this negatively affected the expectation of the given result, it offered insight into broader aspects of the current state of audio development in games. This in turn affected the observations found for the SWOT analysis.

Keywords: computer science, project, thesis, SWOT analysis, VR, 3D audio, games, audio games, sound design, spatial audio.

Acknowledgements

We would like to give a massive thank you to Staffan Björk who supervised this project. His ideas, feedback and general input were invaluable in the making of both the game and this thesis.

We would also like to thank the friends and family members who voice acted in the project: Olof Altenstedt, Daniella Johansson, Samuel Kyletoft and Hanna Viberg. Their contributions made the game into a more cohesive experience.

Furthermore, we would like to thank our playtesters who provided us with valuable feedback about our game and insights into audio-based VR games in general.

Finally, we would like to thank our opponents Deniz Özden and Christos Ringas, and our proof-reader Samuel Kyletoft, who all helped make our thesis shine.

Ida Altenstedt, Rebecka Willig, Gothenburg, 2026-06-18

Contents

List of Figures

1 Introduction

Most games made nowadays are *video games*. This means that they mainly focus on playing the game through visuals, as opposed to *audio games* which are games that primarily focus on sound when it comes to giving information and feedback from the game. This does not mean that sound is not important in video games. The virtual world is seldom quiet, it usually contains everything from environmental sounds for ambience to sound effects that indicate actions or events. The difference is the quality required of the audio functionality in audio games. Because the sounds themselves are a lot more noticeable if they are missing than if the sound bounces across walls and through doorways correctly. The absence of the latter is only noticed when it is looked for.

There are different ways to use sounds to give the player input about what is happening in-game and in the environment. Outside of the basics of adding sound cues to actions, sounds can also be used to give the player a sense of the dimensions of the current environment, be it echoing sewers, an opera hall, or a tiny closet.

Games often use a plethora of shortcuts to make their worlds and their physics feel real, something which is needed to lessen the computations needed for calculating different aspects of the game, making it able to update in real time. This also means that much of the details that go unnoticed in the real world simply go unimplemented as to not use any extra computation time. Using sounds as an example, this can work well in a video game where the sounds only supplement the visual experience, but in an audio game where they are the main form of feedback, the lack of detail becomes a problem.

This is doubly important for virtual reality (VR) games. In VR the player is immersed in an environment in a way that is unlike sitting before a screen, and many games made for VR reflect this. VR games feel like they are made to enhance the visuals by placing the player inside the virtual world. But by doing this they also place the player inside the game's auditory space. It got us wondering, if VR is focused on enhancing the virtual experience, what happens when you use it as the medium for an audio game?

To truly see what aspects are important to focus on when designing a primarily audio focused game for VR, this project made use of the *Research Through Design* method. This meant that the project included making a game, the development of which would bring valuable insights. To aid in the understanding of what is needed for an audio game, this included looking for information related to not just audio games but also audio in general.

The game created for this project is a VR game that lacks visuals completely. This not only did reduced the development time required to create something feasible for the allotted time, it also forced the developers to rely on audio, and haptic, feedback during the development. At the same time, removing visual input eliminated sight as a potential confounding variable during testing. Outside of this, the lack of visuals also made the VR headset act as a blindfold and contributed to visual sensory deprivation for the player. Sensory deprivation, in the form of eyesight, immerses the player more thoroughly in the game world because it demands the player's focus in a way that is not required when there are visuals available to truth check the auditory input[1].

By using the Research Through Design method, the game enabled the authors to explore how audio is treated differently in audio games as compared to video games, with a focus on spatial audio and how it changes according to player and object position in an environment. Another motivation behind developing the game was to place the authors into the role of audio game creators, thereby reducing the risk of important aspects of spatial audio being overlooked.

The hope was for this approach to help give a better understanding, and a concrete example, of what aspects are needed and useful to take heed of when designing audio games for VR.

Research Question:

What are important aspects to consider when designing audio games for VR?

Important aspects is a term that is reasonably broad. The plan is to focus on the aspects related to gameplay and player experience for audio games. When designing a game you design everything from the menu to the levels, the problems to the sound effects used, the feedback from actions to the story of the game. These are the parts that are referred to when talking about *designing* a game throughout the paper. Because the focus is on making a game for VR all of this will need to be designed to fit that medium, as well.

The thesis does not research anything related to board games. *Audio games* here refer to games that mainly give their player feedback through audio. And while there certainly exist board games that do this as well, that aspect of games will not be covered in this report. The plan is to focus on digital games leaning towards VR or other such experiences where the physics of the game are made to imitate reality.

Furthermore, while we were aware that audio games, particularly their lack of visuals, can have an impact on the player's cognitive load, this was not something that we focused on during the project.

What was learnt during the game's development, combined with what was learnt during the prestudy is showcased in the format of a *SWOT analysis*, which highlights the parts that were most important during the research.

2

Background

There are many different aspects to consider when creating a game. Everything from which platforms to use to which engine. The feel of the game and the abilities of the final product can depend heavily on what is possible with the chosen hardware and software.

For example, basic spatial audio is expected in games nowadays[2]. But the quality of said spatial audio differs between games. While how sounds can be simulated within games will be written about more in *Chapter 3 Theory*, the important aspect here is that many of the developments for spatial audio is made using the PC platform. Ultimately locking the soundscape onto a 2D plane, instead of giving it a more complex 3D spatial audio that could be more likened to how humans seek out sound sources in real life. Something which can be more freely modelled in VR which allows the player to tilt their head and therefore the screen.

Other technologies that we used in the project but are generally outside the scope of the research question can be found in *Chapter 5 Process* as they are used.

2.1 Development Tools

When designing and developing a software application there are several tools one can use that will help the process. Some of these are used specifically to support teams when collaborating or to keep track of, categorise and prioritise tasks.

2.1.1 Git

Git[3] is a tool used for version control. This is especially useful when working in teams and/or on multiple devices. A Git repository is a folder that contains the project and its version history, allowing users to track changes, restore previous versions and collaborate more easily.

GitHub[4] is a platform where people can share and collaborate on Git repositories. From GitHub, users can download the latest version of a project and later upload their additions or changes so they can be accessed by other teammates.

2.1.2 Trello

Trello[5] is a simple online workspace that allows for the creation of cards which can then be moved between card-columns. See Figure 2.1 for an example of a *Trello* board from their website.

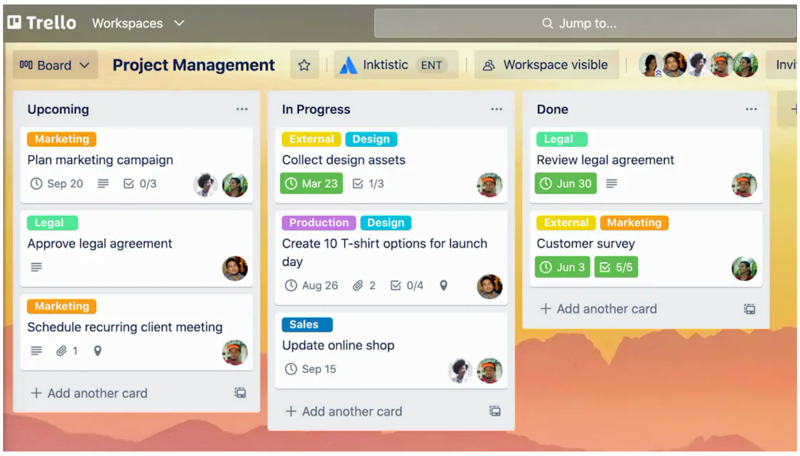


Figure 2.1: Trello board[1]

2.2 Virtual Reality

Virtual Reality (**VR**) is part of what is known as **Extended Reality** (**XR**). XR is an umbrella term that refers to immersive technologies that merge the physical and digital worlds, such as Virtual Reality and **Augmented Reality** (**AR**). Augmented Reality uses the real world as a background or overlays virtual objects onto it, creating the impression that these objects are part of the physical world. In contrast, Virtual Reality fully replaces the real world and immerses the user in a completely virtual world.[6]

The abbreviation **VR** stands for **Virtual Reality**. The term “virtual” have been in use for a few hundred years with the meaning of “being something in essence or effect, though not actually or in fact”[7]. This in combination with technology, means that it has been used in a sense of “not physically existing but made to appear by software”[7].

Virtual Reality then refers to a non-physical world made by software to appear like an immersive world around the user. This is accomplished through the use of **VR headsets**[8].

A VR headset is a head-mounted screen whose visuals change according to the user’s movements. The screen is split into two parts, with slightly differing visuals, corresponding to human eyes and allowing for a virtual imitation of depth perception.

VR has two modes. One is **stationary** which focuses on tracking head rotations and portraying the game state accordingly. Stationary VR games are meant to be

[1]Image from Trello, `https://trello.com/guide/trello-101`, accessed 2026-05-08.

played either standing still or sitting down.[9]

The other VR mode is **room-scale**, which uses VR head tracking and motion tracking on a larger scale to map the users real-world movements into the virtual environment. This enables natural interactions such as walking within a defined space or bending down to look at and/or pick up objects.[9] The physical area the player can move around is often referred to as play area or play space. Figure 2.2 shows a depiction of a user configuring their VR setup, the blue rectangle on the floor indicating the play space.

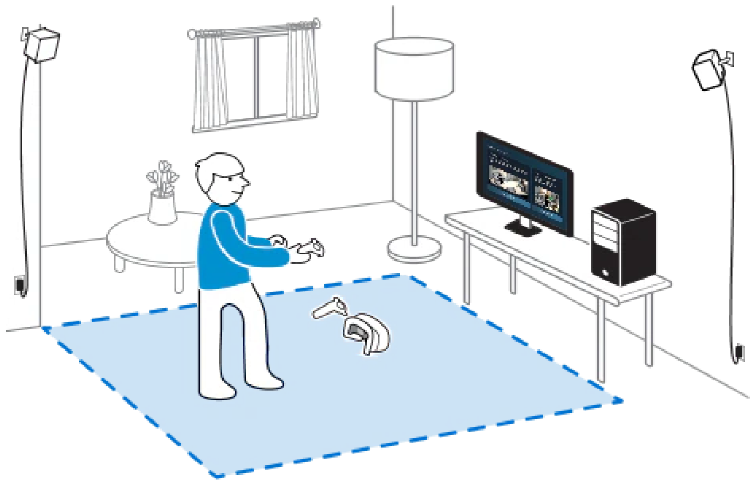


Figure 2.2: Room-scale play space[2]

While having the same core functionalities, all VR headsets are not the same, and often differ in how they choose to implement said functionalities. This ranges from whether the headset can be used wirelessly, to the shapes and positions of the controllers' buttons, and even to what software the user is required to have on their computer.

Two examples of VR headsets are *HTC VIVE*[10] and *Meta Quest 2*[11]. The *HTC VIVE* headset (see Figure 2.3a) works by using two base stations (see Figure 2.3b) that emit light that the headset can see to position itself in the room. The requirement of having the base stations visible makes the *HTC VIVE* inconvenient to move to a new location, when compared to VR headsets that use a different approach, as the base stations are best positioned mounted on a wall. The *HTC VIVE*'s two hand-held controllers (see figure 2.3c) are identical, which can cause confusion when the user tries to figure out which controller to hold in which hand, especially if the game or application they are using does not visually differentiate the controllers (e.g. by having a right and left hand model for the respective controller). On top of each controller is a clickable touchpad, which reacts both to touch, clicking and position of said touch or click. This allows the touchpad to be used for functions such as navigation, be it physical navigation of a playable character or navigating

[2]Image from Vive, `https://www.vive.com/us/support/vive-pro-hmd/category_howto/setting-up-room-scale-play-area.html`, accessed 2026-05-23.

through a menu. The touch and click can also be mapped to different functions, for example the touch can move a cursor to the desired option and then clicking will select it, without having to use a different button for select. A downside to the touchpad however is that users may accidentally touch it when resting their thumb on the controller, potentially leading to unwanted interactions.

The *Meta Quest 2*, originally released as the *Oculus Quest 2*, is a VR headset developed by *Oculus*, a division of *Meta*. It is a standalone VR headset, meaning it can be used without being connected to a PC, as its built-in processor allows applications to run directly on the device. The headset can also be connected to a PC using either cable or Wi-Fi, in which case it functions more like a traditional PC VR headset. The *Quest 2* has cameras that track both the user's head position and the positions of the controllers, eliminating the need for external sensors or base stations. Additionally, it contains an internal battery, allowing it to operate wirelessly, which can provide greater freedom of movement during use. Its controllers are physically different between left and right. This makes it easier to differentiate between them, but it also makes it so they cannot be interchanged like the *HTC VIVE* controllers. The *Quest 2* controllers also use joysticks instead of trackpads. The *Quest 2* headset and controllers can be seen it Figure 2.4.

(a) Headset

(b) Base stations

(c) Controller

Figure 2.3: HTC VIVE[3]

Figure 2.4: Meta Quest 2[4]

[3]Images from Inet, `https://www.inet.se/produkt/1999813/htc-vive`, accessed 2026-05-11.

[4]Image from Amazon, `https://www.amazon.com/Oculus-Quest-Advanced-All-One-Virtual/dp/B099VMT8VZ?th=1`, accessed 2026-05-11.

2.3 Game Engines

Three of the most commonly used game engines within the industry are *Unreal Engine*[12], *Unity*[13] and *Godot*[14].[15]

Unreal Engine is one of the most powerful and widely used game engines as of the start of 2026[16]. It is most well known for its cutting-edge graphics and high-fidelity visuals. Unreal Engine has a steep learning curve and requires a powerful computer to run smoothly.[16] As can be attested by one of the authors, the steep learning curve means that much time that could be used for further development instead went to understanding how Unreal works and how to use VR in its workspace.

Unity is a user-friendly game engine. It is older than Unreal Engine and not as powerful graphic wise. This engine is most known for its flexibility and ease of use. Said flexibility makes it great for a wide variety of games, everything from 2D and 3D to AR and VR.[16] The authors' prior experience with Unity does show that it is easy to use to create 3D games, with a learning curve that is easy enough that a prototype can be finished in accordance with the time set.

Godot is an open-source, community-driven game engine. It has a comprehensive list of tools for 2D and 3D game development and is a popular choice among indie developers. It is the most lightweight engine of the options mentioned.[16] The authors have had positive experiences with this engine multiple times in the past.

2.3.1 Godot Add-ons

Add-ons in Godot, also known as plugins, are downloadable libraries that add functionality to a project, such as tools or assets. Two such libraries are *Godot XR Tools*[17] and *Raytraced Audio*[18].

Godot XR Tools adds functionality for XR, including VR. This add-on is an extensive library containing tools and assets that help the user monitor for example head tracking and controller input, and map them to actions in the game.

The *Raytraced Audio* add-on uses ray-tracing to extend Godot's audio, adding effects like echo, outdoor ambience and muffle.

There are of course many more add-ons for Godot, one example being *Godot Steam Audio*[19], which attempts to implement *Steam Audio*[20] for Godot. A bit more information about *Godot Steam Audio* can be found in *Chapter 7 Discussion* since the existence of this add-on was not known about until far into development.

2.4 Games

In this report, games refer specifically to digital games played on a computer, VR headset, console or other type of electronic device with screens and/or speakers. While the concept of a "game" could be referring to other activities such as board games or table top role playing games, these will not be included or discussed in this report.

This section contains some games that are relevant when investigating the research question, some of which are **audio games**. Audio games are a specific subset or genre of games that rely primarily, or entirely, on sound rather than visuals to communicate information to the player. These games use voice lines, sound effects and other audio cues to convey the game state, provide feedback, and guide player actions.

Audio games were primarily made for the visually impaired at the start, as an alternative to video games. As of today, summer of 2026, the main audience is still the visually impaired users, but it is slowly gaining more notice. One reason for this is because of the issue of accessibility, which has become more mainstream in recent years. This in turn has influenced artists and students to create experimental freeware PC audio games so they can explore the possibilities and limitations of this form of gaming, which in turn contributes to a steadily growing commercial interest.[21]

2.4.1 Perception

There are games such as *Perception*[22] where the player has to make sounds in order to see the environment around them, similarly to echolocation[5]. While this allows for a story where the playable character is traversing the world blindly, this does not mean that the player is blind. The game translates the sounds the character hears into something the player can see, prompting the player to still rely on visuals in order to navigate

2.4.2 Scanner Sombre

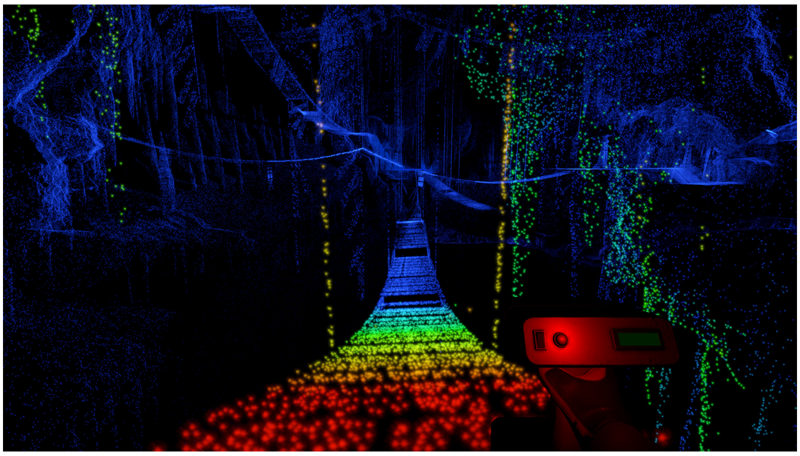


Figure 2.5: Scanner Sombre[6]

Scanner Sombre[24] is a first-person horror exploration game in which the player navigates a dark cave system. Its central mechanic is a scanning device that reveals

[5]Echolocation: A process of using echoing sound waves to map a usually dark environment.[23]

[6]Image from Steam's store, `https://store.steampowered.com/app/475190/Scanner_Sombre/`, accessed 2026-06-10.

the otherwise invisible environment. The scanner emits laser pulses and records their reflections, visualising the surroundings as a cloud of coloured points, see Figure 2.5. The colour of each point corresponds to its distance from the player, allowing the player to gradually construct a spatial understanding of the cave while exploring.

2.4.3 A Blind Legend

A Blind Legend[25] is an audio game in which the player assumes the role of a famous blind knight on a journey to rescue his wife, guided by his sighted daughter. The game is filled with audio-based cutscenes and various diegetic (existing within the narrative) voice lines, which help to enrich the game world and enhance the player immersion. Additionally, the game features non-diegetic (added for the audience) audio cues. These are either explained in the beginning of the game or introduced when necessary during gameplay. The explanations are delivered through a text-to-speech voice, clearly distinguishing informational messages intended for the player (e.g. "loading") from voices that are part of the narrative.

2.4.4 Audio Game Prototype

In a study by Falk in 2020[26], an audio game prototype was created and evaluated in order to explore how to design engaging audio games. The evaluation consisted of observing participants interacting with the prototype, after which they provided feedback. The results showed, for example, that the participants became more proficient as they progressed, and that they thought the lack of visuals made the experience more exciting and tense. Furthermore, the participants seemed to struggle with perceiving the height of sound sources. The study indicates that when creating an audio game, designers have to take the absence of visuals into account and adapt the gameplay accordingly, for example by using a slow pace to compensate for the increased cognitive load.

2.4.5 Return of the Obra Dinn

Return of the Obra Dinn[27] is a puzzle game in which the player has a device which lets them go back in time to witness the deaths of multiple passengers on a ship. The goal is to uncover the cause of each death and, when applicable, identify the culprit. The "witnessing" consists of a three-dimensional walkable freeze-frame of the moment of death, along with an audio-only cutscene of the final seconds leading up to it. While not being an audio game, the sound design of *Return of the Obra Dinn* plays an important role, as these cutscenes often include voice lines from characters involved in or connected to the incident, serving as some of the most important clues available to the player. Details such as accents become crucial for determining the identity of each passenger, which is a major part of the game.

2.4.6 Portal

Some puzzle games such as *Portal*[28] are sectioned into various levels or test chambers where each one is its own puzzle. Usually the player has to complete one or several level(s) in order to move on to or unlock the next one(s). These games teach the player various gameplay mechanics and tips that will be useful throughout the game, continuously adding new knowledge on top of what they learnt previously.

Portal features the character GLaDOS, who serves several functions throughout the game. Initially, she acts as a guide, providing instructions about the test chambers and introducing gameplay mechanics. She also delivers exposition, giving the player information about the setting and gradually revealing elements of the story and world. As the game progresses, GLaDOS is revealed to be the primary antagonist. She transitions from a seemingly helpful guide into an active opposing force, using her control over the facility and its test chambers to obstruct and threaten the player. This role reversal establishes the game's central conflict and motivates the player's escape from the facility.

2.4.7 Half-Life: Alyx

Half-Life: Alyx[29] is a VR first-person shooter game with room-scale. In this game, the room-scale allows the player to naturally look around from different angles, lean out from behind cover, and make small positional adjustments by physically moving within their play space. However, because the game world is a lot larger than the limited play space available to the player, most movement through the environment is handled through controller input.

The game's controller-based movement options are as follows:

- "Blink: Teleport to destinations with a brief screen fade. The most comfortable movement type." (See Figure 2.6.)
- "Shift: Teleport to destinations with a fast linear movement."
- "Continuous: Move continuously based on your head orientation."
- "Continuous hand: Move continuously based on hand orientation."

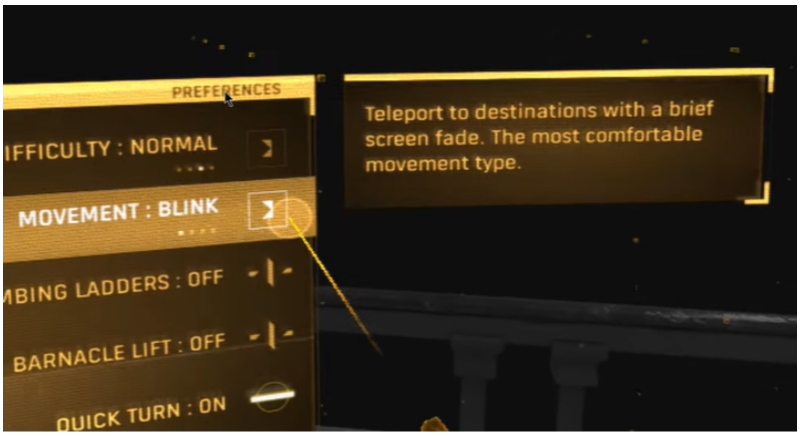


Figure 2.6: Half-Life: Alyx blink movement option

The reason the game describes the first option as the "most comfortable movement type" is that the other movement types may cause motion sickness, since the visuals change in a way that does not match the player's physical sensations. The blink option mitigates this by fading the screen to black, teleporting the player, and then fading back in.

2.5 Audio Game Guidelines

In *Design Guidelines for Audio Games*[30], the authors present a set of guidelines intended to support designers in the development of audio games. The guidelines presented in the paper include, but are not limited to, the following:

- "Use audio for all game events, and provide enough time for player reactions."
- "Allow repetition of information."
- "Provide immediate sound feedback for player inputs."
- "Avoid reproducing many sounds at once."
- "Try describing objects with the most accurate and representative sound possible."

The authors suggest that these guidelines can aid in the design of audio games and improve their accessibility for visually impaired players[30]. Although the paper primarily focuses on the accessibility aspects of audio games, the research is applicable to audio games in general, as many of the principles discussed are relevant to audio game design regardless of the target audience.

3

Theory

With the chosen question for this project there exists some information to find about other projects that have also worked with spatial audio. Below contains both explanations of useful terms that will be used throughout the paper as well as brief explanations of developments that other teams have made. These are chosen to help explain choices, and according to what is most likely to be utilised in different parts of the development of this project.

3.1 Spatial Audio

Spatial audio refers to techniques for recording and reproducing sound in a way that allows listeners to perceive it as occupying a three-dimensional space. It aims to provide sound cues to the listener to give information about their environment, making sounds appear as if they are originating from specific directions[31].

It is common for three-dimensional games to use spatial audio to enhance player awareness and immersion[2]. By positioning sound sources within a virtual environment, it enables the player to perceive position and movement of objects and characters. This may include auditory cues such as the footsteps of an enemy or environmental sounds like a waterfall. As a result, spatial audio can be a great help in both navigation and situational interpretation by providing information about elements in the game world which may not be visible to the player.

When spatial audio is reproduced through stereo loudspeakers or headphones, listeners are generally able to perceive the direction of a sound source with considerable accuracy. However, when a sound source is positioned directly in front of or behind the listener, **front-back confusion** may occur[32]. This phenomenon arises because sounds located symmetrically around the listener's median plane produce identical binaural cues – specifically, **interaural time differences (ITDs)**, the difference in arrival time of a sound to each ear, and **interaural level differences (ILDs)**, the difference in sound intensity for each ear caused by the head shadowing the sound. For sources directly in front or behind, ITD and ILD are the same at both ears, creating the so-called cone of confusion, where lateral cues alone cannot disambiguate the sound's position[32].

Front-back confusion can be resolved through slight head movements, which introduce dynamic changes in ITD and ILD, allowing the listener to perceive the true direction of the sound source[32]. This represents a key advantage of using a VR headset, as its head tracking allows the user to control how much and how fast they rotate. Furthermore, head movements such as tilting may also allow them to infer the vertical direction of the sound source.

In natural hearing, humans are capable of perceiving the full three-dimensional location of a sound source, despite only having two ears. This is possible because the shape of one's body, head and ears alter incoming sounds differently depending on its direction. The brain is then able to interpret these alterations and deduce a perceived direction. **Head-related transfer functions (HRTFs)** mathematically model these alterations and can be applied to audio signals to recreate spatial sound perception using only two output channels, such as headphones[33].

A significant limitation of this approach is that HRTFs are highly individual-specific, as they depend on the unique anatomical features of each listener. To address this, **non-individualised or generalised HRTFs** are sometimes used, which are derived from averaged measurements across multiple subjects or from generic head models. While generalised HRTFs can be used to reproduce spatial audio, listeners may not perceive the sound as originating from the intended location and can also reduce the listener's sense of presence[33].

3.2 Immersion

Games have become more immersive over the years[34]. With technology advancing in both processing speed and power, there is a lot more ability to immerse the player in the world of the game. According to Cambridge Dictionary the word immersion means to become completely involved in something, with as example being "Total immersion in a videogame is almost like living another life"[35].

There are many different types of ways a person can be immersed. Be it feeling like they are in a different place, completely immersed in completing a task or feeling like they are part of the story. In the text *Four categories for meaningful discussion of immersion in video games* they suggest to split immersion into four different categories; systems immersion, spatial immersion, empathic/social immersion, narrative/sequential immersion.[36]

Their different aspects are usually split differently according to how the author groups their meaning. An example is how empathic/social immersion and narrative/sequential immersion are grouped together in *The Psychology of Immersion in Video Games: How Developers Craft Realistic Virtual Worlds*[37] while also separating audio- and graphics immersion. A basic overview of the spirit of these different types of immersion are:

- The player being immersed in the mechanics, challenges, and rules of a game. They are immersed in the game's system and is usually described as the state of "flow"[36].
- The player is immersed in the world itself, they feel as if they are present in or have been transported to the game world[36]. Both graphics and sound design

are key aspects of this type of immersion[37].

- The player can also be immersed in a game's story, feeling as if they are part of the narrative, and how they can relate to characters and events[37]. This immersion is related to a game's plot and dialogue, how the player feels the need to continue to explore and furthering the story[36].

The immersion that will be focused on here is the auditory aspect of spatial immersion.

Spatial immersion focuses on the aspects that make the player feel like they are present in the game world. Like they are standing at the cliff edge they see on screen, or that they can hear how the wind whistles through the wheat field. The visual quality of the game is an important aspect to create this form of immersion. With realistic and cinematic visuals for 3D worlds being the most likely to simulate a player's transportation into said world. It focuses on making the player feel like they embody their character.[36]

Sound effects are another way to draw the player into the game world[37]. Sound, in of itself, is a well known way to increase a player's sense of immersion in virtual environments[38]. While relevant sound effects to complement events and actions are needed to make a game feel complete and responsive. Dynamic soundscapes made from environmental noises bring the virtual world to life around the player[34].

The intersection between these two points boils down to the environmental sounds that helps simulate the transportation into the game world. Directional audio is one way to help the player feel like they are standing inside the game[34]. While multiple, small, sounds that together make up a background symphony is another, just as the correct background music helps set the tone and make the player feel touched by the virtual world[37].

This project will focus on how to make the world feel more real through added depth in how sound moves throughout a space. How does sound echo in different sized rooms? How is sound affected by a player's placement in said room? Can they hear how frequencies mute as the player follows the walls? All of which will hopefully be easier to realise in VR.

3.3 Audio Techniques for Spatial Immersion

In game development, it is important to consider how computational processes can be simplified to improve efficiency and performance. In the context of audio games, this includes understanding how sound is affected by the environment, as well as how the human ear processes three-dimensional sounds to perceive their locations and directions.

3.3.1 Collision

Currently game audio is made by recording sounds beforehand which can be played back when the corresponding event occurs in game[39]. The in-game collision event is a good example of where this is used. It is well known and easily tested that a real life collision, such as knocking on a table, will sound slightly different each time. There are many aspects that factor into these differences, a few of the important ones are; the shape of the object, the location of the impact, the materials of the objects colliding and the force used.[39]

One way to simulate this is to take many samples of a specific collision and then play-back the one that is closest to the final collision. The sound amplitude can be altered to correspond to the force of the hit, making harder hits louder[39].

This is an example of how a single prototype sound can be slightly altered to create a larger depth to the final audio[39]. This way allows for manipulating sounds in real-time by processing it with parametrised "effects"[39].

Other such effects of note is the timbre which is affected by where on the object the collision occurs[39]. With the knocking on a table example, this would be the difference in sound when knocking in the centre as compared to where the table legs connect. The sound for each point can be calculated using linear interpolation of the placement to get the proportions of the different resonances for each spot[39].

3.3.2 Diffraction

The report *Efficient 2D Sound Propagation in Video Games*[38] proposes a ray-based sound model for simulating how sound travels around corners.

This model focuses on simulating three aspects of how sound travels – diffraction, reflection and transmission. Diffraction describes how waves moves around corners, such as light waves, sound waves, or water waves. Reflection explains how waves are bouncing back from a surface, how much of the light or sound is reflected away from the wall. Transmission refers to how sound penetrates through surfaces.[38]

The terms diffraction, reflection and transmission will exclusively refer to sound waves throughout this paper unless stated otherwise. Diffraction and transmission are used for calculating how sound moves around obstacles in this model. With transmission supplementing for the places that are not properly accessible by their diffraction model. The reflection part of the model is mainly used to simulate echoes and delays in sound propagation, instead of exploring the environment.[38]

The model functions by creating secondary sound sources at vertices that are visible to both the primary sound source as well as the occluded area. These sources of sound have a lower amplitude decided by the sound propagation equation. Then iterating for the new sources as long as the amplitude is above zero and the placement conditions for creating a diffraction source is met.[38]

These are the basics of the spatial sound model by *Efficient 2D Sound Propagation in Video Games*. We plan to emulate their model to have a good starting point when it comes to how we want sounds to travel within our game. But with added aspects such as attempting to add a generalised HRTF for increased spatial understanding, as well as taking heed of how different frequencies are affected by aspects such as head shadowing and the player standing close to a wall.

4

Methodology

There are a lot of different methods that can be used when doing a project such as this one. Below are some of the methods entertained for this project or used to help clarify other methods. This includes ideation methods, software development methods, and evaluation methods.

4.1 Research Through Design

The main method used in this paper is the **Research Through Design** method[40]. It covers all parts of the process. Research Through Design allows the developer to research at the same time as they get to practise the craft. Letting them do research and design at the same time. It is useful when the researcher also wants to work on the development of a product or function related to the question.

Today, Research Through Design is popular with interaction design students for human-computer-interaction. It allows them to develop and test a product that they have created which can showcase the research they are doing. This is especially useful for research questions that concern either the *Design itself* or *People*.

When it comes to the design itself it works by allowing the developer to work with the materials and technologies. It allows for engagement with the resulting situations or programs. Which helps the researcher discover further insights and new questions about the subject. Many of which are only discoverable through actually building the designs and doing the experiments.

This method is also useful when it comes to analysing people. How they think and behave in different situations. Why they interact with a specific part of the program or reasons why they are not interacting with something. All of these are questions where the Research Through Design method is useful to find a more complex answer.

> "Ultimately, Research through Design allows us to translate the direct experience of building designs and interacting with them into scientific knowledge."
>
> — Mehmet Aydin Baytas (2022-07-07)
> Design Discipline, https://www.designdisciplin.com/p/rtd-origin, accessed 2026-05-20

4.2 Ideation

First of all, the research question for this project could be described as a **wicked problem**[41]. Wicked problems are complex, open-ended problems that do not have one correct answer or solution. There are several methods one can use to tackle a wicked problem, and the solutions can vary greatly. Even by using the same methods one could obtain vastly different results. Furthermore, there is no definitive stopping point for when a solution to a wicked problem is complete, the designer must decide when the results are satisfactory within the given context.

One way to tackle a wicked problem is through the **Double Diamond**[42] design process model. This model consists of four phases – discover, define, develop and deliver – encompassing the entire design process (see Figure 4.1). It works by diverging and converging, or expanding and condensing, the problem and solution. The first diamond contains discover and define, and is focused on the problem. During the discover phase, the designer diverges to explore, research and understand the problem. In the define phase they converge and combine what they have learnt into a more clearly defined research question. This diamond can be iterated upon until a satisfactory research question has been defined. The second diamond contains develop and deliver, and is focused on coming up with a solution to the problem. In the develop phase, the designer will brainstorm, experiment, explore and test various ideas. Finally in the deliver phase, the designer will again converge and narrow down to focus on the best solution(s) and create prototypes and evaluate them. This second diamond can be iterated upon until a satisfactory solution is achieved.[42]

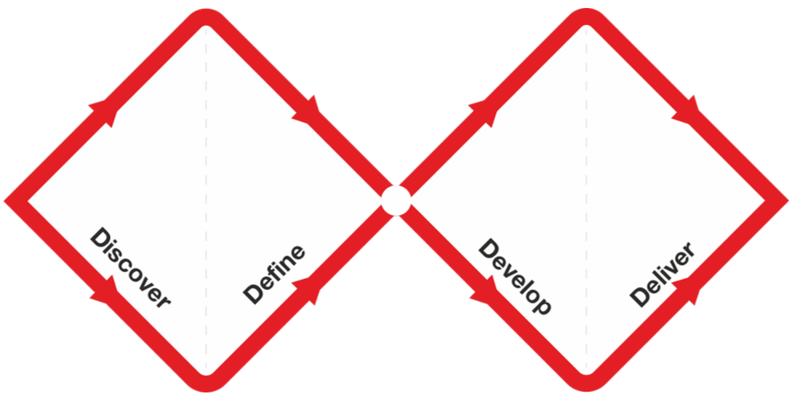


Figure 4.1: The Double Diamond design process model[7]

4.3 Development

Agile development[43] is an iterative and flexible approach to both project management and software development. It breaks down everything needed to create a larger project into thinner slices. Allowing for iterative **sprints** to be used during

[7]Image from the Design Council, `https://www.designcouncil.org.uk/our-resources/the-double-diamond/`, licensed under CC BY.

development, which helps break down everything required into more comprehensive points.

Sprints are often used in conjunction with the **Scrum framework**[44]. Scrum is a way to organise sprints for teams, utilising defined roles such as Scrum Master who makes sure each sprint runs smoothly, and Product Manager who sits in on meetings to make sure that the project is still continuing in the desired direction.

Together this helps deliver functional, high-quality software quickly. Each sprint contributes with delivering high-value, incremental improvements. For software development the sprints are often formatted to take heed of the agile technique **vertical slicing**[45]. Vertical slicing focuses on creating connected aspects all in the same sprint. Meaning everything from the user-facing UI to the back-end code, this allows for easy, continuos, testing throughout the development process, helping to mitigate integration problems down the line since related aspects are created together. It allows for the focus of each sprint to be tested and showcased to the team, Project Owner or investor.

The counterpart to vertical slicing is **horizontal slicing**[46] and refers to a development process where the focus is on developing one technical layer at a time, finishing one level before the next is started. This type of development is good for projects where every tiny aspect have to be finalised in advance, such as building a house. It is rigid in its ability to adapt and, for software engineering, it would require an exorbitant amount of time to both plan how to implement and then later again to fix errors found during integration and testing.

When working with Agile development and especially with the Scrum framework, it is usually combined with **User stories**[47]. User stories are a way to write tasks – also called cards – in a way that is clearly defined. They are structured as such:

> As a [role] I want [goal], so that [benefit].

This gives the developers a clear view not only on which aspect of the project the card is focused on, the [Type of user/persona]. It also gives the task itself in the form of the [goal/objective] this person wants, as well as a clear indication of when this task is considered done in the form of its [benefit/result/some reason]. *Trello*[5] is a simple online workspace that allows for the creation of cards which can then be moved between card-columns, usually named according to the Scrum framework when used for such. Figure 4.2 shows an example of a *Trello* card user story.

4.4 Evaluation

There are a lot of different methods when it comes to testing and evaluation of a product. These can broadly be split into two groups: **quantitative** and **qualitative** data[48]. Quantitative data is the information that boils down to numbers. The quantitative data gathering method that is most commonly used by the authors is **questionnaires**. Questionnaires can be sent out to users without requiring time from the creators for each user. Quantitative data becomes useful when there are enough participants that the predicted level of error enters acceptable ranges.

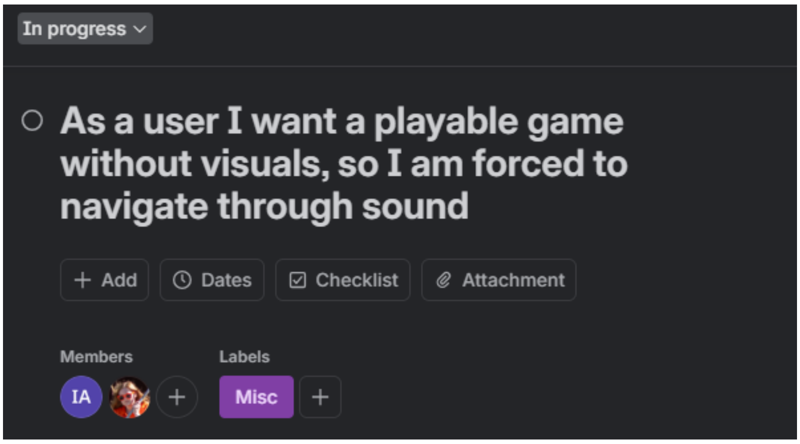


Figure 4.2: A Trello card in the shape of a user story

The other group of evaluation methods belong to qualitative data gathering. This group contain more personalised methods which takes up more of the developers' time to gather. **User observations**, **talking-out-loud method** and **interviews** can be found here. User observations allows the information gatherer to observe the user as they interact with the product, it helps figure out how a user interacts with the product without it being explained to them by the creator. The talking-out-loud method, if used, is used in conjunction with observations because it encourages the user to speak their thoughts out loud to help the observant understand what is going through the user's head.

Another evaluation method that is well recognised is the conduction of interviews. Each interview can be set on a scale from a **formal interview** to an **informal interview**. A formal interview is more structured with questions created in advance and the interviewer is specifically interested in the answers to the preplanned questions. An informal interview can be seen as more of a guided conversation. Here the interviewer has a few guidelines to follow to gain information regarding parts of interest. Except for subtly guiding the interview to revolve around these subjects, the interviewer is more interested in gaining information surrounding the question that might not have come up in a more formalised interview.

Two possible approaches to project evaluation are **summative** and **formative evaluation**[49]. A summative evaluation is an assessment conducted after the completion or a project to determine its merit, worth or value. These are useful in for example large-scale projects to justify their expenditures and/or continuation. A formative evaluation, on the other hand, is conducted during the development process and is intended to support improvement and iteration. It is for example used to find strengths and shortcomings while the project is still in progress, consequently improving the eventual finished product.

Another method of project evaluation is conducting a **SWOT analysis**[50]. SWOT stands for strengths, weaknesses, opportunities and threats, and aims to provide a clear understanding of factors that influence performance. A SWOT analysis is divided into two parts: analysing internal strengths and weaknesses, and analysing external opportunities and threats. The internal analysis identifies, for example, the source of competitive advantage and the resources needed to complete the project. The external analysis considers the broader perspective and explores aspects like implications for the future.

4.5 First Plan

The first plan was made before the prestudy took place. The plan was to begin by researching the topic of audio design in games and its affect on players. Additionally, the plan was to look at the design of VR games. Before the project started, we made sure that there would be enough pre-existing research within the scope of this project to guarantee that it would be a solid academic foundation for the work.

By reading about, watching and playing games, the hope was to gain a better understanding of how music and sound effects are, and can be, used to create the desired emotions and reactions in players.

At this point in time the focus was still on the development of an audio simulation that could be utilised for games. Therefore the desired outcome was to design and develop an audio simulation that is accurate and efficient enough to stand on its own, so that it can be used in games to let players navigate through the environment using sound.

The plan was then to test and evaluate this simulation by designing a game around it. It would be a VR game without visuals, to further force the player into navigating based on what they hear. The VR component would be helpful as it would give the player more control over when and how they rotate and orient themselves in the world, as they could do it by moving their physical body rather than just through the press of a button. VR's head-tracking can also be utilised to easily transfer the player's direction into the game.

An overview of how the plan was structured follows below, as well as a simple Gantt chart to give an overview of the plan. This chart can be seen if figure 4.3.

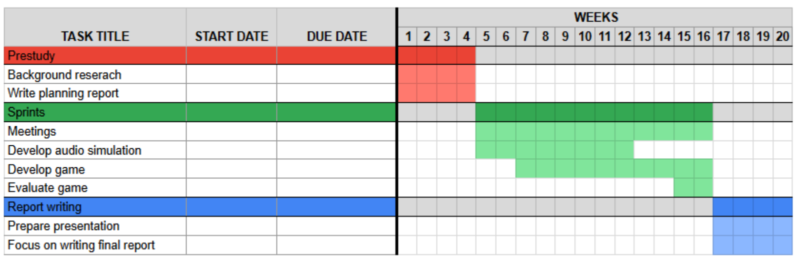


Figure 4.3: Gantt chart of the expected project timeline

The first four weeks were meant to focus on the prestudy. This contained three parts. The first focus was on finding out what others have already done within this area. The second is to make sure to write the planning report. And the last was to keep an eye out for aspects that could be useful for the final SWOT analysis.

The following twelve weeks would then be mainly filled with creating the VR experience. The work here would be split into weekly sprints for creating the program, with time set aside to write in a project diary. This was to make sure that we did not forget any potential problems that could occur during the project. It was also meant to track why the problems occurred and how they were solved. This would later aid in the writing of the thesis. Beyond this the plan was to once again make notes for the SWOT analysis as well as keep up with small additions to the project report.

The last four weeks were meant for finishing the report. This was also where the preparations for the presentation were to be done.

5
Process

This chapter describes the project process, including the prestudy, the design and implementation of the game, as well as the evaluation of the game, its audio design and the implications for audio VR-games in general.

5.1 Prestudy

The first four weeks of this project was allocated to being research time. Here we researched topics that we believed would be useful for this thesis. At the time our intention was to research different ways to do audio simulations in both academia as well as in other commercial games.

During our search for previous research we made sure to save all the relevant sources in a shared document. This document was meant to contain information about each source we found so we did not have to go back and re-read them again just a few months later. This information included: a link to where it was found, which one of us found it, when we found it, a short summary written by us, and potential quotes from the source material that either explain its contents or that we thought would be of use later.

This sped up our ability to search and check for relevant sources while the concise summaries and relevant quotes made it easier to come back and quickly find which source was relevant for which part of the project, without having to re-read them all.

During the search we gravitated towards different aspects of the research as to not overlap each other. One of these parts revolved around how sound functions in reality, with a focus on how we perceive it. The other focused on how sound have been simulated digitally, with a later focus on game like environments. Most of the sources were found using *Google Scholar*, but we also looked through *IEEE*, and *ACM*. Some of the search terms used were "audio games", "spatial audio", "virtual environments and games sound rendering", "computer games spatial sound", "game world ambient sounds", "game audio sound design", and "realistic 3D sound for games". We also made sure to look up the original sources of interesting papers. This helped give us a better understanding of the concepts and references used by said papers.

As part of the prestudy phase, one author also conducted research by playing relevant games. One of these was *Half-Life: Alyx*, which provided first-hand experience of VR-related motion sickness when testing continuous movement. The author also played the audio game *A Blind Legend*. Although the game itself lacked visuals and therefore did not provide any visual input, she found herself distracted by visual stimuli in the surrounding environment. This prompted her to use a sleeping mask to block out distractions, allowing her to focus on the audio experience. This, in turn, both improved concentration and increased immersion. We noted that this benefit would come naturally with our planned use of VR, as the headset would effectively block the player's entire field of view and remove external visual distractions.

A large part of the idea for this project was to create a game to get a better understanding of all the parts that goes into making a good audio simulation. We chose to use the Godot game engine for this, since we have had positive experiences with it in the past. Our project idea was adapted when we discovered the *Raytraced Audio* add-on. The explanation of this add-on implied that it contained everything from audio panning and muffled walls to diffraction. The existence of already created spatial audio functionality made it redundant for us to take the time to create our own when we could utilise the pre-existing add-on and instead focus on the next step in the process, the gameplay experience for audio-games made for VR.

5.1.1 Ideation

The ideation for our audio-based VR game was started almost as soon as the idea for a game without visuals was established.

At this point the idea was still to create the backend for a more realistic audio simulation which is why the focus was on either finding a sound source or to map an area through sound.

There were quite a few different ideas for what kind of game to make. One of the ideas was to create a game where the player is meant to walk around an empty room to figure out its shape, after which they get a visual cue to choose the room shape via a multiple choice question.

Another idea was for the player to be completely stationary and the point of the game was to face (and/or point towards) the sound source to showcase that it had been located.

Different aspects of these ideas got merged to create the idea of the player being able to move around the room in an attempt to get as close to the sound source as possible to point at it. The ability to move around obstacles not only allowed for higher accuracy it also allowed for more complex levels.

This in turn then congealed into the idea of a game where the player is meant to locate and walk into the sound source's hitbox. The plan was to split the game into different levels, where the layout of each room increases in difficulty. It was decided to use this approach since it would allow us to test how well the players could navigate the space using the created spatial audio simulation.

This idea was then further refined into being a completely audio-based game without any visuals. This decision was made to reduce the necessary work needed to create the game itself, as well as to remove factors that could give misleading results. These include visual inputs that are used to represent sounds, such as echolocation, or players being able to guess the most likely placement of a sound source in a visible room.

When we, towards the end of the ideation period, found the *Raytraced Audio* add-on the focus slightly changed in response. Since a simulation of realistic audio had already been made for the Godot engine, we shifted our focus to instead analyse players' experience playing an audio game, with a focus on spatial awareness and navigation towards a goal.

Here we truly started discussing the game's story and themes. Since we wanted to analyse the players' gameplay experience, we needed to create an actual game they could experience. This is what sparked the idea of creating a more complete game.

The lore of the game, *Forest Mole*, was further iterated upon during the first few weeks of development while a **Minimum Viable Product** (**MVP**) was made. These future changes were largely superficial such as alterations in voice lines and tweaks in level layouts to create an increasing difficulty. The largest alteration in this stage of the ideation was the inclusion of a separate entity – which we call "Robot" – that the player was meant to lead to the sound source. The reason for this was both to add a further element of difficulty as well as to break up the monotonous gameplay of just walking towards the sound source.

5.1.2 Game Idea

At this point the idea was to make a game where each level consisted of the player locating and walking to the sound source. This sound source has been called the "music box" during development. The levels were meant to be easy to traverse and lack factors that could take away from the goal of finding the music box.

We wanted to add player-made sounds, both to give them a better feeling of control as well as to help immerse them in the world by giving them clear cause and effect feedback. We also wanted to attempt to make a clear distinction between being in the menu and actually playing the game. Our plan for this was to keep the non-diegetic sounds to the menus only. This way the player would not get thrown out of the immersion because the diegetic voice lines were interrupted by non-diegetic instructions.

Together this created the game idea of traversing a bare bones area all the while something intrinsic to the game world is in the logical position to give information to the player. With the further logic being that this person who gives exposition also gives it to the protagonist for an understandable reason that fits with the overall game world and its lore. At this point in time the more concrete parts of the plot were yet to be nailed down and only the abstract overarching idea had been decided. The development of the lore, the world and the narrative continues in *Section 5.2.2 Manuscript* found later in this chapter.

5.1.3 First SWOT Analysis Iteration

The SWOT analysis in this paper will be about audio VR games, many which are lacking visuals entirely. Below is a short summary of the Strengths, Weaknesses, Opportunities and Threats identified during the prestudy, by reading about, talking about and playing audio and/or VR games.

Strengths

- You can play these games if you are visually impaired.

Weaknesses

- If you are hard of hearing, you will most likely not be able to play these games.
- It may be difficult to change/select language.

Opportunities

- Job opportunities for audio designers.
- Could be used to train people to navigate through sound, in case of emergency.

Threats

- Games requiring more powerful/expensive hardware.
- Games becoming more expensive.

5.1.4 Updated Plan

At the end of the prestudy, an updated plan for the rest of the project was created. Given our research question, time restraints and hardware available, we chose to work in the way described below. Much of it is similar to the first plan, but more detailed. Figure 5.1 shows a Gantt chart of the updated plan.

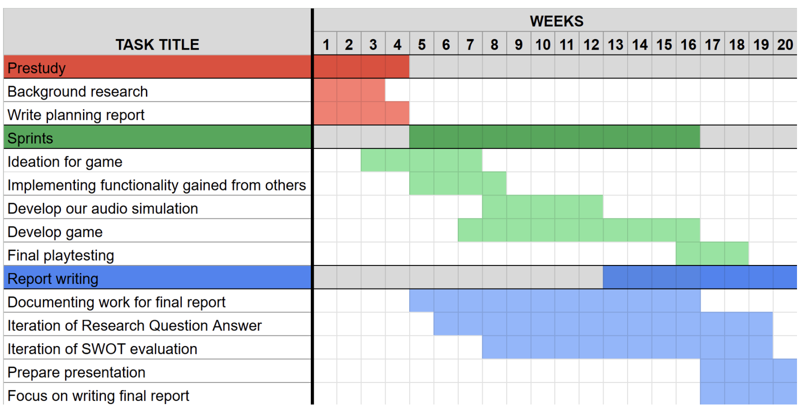


Figure 5.1: Gantt chart of the planned process

Our timeline was still split into three parts. The first one encompassed the first four weeks and focused on three things. Firstly, gathering information about the following topics: audio games, how spatial audio can be integrated, useful parts of VR and how our hearing works. Secondly, creating the game idea and then iterating over said idea to refine it before we start programming. And lastly, putting everything together into the updated plan.

The second part focused on designing and developing our audio-based VR game using the Godot game engine. The time would be split into twelve sprints spanning twelve weeks. Each sprint contained the regular start-up and end meetings as well as mandatory notes of what we had accomplished each week, problems we had run into and the like. The hope was to set aside time to sporadically write about our development process in-between the sprints.

The hope was that it would not take long to implement the basic functionality mentioned in the Theory chapter, with the plan being to have it integrated after the first four sprints and potential polish being added down the line. That way we could continue building onto what already existed and focus on the player experience as well as refining the implemented aspects. At the end of these twelve weeks we planned to have a game – or well-developed part of a game – which would showcase an audio VR game made with gameplay and player experience in mind.

The last four weeks would focus entirely on catching up with, completing, and refining the final report, as well as presenting the project. This approach would ensure that we did not place too much emphasis on creating and tweaking the game aspects of the project at the expense of the report itself. During this final stage we would finalise the SWOT analysis, which would have been continuously iterated upon throughout the game's development.

During the last few sprints, and the start of the last stage of our timeline, we also planned to evaluate our game. The feasibility of this evaluation would depend on whether the game had reached a playable state. If so, we aimed to involve external users in playtesting the game and providing both qualitative and quantitative feedback. Qualitative feedback would be gained through playtests conducted in a controlled environment, where participants would describe their observations and experiences throughout the playthrough, complemented by informal interviews. Quantitative data, on the other hand, would be gathered through a survey. This would most likely only be possible if we were able to distribute the game to a larger audience. The collected data would then be used to do a formative evaluation.

5.2 First Development Cycle

The first cycle focused on getting Godot up and running as well as getting a functioning MVP to prove that the required functionality works. We started by setting up the VR implementation in Godot and making sure that it functioned properly. We then worked in parallel with writing the game's manuscript and developing the MVP, since these were possible to do simultaneously.

5.2.1 VR

The first thing we did was create and access all the shared spaces we would need for this step of the process. This included both the *Trello board* for our Scrum cards and the creation of a *GitHub* repository for our project.

When we had this repo and the ability to access the same code, we connected it to our blank Godot project. We were already aware that Godot supports the creation of VR games and it did not take long to download the *Godot XR Tools* add-on (Figure 5.2). This add-on is a library that provides VR functionality for Godot.

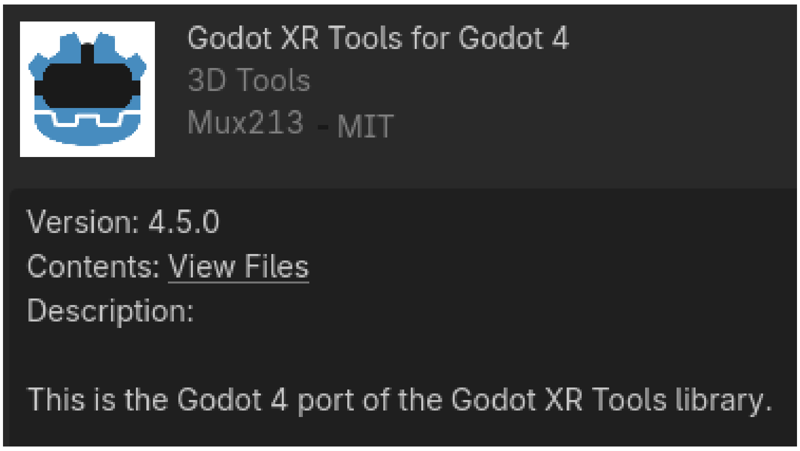


Figure 5.2: Godot XR Tools add-on

To be able to test this add-on and make sure that it functioned as intended we created a simple scene where we could test the basic features and make sure it contained all the VR functionality required for the game. Some aspects tested include the following:

- Head tracking to move and rotate the player character
- Detect and track physical movement of the VR-controllers
- Detect input (e.g. button presses) from the VR-controllers

These were the most interesting and important aspects for us, since they were the reason we decided to make a VR game. An example is rotation. When sitting in front of a screen it can be hard for the player to understand how much they are turning without visual input, something that is completely bypassed when the player can turn physically while wearing the headset. It was ultimately concluded that the VR functionality featured in the add-on would be sufficient for our project. With the add-on in place, we were able to test each aspect of the game as it was added.

During the testing of this add-on, however, we encountered a problem. We, the developers, used different types of VR-headsets: one of us used a *HTC VIVE* and the other used a *Meta Quest 2*. While the *VIVE* functioned normally, the *Quest 2* experienced issues. The moment the *Quest 2* was connected, the headset screen started flickering with each tick. After many hours of troubleshooting and attempts to alter the update rate, the solution was found in GitHub question thread[51]. The issue was solved by changing the “OpenXR API version” to 1.1.53 (see Figure 5.3). According to the thread, the current theory is that the problem stems from a bug on Meta’s – the developers of *Quest 2* – side, and that the workaround is to target an API version unaffected by this issue.

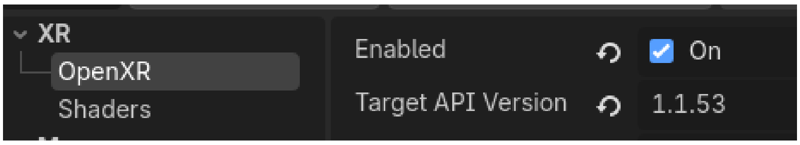


Figure 5.3: OpenXR API version

This solution, however, introduced another problem, as the API version was not compatible with the *HTC VIVE* headset. At a result, we removed the “.project” file from the GitHub repository. This allowed us to run our devices on different OpenXR API versions depending on whether the computer was intended to connect to a *HTC VIVE* headset or a VR device made by Meta. Surprisingly, this solution caused few issues later in development, which we were pleased with.

5.2.2 Manuscript

Once we were able to return our focus to the development of the game, we divided our time between two main tasks: the writing of the manuscript and the implementation of the MVP. Since these were developed in tandem, there may be some overlap between the sections describing them. The development of the manuscript also had a slight influence on the creation of the MVP.

The manuscript contained more than just the voice lines, it also necessitated an understanding for the game’s setting and lore, something which in turn led to constant communication between the developers to decide on these overarching story ideas.

Making the manuscript and its connected lore started with understanding what we wanted the gameplay to be. Not in the form of how the game is played on a technical level but more in the aspect of creating a “why” behind the player’s actions.

We knew since we started looking into how audio bounces, that we wanted to create levels that are simplistic in shape. This is because it reduces the difficulty of navigating the space when the only thing distorting the sound are walls that are exactly set at a 90-degrees angle to the floor. These type of environments are similar to the idea of controlled environments used for testing in real life. They remove any and all external factors that can skew the results as to better test their desired concept. In our case this is navigation through audio, so we put our idea of testing how to navigate through audio into the game itself, resulting in a setting that is similar to the test-chamber approach, for example found in the *Portal* games (see Figure 5.4).

[8] Image from Steam’s store, `https://store.steampowered.com/app/400/Portal/`, accessed 2026-05-12.

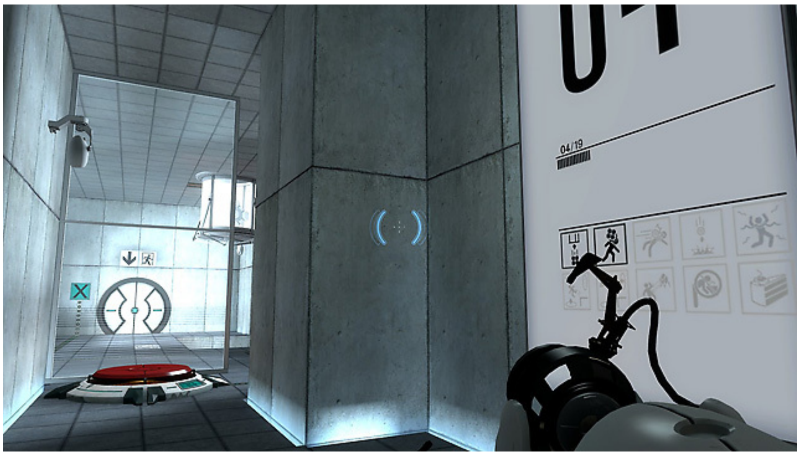


Figure 5.4: Portal test chamber[8]

We have taken more inspiration from the *Portal* games in the form of our "Handler" which functions as GLaDOS in the form of giving exposition to the player. We also utilised the Handler to give the player diegetic instructions on how to play the game. This is clear in voice lines such as the one found in level 1, which explains what the player is supposed to do in the level as well as how to access the pause menu just in case. A quote of the level 1 starting voice line is shown below.

> "Okay Seek. This is the first level so it will be easy. You just have to find the source of the music. Once you touch it I will mark the time it took. And remember: if anything comes up you can signal me to pause the training. Just raise your hand high like always and keep it there until I can pause the level. Now lets start, Three, Two, One, Music!"

This voice line also contains the ending "Three, Two, One, Music!", which we have decided to use as a further indicator of when the player can start moving, the first being that music starts playing.

With this information of what the player is doing as well as an idea of where the protagonist exists inside the in-game world, it naturally evolved into figuring out the "why" of the matter. Why are they being used as a lab-rat? And why them specifically? We came up with the idea of our protagonist being able to hear a noise, the music, which no one else in the facility can do. Combined with the concept of unethical human governments, turned into the idea of our player character having been raised in captivity so our Handler can monitor and understand how they work.

With the continued question of "why" this evolved into asking why the player existed and why testing the them is a priority. Therefore the lore of *Forest Mole* is that the protagonist is something human-adjacent which the human race is at war with. They are part of a concept testing to see if something like the protagonist can be raised to work against its own race. This is a desirable approach because only those like the player character can hear the music and when you can hear it, locating the source is as easy as following the sound.

The manuscript is filled with small details that hint towards the larger world being like this. But a game with monotonous gameplay and a clear villain and outcome can feel flat. Therefore we added two things. The first is the idea of the “Robot” which is a companion that the protagonist gains halfway through the game and at which point the gameplay switches from “walk to the sound” to “lead the Robot to the sound”. This is done through the player creating noise such as clapping their hands or bumping into walls. These are sound effects that exist on the Robot’s auditory spectrum and therefore allows the Robot to move towards the sound and the protagonist.

The second thing we added to make the game feel less flat is a choice. At the end of the game the player gets to choose if they want to complete their mission or join the rest of their original race. The “correct” choice is still fairly obvious but we did add a description of why the human and the protagonist’s races have such a violent conflict. The reason they are fighting, according to the humans, is that the protagonist, and its like, are “radioactive” and actively dangerous to humanity’s survival.

The lore developed throughout writing the manuscript made sure that the MVP contained not just a sound source, but also the functionality of voice lines both before, after and during the level itself. Later it would also be the first testing ground for the Robot once it was created.

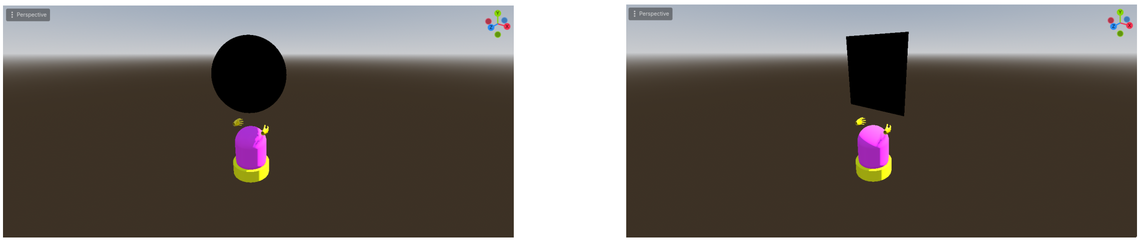
5.2.3 MVP

After connecting the VR-headsets and checking the *Godot XR Tools* library, the development towards the MVP for the game was started. The MVP was finished five weeks into development. It included much of the basic functionality used in the final product, albeit unrefined. This is because the MVP contained a basic functionality for all the aspects needed to finish a level. Said functionality was decided as the manuscript and lore for the game was developed.

During these first weeks of development, a large part of our focus was placed on the player’s interactions and controls. Since the player cannot rely on vision, it was important to provide feedback in other ways. Audio cues served as the main form of feedback, for example playing a hitting sound if the player touched an in-game wall, while haptic feedback complemented this by causing the controllers to vibrate while the player’s hand was inside the wall.

To achieve the effect of a game without visuals, we decided to place a black orb around the player’s “head” (the first person camera the player views the game from) to make the entire screen inside the VR headset pitch black. To make this work as best as possible we had to tweak the the radius of the orb. If the orb was too small, it ended up closer to the player than the near-plane (the cut-off point for what is visible), and became invisible to the player, rendering it useless. If the orb was too big, however, objects such as walls or the player’s hands could peek through. The orb worked fine as a placeholder for initial testing, but its ability to block out all objects was somewhat unreliable. Because of that, we later changed it to a plane and placed it directly at the position of the near-plane, which worked a lot better.

Figures 5.5a and 5.5b shows how the orb and plane, respectively, would look like from a third person perspective.



(a) First version – orb

(b) Second version – plane

Figure 5.5: Two versions of hiding visuals from the player

To quickly get the MVP up and running, many temporary sounds were added so that different features and mechanics could be tested.

The idea of the MVP consisted of one level shaped like a wide corridor with 2 horizontal walls in-between the player and the goal. Figure 5.6 shows the layout of the MVP level – the player starts on the far left side and the goal is to reach the music box (the teal cylinder) on the far right side. This level was tested both by the developers and some friends. Everyone that tested the MVP were able to locate and reach the sound source.

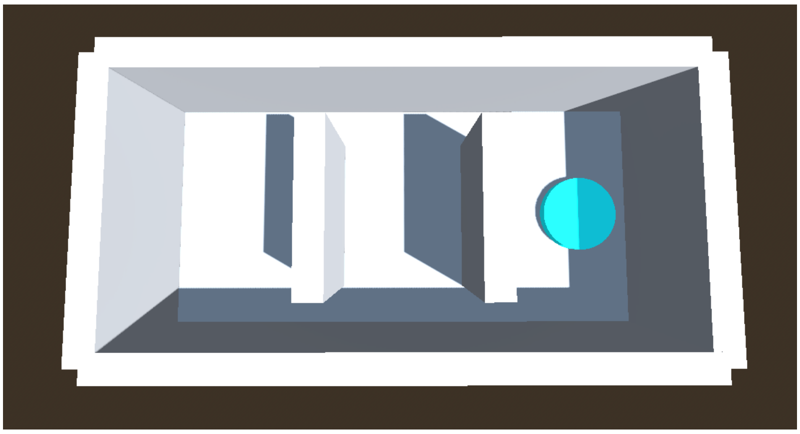


Figure 5.6: MVP level layout

The MVP evaluation was done very informally, simply having the players test the game and see if they could locate and reach the source. This allowed us to test our core concept, which was the goal of making the MVP. Since the informal playtesting of the MVP was successful (and sparked various ideas), we moved forward with designing the remaining levels, tweaking player interactions, etc. During the second development cycle, one featured added for easier development was a top-down "spectator" camera that made the game visible on the computer screen but not inside the VR-headset. This would allow us to know where the player is inside the game at all times during future playtests.

The MVP and the manuscript were finished around the same time, allowing for an easy segue towards implementing everything related to the game itself.

5.2.4 Second SWOT Analysis Iteration

During the first development cycle, additional aspects for the SWOT analysis were identified. The focus of the analysis is on audio VR games in general, but using our own game as an example. Therefore, some aspects may not fully concern audio VR games overall but are more specific to this game and games like it. Our focus on gameplay during this stage also made us more likely to identify strengths and weaknesses, which is why fewer opportunities and threats were found.

Strengths

- Our game can be played while stationary as compared to many other VR games and experiences.

Weaknesses

- Possibility of VR-dizziness from traversing the environment.
- VR games require having a VR-headset which might reduce their accessibility.
- It could be more difficult to convey the story and its meaning in an audio game, because of its lack of visuals.
- May require headphones with high quality.

Opportunities

- Highlights the existence of audio games.

Threats

- No threats were identified during this period.

5.3 Second Development Cycle

The second cycle focused on the development and evaluation of the audio VR game *Forest Mole*. This phase includes everything from designing and implementing menus and levels, to recording and editing voice lines. It also includes composing and producing music.

5.3.1 Menus

Forest Mole has two menus: a main menu that the player will enter whenever they boot up the game, and a pause menu that the player can access at any time while they are playing a level. When the player is in either of these menus, they can cycle through the menu options by clicking left or right on either of the controller's joystick (or touchpad in the case of the *VIVE*), and select their choice by using the controller's trigger.

Upon starting the game and subsequently entering the main menu, the player is met with the opening voice line: "Welcome to Forest Mole. Use the joystick to navigate the menu. Select your choice by using the trigger on the underside of the controller."

When a menu item is focused, the player hears a corresponding voice line describing what will happen if it is selected. Upon selection, an additional voice line is played as confirmation, providing feedback for the player's interaction.

The following menu items are available in the main menu:

- "Start Game" will start the tutorial level and continue through the rest of the game.
- "Select Level" will open a submenu where the player can choose any level from the tutorial through level 10, or return to the main menu. Selecting a level from this submenu will start the game at that level and continue through the remaining levels from that point onward.
- "List Controls" will play a voice line that provides the player with knowledge of how to navigate within the game, such as walking and making noise.
- "Enable Debug Visuals" will disable the black plane that hides visuals from the player. This option is only available when debug visuals are disabled. When enabled, this option is replaced with "Disable Debug Visuals" which re-enables the black plane and hides the visuals again. The reason we decided to add this toggle was, as the name suggests, for debugging purposes. This option made it easier for us to test various things inside the game without having to edit the code and reload the game.
- "Credits" will play a voice line that tells the player about who contributed to make the game and what they did, including the people who recorded voice lines and what role they played.
- "Exit Game" will play a voice line saying "goodbye" before closing the game.

Figure 5.7 shows what the main menu looks like with debug visuals enabled.

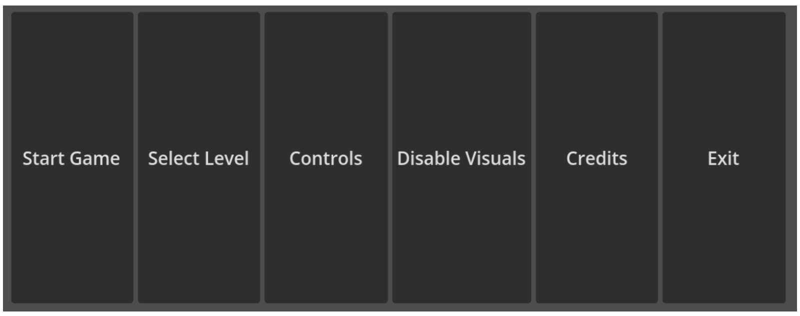


Figure 5.7: Main menu

If the player raises their hand above their head during a level the game will be paused. The reason we decided that this should be the way to access the pause menu is because it felt like an interesting use of VR as well as a diegetic way to signal that you need something. It also moved an input/interaction away from the inputs of the controllers, and we wanted to avoid too many inputs relying on controller button presses since the player will not be able to see the controllers and therefore might not be sure which button they should press (especially if they are new to VR and/or the used VR-components).

While the pause menu is active, all sounds will stop playing and the player is no longer able to move around the world.

The following menu items are available in the pause menu:

- “Resume Game” will close the pause menu, continue any sounds that were stopped and let the player to move again.
- “Restart Level” will start the current level from the beginning, including resetting the player’s position and playing the initial voice lines of the level.
- “Repeat Instructions” will play a voice line that explains the goal of the level.
- “List Controls”. See the main menu list for an explanation.
- “Enable Debug Visuals”. See the main menu list for an explanation.
- “Main Menu” will stop the level and return the player to the main menu.

5.3.2 Levels

The levels were organised in increasing difficulty, such as an early level being a straight corridor and a later level being a semi-circle. The levels were intentionally designed in a way that would allow us to test the audio for various layouts, with a focus on how sound travels in the game and how players interpret the game’s areas based on the given audio.

Both the tutorial and level 1 were made to be just a straight line, with the differing factor being that level 1 would require the player to move in the exact opposite direction from the tutorial. This would hopefully help the player realise that they have to make use of the full range of rotation for later levels. With this lesson being taught early, the hope was that it would help the players feel comfortable turning on the spot to locate the sound, as well as stopping them from assuming that they start a level facing “forward”. Level 1 can be seen in figure 5.8.

Figure 5.8: Level 1

Level 2 was made to introduce 90-degree corners while otherwise keeping to straight lines. To accomplish this the level was made to have an S-shape.

Level 3 was meant to test diagonal corridors, both to stress-test the *Raytraced Audio* add-on but also to dissuade the player from figuring out one wall and then only using 90-degree turns from there. This level laid down the groundwork for level 4.

Level 4 was made to introduce curves. It was originally meant to be a corridor shaped as a full circle, so that the player would be familiarised with the concept of sound being muffled through walls. The intention here was to have muffled sound come through the wall while unmuffled sound is diffracting around the circle. This was later changed to instead only be a half circle because the muffled sound through the wall was much louder than the sound that travelled around the circle. The half circle for level 4 can be seen in figure 5.9.

Figure 5.9: Level 4

Level 5 was the first time “pillars” were introduced. This level was designed to highlight how sound differs as the player approaches a wall, or in this case pillars. The functionality of sound changing when close to a wall was not included in the *ray-traced audio* add-on, and as such was something we wanted to implement ourselves. However, we knew that this would be too large a task to create within the allotted development time for the project. Instead, level 5 was used to introduce the player to obstacles in the room that are not directly affecting the room shape.

Level 6 went back to using simple 90-degree corners because it was mainly meant to introduce the player to the Robot and let them learn how it functions.

Level 7 once again was made to test the player’s problem-solving ability. Here the player is meant to start right next to the goal and then have to walk away from it to find the Robot so they can lead it to the goal. The Robot was intentionally situated on the other side of a wall, using the same muffle ability that was meant to be taught in level 4.

Level 8 was meant to test everything that the player had learnt up to that point. Not only does the player have to walk away from the muffled sound of the source to find their way to it, they also have to traverse diagonal corners together with the Robot and get around a small ridge at the end of a U-shaped corner just before reaching the goal.

Level 9 was made to help prepare the player for the ending level which takes place outside. Therefore level 9 was filled with pillars to imitate trees. The thought was that up until this point the player has been able to hear how sound is reflected on solid, smooth walls. This level introduced a more rough surface and with it a different echo effect from the sound. The point here was to let the player get introduced to how this type of environment alters the sounds they are used to before entering level 10 which was made to be three times as large. Level 9 can be seen in figure 5.10, the yellow capsule in the bottom right corner is the Robot.

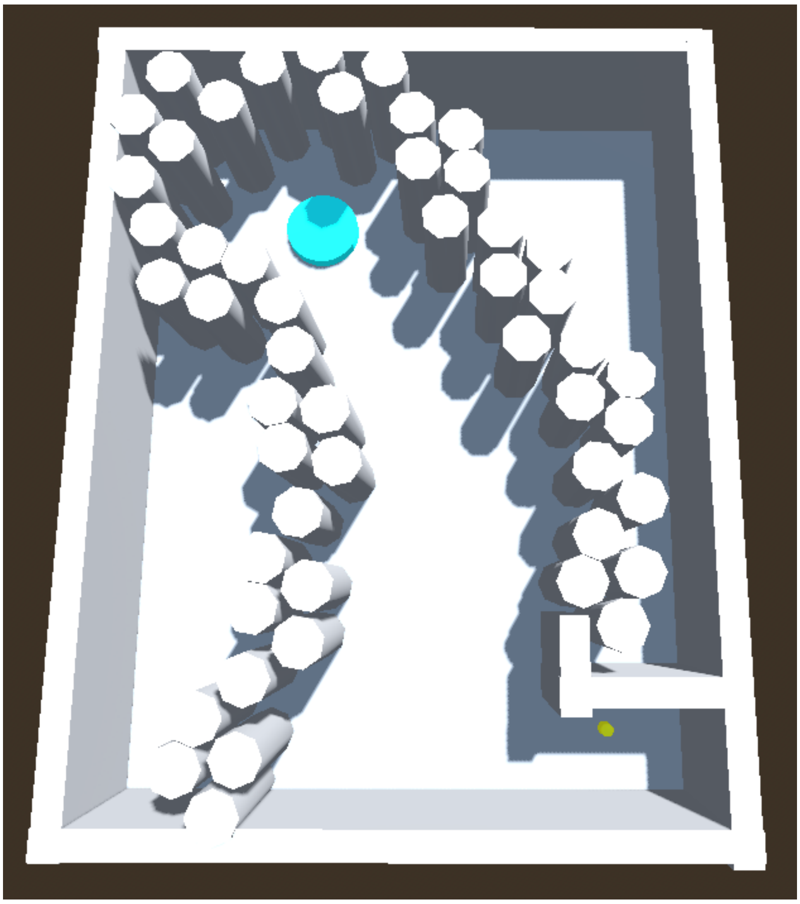


Figure 5.10: Level 9

Level 10 was split into two parts, the first being the actual level where the player finds the source like usual, and the later half focusing on allowing the player to make the big choice of the game. Here the player gets to choose a side through their actions, where the ending they get is dependent on that choice.

As we worked, we regularly playtested the game and its levels on ourselves. These tests were fairly informal, with one wearing the headset and the other taking notes. The one using the headset also used the “thinking out loud” method to both give feedback of what is happening at each point in time, as well as noting every time a bug or something else of note popped up so we could fix it the coming week. These playtests, and also later the more formal playtests, were made easier and more informative with the addition of the spectator view.

The spectator view is a top-down camera that shows the entire current level from the top. This view can be seen on a computer screen, but will not be visible inside the VR headset. It was added as a way for us to see where the player is inside the game, mainly to get more information during playtests (how is the player moving, where are they going, etc), but also to be able to help or guide the player if they get stuck. It also allowed us to more easily find various bugs related to player positioning, such as spawn points not working properly or players falling through the floor.

5.3.3 Sound Design

All audio used in this game was created by the developers, from voice lines to sound effects and ambient sounds. This is particularly noticeable in the game's sound effects, where familiar sounds were recreated using other sources. For example, the sound of an alarm clock used to wake up was replaced by clapping, and the "beeping" of the Robot was created using the pinging sound of metal chopsticks colliding. This type of audio creation is generally known as Foley work[52] and encompasses the recreation and recording sound effects and ambient sounds.

We used Foley work for all the sound effects we made, but the only time we used it to simulate a totally different sound was for the Robot. Since we did not have access to a real robot whose sound we could record, the Robot's sounds were instead created using an electric golf bag moving across different surfaces like concrete and gravel. The golf bag also ran into walls and trees to simulate the Robot doing the same.

In general we tried to keep the sounds as easy as possible by simply recording ourselves doing the actions of the player. One sound that could have benefited from us utilising tricks to imitate it was the "clap", since it did not sound like a clap when the recording was played back.

In addition to sound effects and ambience, the game required a large number of voice lines. A tool that was useful when recording and editing these was *Audacity*[53]. Ensuring that recordings were clear and free from excessive background noise was important, particularly for the voice lines, where distracting sounds could negatively affect the player experience.

One challenge we encountered was finding suitable voice actors for each role. To avoid delaying development, temporary voice lines were used initially and then gradually replaced as voice actors were recruited and their final recordings became available. Six voice actors took part in this project, two of whom were men and four women. One actor was in their 50s, while the rest were in their 20s.

The voice lines were recorded in two different locations, resulting in two different microphones being used for this project. This was taken into account when assigning recording locations. For example, all voice lines intended to sound as though they originated from the in-game speakers were recorded using the same microphone. In addition, an audio effect was applied to these recordings to further reinforce the impression that they came from a speaker and to ensure a consistent sound.

The game contained a substantial number of voice lines, with over seventy created for the menu alone and more than thirty recordings used throughout the game. Most recordings consisted of conversations between multiple characters or monologues with three to six sentences that provided exposition to the player. Only the level-ending recordings and menu voice lines were reused throughout the game.

Each voice line was recorded multiple times using different recording configurations. Factors such as the voice actor's posture (standing or sitting) and the distance between the voice actor and the microphone were varied between configurations to obtain a range of recordings. To reduce plosive sounds, we also constructed pop filters[9], one of which was made by stretching a piece of fine nylon fabric over a wooden ring. While this worked well to reduce unwanted noise caused by certain consonants, it unfortunately also attenuated some desired sounds, particularly the initial 's' sound in phrases such as "Start game". Another improvised pop filter consisted of a crocheted circle draped over the microphone, which also helped to remove these unwanted sounds.

After all voice lines for a character had been recorded, we began the editing process. First, all recordings were normalised to achieve a consistent volume level. We then assembled the strongest segments from different takes into complete voice lines. This was where *Audacity* proved especially useful, as it allowed us to combine the best portions of multiple recordings when necessary.

Ideally, an entire voice line could be used from a single take, as this required less editing and generally produced more natural-sounding results. This was important because the timbre of a voice could vary between recordings depending on factors such as microphone placement or whether a pop filter was used. In some cases, however, it was possible to repair an otherwise strong take by carefully inserting a missing sound, such as an initial 's', taken from another recording. Although this technique was used sparingly and only for minor adjustments, it proved surprisingly effective. Once each voice line had been edited to an acceptable standard, it was exported as a separate audio file for import into Godot and use within the game.

One of the last elements added to the game was the new music for the music box. The music and its melodies play an important role in the game's narrative, as they are connected to the protagonist's heritage. The fact that the protagonist is the only one in the facility able to hear the music is also the reason they were taken in the first place. Because of this, we wanted to capture the feeling of nature and forests, and ultimately decided that the music box should resemble the sound of wind chimes.

The composition itself consists of several melodies layered together in harmony. It was composed using *Noteflight*[54] and produced in *Cakewalk Sonar*[55]. *Noteflight* and *Cakewalk Sonar* were used because the developer creating the music had previous experience with these tools.

To create the sound of wind chimes, we searched for "wind chimes" on Plugins4Free[56]

[9]Pop filter: A noise-protection screen for microphones that diffuses sudden blast of air from consonants such as 'p' or 't'.

to find a suitable VST plugin[10]. Three plugins were found – *Breezy Day*[57], *Christmas Belly*[58] and *Wind Chimes*[59]. After testing all three, *Breezy Day* was chosen because it best matched our vision for the music.

There are three versions of the song used in the game: one instrumental version and two fully vocal versions. The instrumental version was created in *Cakewalk Sonar* using *Breezy Day*, while the vocal versions were recorded and edited using *Audacity*. In the instrumental version, many notes were intentionally slightly offset in timing to enhance the natural atmosphere while also introducing a subtle sense of eeriness. This version is played through the music box in levels 1-9.

One of the vocal versions consists of several recordings of voice actors humming different melodies to create harmonies resembling a large choir. The final version features a single melody hummed by the voice actor portraying the protagonist's mother.

The vocal version featuring only the mother's voice is heard during the tutorial level and serves two purposes. First, it supports the game's mechanics by signalling to the player that this song is the sound they are meant to follow. Second, it contributes to the story by suggesting that the mysterious music, audible only to the protagonist, is connected to their identity and heritage. This connection is further reinforced by hearing the choir version in the forest during the final level.

With all these components in place, we felt ready for the last week of the development cycle, which would, among other things, include formal playtests.

5.3.4 Last Week

Towards the end of the time allocated for creating the game we stumbled upon a problem – the volume of the sound dipped too sharply when the audio had to bend around corners. We were peripherally aware of this issue, and when we played the game during development we always altered the volume manually on one specific level that was affected by this sharp dip in sound. However, we were unaware of the severity and extent of the issue, and did not place too much focus on it. Therefore, we did not get it fixed in time for the first of our two final playtests.

These final evaluations were planned to be more formal. While the playtest itself was largely the same as during the testing for the MVP, with players playing through a demo of the game containing the tutorial and the first three levels, we did also have questions we wanted answered.

During the first main playtest the players were a few friends who had all played VR before on this setup. The focus of this test was, once again, on seeing how they navigated the space. We followed their progress using the top down spectator view of the level.

[10]VST plugin: A VST (Virtual Studio Technology) plugin is a software add-on used within a Digital Audio Workstation (DAW) to generate or process audio, for example virtual instruments or audio effects.

We had two main aspects that this test focused on. The first and main one was to test the navigability of each test. With this we tested how different level shapes altered the players ability to navigate as well as if the ability to find the source changed between the two different sound bites of the tutorial and the regular "music box". This part also included testing how much of a problem the rapidly decreasing audio was. Another aspect we focused on was getting feedback on what we should focus on for the future, this included both what the participants did not think worked as well as what they would want in the future. Beyond this we also kept an eye on potential ideas for the SWOT analysis as well as noting down any bugs that the participants stumbled upon.

Most of the participants of this test had a problem locating the sound source because they did not notice any difference in the sound panning. They also all had problem understanding what the controllers' vibrations meant or indicated.

With the feedback from this test and only a few days until the absolute final evaluation we focused on getting a work around for the quickly deteriorating volume. After many hours of looking we found a workaround by shifting the *Raytraced Audio* add-on's low-pass filter to something more useful for our music which contained almost only frequencies that were affected by this filter.

This was done in time for the final evaluation.

5.3.5 Final Evaluation

This playtest took place in a more controlled environment than previous ones, namely a separate, silent room with a lot of space. Since this playtest was open to a lot more people who might not have experience using VR or a *HTC VIVE* headset before, we made extra sure to explain the controls and how VR works in general as well as what aspects are most important for playing this game. These include the hint to turn in real life instead of using the built-in shortcut in-game.

We once again tested for navigability with the help of audio as well as making sure to keep an ear out for any ideas for future development, problems or wants from the play-testers. During the playtest we took notes on how each participant traversed the game and tackled problems. After the playtest itself we asked a few open ended questions to each participant, hoping to get their view of the gameplay with a focus on the audio navigation part. A few of the questions we asked are as follows:

- What worked well? What did not work well?
- What aspects made it the most difficult to play?
- What are your thoughts and opinions about the gameplay?

We had seven playtests in total during this session. Many of the participants mentioned the same aspects of the game and therefore gave similar feedback.

When it came to navigating the levels, all playtesters got stuck on the same place when encountering the problem of the music box being behind an obstacle. While they all were fairly good at finding the source of the sound when there was nothing between the player and the source, it got immediately harder when the music box was "out of sight". Figure 5.11 shows the layout of level 2, one of the levels participants often got stuck on. This level is shaped like an S, with the player starting in one end and the music box positioned at the other end.

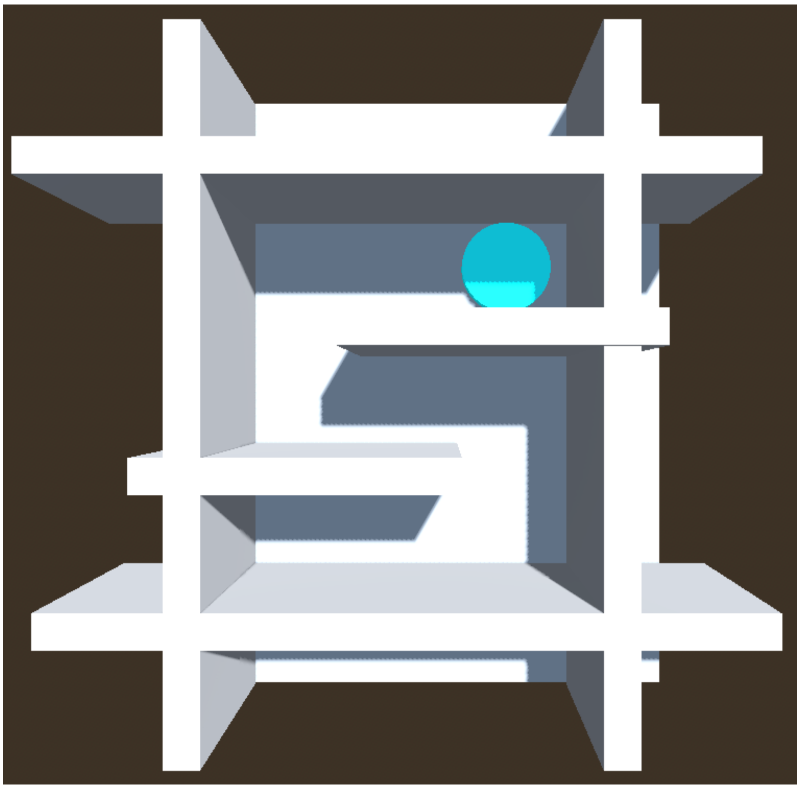


Figure 5.11: Level 2

They mentioned quite a few problems related to the navigation aspect. Many of the participants had a generally hard time locating the direction the sound came from. They had difficulty hearing the sound panning between the left and right headset-speaker as well as an inability to differentiate between if a sound is directly in front of or behind the player. Part of why some players had difficulty locating the source also stems from how they had a tendency to keep their feet stationary and kept their search view to a comfortable 180-degrees. It was not until they started spinning on the spot to take advantage of the full range of rotation that they could start moving in the right direction.

Another aspect that did not help matters is how the participants clearly heard the music box through the wall. If the music box existed on the other side of a wall all participants registered the sound as if it was coming directly through said wall instead of from around the corner as intended.

When it comes to the walking controls almost all players had problems understanding how fast they were walking and commented on how the sound bite was not affected by different walking speeds, something they wanted to be added. One of them also mentioned that they wanted the game to aid with re-positioning the character in game, such as moving them back to the start if they feel lost. The confusion of navigating the space was not helped by the problem of not noticing when their character walked into a wall or stopped moving forward. During our observation we noticed how multiple players simply stood facing a wall while they thought they were moving forward. Another thing that did not help with this was the lack of understanding of what the haptic feedback of the controls meant. One of the players even stood with both hands inside the wall and thought they were making progress. The participants were unified in that the haptic feedback was confusing and needed refining. Our choice of having the left and right joystick handle different movements was also criticised and something they wanted removed.

Only one participant explored the menu, and that only after having been told by us that it was possible. This playtester thought the menu instructions were not intuitive and therefore wanted clearer menu instructions. They also commented that they wanted the end of the menu to loop back to the start of the menu. This person also agreed with the other players that the voice lines were hard to hear and recommended subtitles. This lack of understanding affected the players' ability to take in and understand the diegetic instructions given, such as how to access the pause menu.

On a broader scale, when it came to the sound design on the game the players had a few things of note and a lot of ideas when asked. As mentioned earlier they noted how the current choice in music had a problem in its constantly changing volume, making it unnecessarily hard to locate its direction through sound panning. When asked about what improvements they wanted to see in the game, related to the sound for the gameplay, we got the following:

- They wanted a more dynamic range of sounds to make up the music box and help mitigate the problem of changing volume.
- They also wanted more ambient sounds to make the world feel more immersive. This note came in conjunction with the feedback that the lack of visuals takes away from the immersion.
- One idea we were given was to have sounds that showcase how the world is reacting to player movements, as well as adding more sound cues such as sounds that indicate hazards to avoid or safe places.
- Another idea they came with was for faking sound diffraction. The repeated idea they mentioned was to split the level area into sections and each section has its active sound source in the correct place according to where the diffracted sound should appear.
- A smaller note they had was that they wanted cleaner and clearer sound panning to help locate the music.
- Here they also wanted more consistent volumes between voice lines.

Other notes that they had in general concerning what they would want added to the game include objects that are only waist high as to give more life to the environment.

On a different note, the overall opinion of the current gameplay was less than stellar.

To start with they thought the idea of audio focused gameplay was exciting and that it was not a well-developed area in VR. But for this specific game they thought the gameplay was repetitive, something which was partially affected by the music being repetitive. The gameplay also felt tutorial-like throughout all the three levels tested. They thought the game lacked a hook and was fairly one-dimensional story-wise. Outside of this they also felt that the difficulty level did not increase with the levels and instead remained the same throughout. This was because they felt that the positioning of the player and their navigation was what gave the game its difficulty, not the layout of the levels themselves.

When it came to asking about other types of audio focused gameplay they wanted to see, both for this game as well as outside of the framework for *Forest Mole*, we got a lot of ideas. Both for building upon the current iteration of the game as well as completely unrelated ideas that would best suit in a different type of game. They are summarised as the following:

- Puzzles containing several different sound sources, with a problem that they already noticed being that it could confuse the player.
- The idea that the player has to use sound to distract another character to move forward.
- Doors opening and closing, leading to sounds appearing as if they are turning on and off.
- Moving sound sources.
- Adding different sound sources with different tones that together harmonise into an ambience that helps lessen the feeling of one-dimensionality.
- More complex terrain, such as elevation and the ability to climb over or around obstacles.
- Multiplayer gameplay, e.g., PVP, slow-paced, tag-system.

5.3.6 Third SWOT Analysis Iteration

During the second development cycle and its evaluations, even more aspects for the SWOT analysis were identified. These were found while continuing to develop and test the game, consequently containing specific ideas about things such as voice acting and player experience.

Strengths

- Less likely to get VR-related motion sickness because of a lack of visual input.
- Smooth rotations without feeling dizzy.

Weaknesses

- Requires good quality spatial audio to make the player feel in control and aware of their spatial positioning in game.

- Clarity and emotion trade-off for voice acting.
- Room-scale VR can affect spawn point placement, which can be difficult to notice if you cannot see. This might break the game.

Opportunities

- Audio games may feel like a new type of gameplay compared to traditional video games.
- Opportunity for the development of innovative gameplay.
- Games without visuals do not require good quality graphics, which makes these games playable with “worse” hardware, and therefore more inclusive.
- Job opportunities for voice actors, since audio games require strong voice acting performances.

Threats

- Making an audio game often lead to writing a story about blind people. This has the potential to be discriminatory if the writers are not careful.

6

Results

The results of this thesis are given in two parts to give an easier overview of what has been learnt throughout this paper and why the results are what they are. The hope is to give a transparent view of the outcome which would allow the readers the best possible ability to understand the still early iteration of the game *Forest Mole* and how it could have had an effect on the focus of the results.

6.1 Forest Mole

The result learnt from the game is split into two parts. One is the focus on the results given from the evaluations of the game and what they mean when put into context. The other part focuses on describing the current iteration of the game to give a better understanding of potential shortcomings or blind spots in the result.

6.1.1 Result of Evaluation

The latest iteration of *Forest Mole* was created using the *Godot XR Tools* library and the *Raytraced Audio* add-on. The add-on was supposed to deal with diffraction, as given by the approved explanatory videos liked in the documentation. However, the add-on does not have adequate diffraction for building an audio game that relies on diffracted audio to navigate.

This knowledge was gained from the playtests which clearly showcase how the players do not get correctly diffracted audio to follow, leading to a deeper understanding of the level of quality needed for navigational audio in audio games. The same lack of precision was noticed in the panning used, which in turn highlighted the importance of using sound bites which are level in volume throughout the whole clip.

Similarly, *Forest Mole* also showcased the importance of filling the area with different sounds to create an ambience and induce a higher level of immersion. This is because the lack of visual input puts a so much larger weight on using audio to visualise the world.

In the same way it is also more important to give auditory or haptic feedback to the player for every little thing, especially when an action does nothing. This was noticed when players got stuck in place because they still thought an action, such as walking, was taking place normally.

In turn, this showcases how important it is to communicate clearly and directly to the player in a way that they can understand. This goes for everything from what each reaction is meant to indicate to what actions can be done. The player cannot get the usual visual feedback to help indicate what the feedback is responding to or why, putting even more pressure on making sure that the correct information is being delivered.

These observations further connect back to the audio and voice lines used in an audio game. To have a functional audio game the quality of the spatial audio is very important. To have an engaging and interesting audio game experience, the audio needs to be brought to the next level to induce immersion as well as having interesting and challenging audio puzzles and/or exercises.

In a video game it is important to have good spatial audio to make the game feel polished but in an audio game the precision and quality of the audio needs to be much higher.

6.1.2 Latest Iteration

The latest iteration of *Forest Mole* as of this thesis is a prototype of the game which has just had its first major playtest. It is missing many of the finer details that comes with time during the development.

The game has two menus. The main menu and the pause menu. The main menu is accessed through starting the game or through the pause menu. It contains the following menu items: “Start Game”, “Select Level”, “List Controls”, “Enable Debug Visuals”, “Credits”, and “Exit Game”. “Select Level” further contains the ability to select each individual level. The pause menu, accessed through raising a hand above the players head during the gameplay, contains: “Resume Game”, “Restart Level”, “Repeat Instructions”, “List Controls”, “Enable Debug Visuals”, and “Main Menu”.

The game itself consists of eleven levels. The tutorial, nine normal levels in the test-chamber style, and one last ending level which ends in a choice for the player and two different endings depending on said choice.

The ending level is not complete and neither is the Robot which is intended to be introduced halfway through the game. With the discovery of the *Raytraced Audio* add-on and the trust in its description, much of the development was focused on the completion of a first, testable, iteration of the game instead of creating the backend code for a spatial audio simulation that could then be used for the game.

The game itself contains walls of differing thickness and shape; speakers for the voice line audio from the Handler; the basic functionality which was added to the MVP; actual lore and a full story line for the game.

Much of which was overlooked during the playtest because of the failure of the spatial audio.

6.2 SWOT Analysis

During our three iterations of the SWOT analysis, we identified a plethora of relevant aspects to consider when designing an audio-based VR game. The points mentioned here are focused on giving readers ideas of what to think about. They are not to be taken as explicit suggestions or advice, but rather things one can think about or take into account when continuing to research the area of audio-based VR games and the audio needed for them.

This section divides these aspects into categories to allow for more complex answers without having to repeat similar information multiple times.

6.2.1 Audio Games

Some of the identified weaknesses stem directly from the absence of visuals in audio-based games. In a completely audio-based game, it is essential that players understand the spoken content. Therefore, such games should provide the option to select a language, although designing and implementing this feature may prove challenging.

In games with visual interfaces, languages are often presented using their native names (e.g. "English", "Deutsch", "Svenska") and/or flags, see Figure 6.1. Common implementations include displaying all available languages simultaneously or providing a drop-down menu. Many games also prompt players to select a language when the game is first launched by presenting a list of available options. In audio games, however, players may struggle to recognise that a language selection is required, understand how to navigate the menu, or determine how to confirm their choice.

Figure 6.1: Language selection from Among Us[60]

Another weakness identified in audio games is that the absence of visual input may make the narrative more difficult to convey. Cues such as body language and facial expressions must instead be communicated entirely through voice acting. This created a trade-off between emotional expression and clarity in the game's voice lines, highlighting the importance of both the voice line delivery and the manuscript writing. Since all information is conveyed through audio, it is essential that relevant details, such as instructions and narrative elements, are communicated clearly to the player while still allowing the voice acting to deliver the emotions and personalities of the individual characters.

When writing a story for a game, writers will likely take the gameplay into account in order to create a cohesive experience between the narrative and the mechanics. Examples of this include shooter games often focusing on war and detective games revolving around solving crimes or murder cases. Similarly, games with audio-based gameplay may encourage writers to create stories in which the protagonist is blind, as was the case in the game created for this thesis.

However, this may also raise concerns regarding representation, as such portrayals could unintentionally become discriminatory or reinforce stereotypes about visually impaired individuals. Writers therefore have to be careful and considerate, especially if they themselves do not have lived experience of visual impairment.

Summary

- W: It could be difficult to change language.
- W: Clarity and emotion trade-off for voice acting.
- W: The absence of visual input may make the narrative and emotions more difficult to convey.
- S: Audio games can easily focus on stories about a blind protagonist
- W: Portrayals of blind could unintentionally become discriminatory or reinforce stereotypes.

6.2.2 VR

Early in the development process, continuous movement and smooth rotations in VR was believed to increase the risk of motion sickness and was therefore considered a weakness of VR. This view is reinforced by games such as *Half-Life: Alyx*, which recommends teleportation over continuous locomotion.

While this concern is valid, motion sickness in VR is generally caused by a mismatch between visual input and physical sensation. Based on this, it was hypothesised that removing visual stimuli could mitigate the issue. This hypothesis turned out to be true, at least for *Forest Mole* and its playtesting sessions, with no mention of motion sickness whatsoever. It was therefore concluded that when comparing audio-based VR games to VR games with visuals, the audio-based ones have a strength in being able to use continuous movement without causing motion sickness. The same observation was made for smooth rotation.

Another concern regarding audio-based VR games is directly related to the nature of VR itself, specifically room-scale VR. Since the player's in-game position mirrors their physical position within the playspace, it is possible for players to move out of bounds or unintentionally end up outside the intended map area. During development, it was also observed that if the player was not positioned near the centre of the playspace, they could sometimes spawn outside of the map. While this is problematic for VR games in general, it is arguably even more severe in audio games, where players may have no visual indication that anything has gone wrong.

There are of course ways to mitigate this issue, such as automatically teleporting the player back inside the playable area if they leave it. However, it still represents a weakness of VR when compared to non-VR games, particularly in the context of audio-based VR games.

It may be difficult to know which controller is right and left, especially if they are shaped the same, for example like the *HTC VIVE* controllers. In VR games where each controller serves a different purpose, they often look different within the game, such as showing a model of the respective hand or having different colours. This is more difficult to do in a game without visuals.

Summary

S: No motion sickness from continuos movement or smooth rotations.

W: Unintentionally moving out-of-bounds in-game from physically moving through the play-space.

W: Audio games lack visual feedback for when bugs occur.

W: Left-right interchangeable controllers can be mixed up from a lack of the visual indicators given in-game.

6.2.3 Accessibility

One of the most apparent strengths of audio games in general, when compared to video games, is that they can be played by people who are visually impaired. This provides a strong argument for both the existence and continued development of audio games.

In contrast, the strong reliance on sound also means that audio games may be less accessible to players who are hard of hearing. The playtests indicated that even participants with normal hearing experienced some difficulty perceiving left-right panning. As a result, players with impaired hearing in one or both ears may experience even greater challenges when interacting with games that rely heavily on spatial audio.

VR games often utilise the room-scale aspect to allow players to move around more freely. It can also enable interactions such as crouching to gain a clearer point of view when objects obstruct the player's vision. However, these mechanics may reduce accessibility for players who have limited mobility or difficulty performing physical movements.

In the case of *Forest Mole*, the game is fully playable while seated. Although players are encouraged to rotate physically for greater spatial accuracy, the option to rotate using controller input makes the game more inclusive and accessible. Additionally, the absence of visual elements removes the need for players to adjust their elevation or physical position in order to obtain a better viewpoint.

Another aspect of game accessibility concerns the hardware available to the player. In the case of audio-based VR games, two important requirements are VR equipment and an audio output device, preferably headphones. VR also introduces additional physical requirements, as players need sufficient space to safely use room-scale functionality.

To achieve the best possible positional accuracy and spatial awareness, player may also require high quality headphones, which can be expensive. Together, these factors may make audio-based VR games less accessible to individuals who are unwilling or unable to invest in the necessary equipment and physical space.

Because audio-based games do not depend on advanced graphics, they require less computational power and can run on a wider range of hardware. This makes them more accessible to players who do not own high-end computers or graphics cards. Although many graphically intensive games offer lower graphics settings, reducing the graphical quality can negatively affect the intended experience.

However, highly advanced audio simulations that use techniques such as ray tracing or diffraction may also require significant computational power to process real-time audio updates. As a result, games with highly realistic audio could face similar accessibility issues as graphically demanding games, potentially limiting access for players with less powerful hardware.

At present, it is uncertain how the computational requirements of realistic audio compare to those of realistic graphics. It is therefore difficult to determine whether this represents an opportunity or a threat for audio-based games. What is likely, however, is that games combining highly realistic audio with highly realistic graphics would place considerable demands on hardware, requiring more powerful systems to run smoothly.

Summary

S: Inclusive for visually impaired players.

W: Less inclusive for people who are hard of hearing.

W: VR may reduce accessibility for people with limited mobility.

W: VR games require a VR headset.

W: More complex audio may require higher quality output audio devices.

W: Players need sufficient space to safely use room-scale VR functionality.

S: The lack of advanced graphics used means it does not require advanced graphics hardware.

T: Future highly advanced audio simulations may require advanced audio hardware.

O: Future highly advanced audio simulations may not require advanced audio hardware.

6.2.4 Miscellaneous

The identified opportunities and threats concern the broader consequences a fully functional spatial audio simulation could have. These aspects could have effects extending beyond individual games and influence the wider game industry.

One major opportunity presented by this and similar projects is that they draw attention to the existence and potential of audio games. Since most modern games are mainly use visuals, many people may not realise that games can also be experienced through sound alone, without relying on a screen as the primary medium.

By making audio games more engaging, they may also help challenge the common perception that audio games are exclusively intended for visually impaired players. In addition, they could encourage developers who traditionally focus on video games to contribute to the audio game industry and explore new forms of gameplay design.

Another opportunity relates to the previously mentioned weakness of audio games requiring strong vocal performances. Developers of audio games may need to hire more people to record voice lines, potentially creating additional job opportunities for voice actors.

Furthermore, it is not only the voice acting that must meet a high standard, but also other audio elements such as sound effects and ambient audio. These elements must be carefully balanced and adjusted to ensure that players receive important information and atmosphere without becoming overwhelmed. As a result, this could also create additional job opportunities for audio designers and sound engineers.

A potential downside, however, is that employing more people during the development process may increase development cost. This could in turn lead to games becoming more expensive, potentially making them less accessible to consumers who are unwilling or unable to pay higher prices.

Another aspect that extends beyond the game industry is the opportunity to practise navigation based primarily on sound, not only within virtual environments but also in real-world situations. For example, a crew member on a ship could use an audio simulation to practise navigating without vision in the event of an emergency, such as a power outage causing all lights to fail. In such a situation, they may need to locate an object or another person among various surrounding noises while unable to rely on sight.

Summary

O: Increased awareness about the existence and potential of audio games.

O: Help challenge the perception that audio games are only for the visually impaired.

O: Game developers may be encouraged to explore new forms of gameplay design, focused around audio.

O: New job openings and opportunities in the audio sector of game development.

T: The employment of more people for the development process will increase the cost of making a game.

T: Games may become more expensive for consumers.

O: The audio developed for games can be utilised in other fields.

7

Discussion

The discussion will focus on, and around, what we discovered and found relevant for this paper while creating *Forest Mole*. This includes not only a reflection of our game but also a reflection on audio-based VR games as a whole and what we have learnt about new possibilities of gameplay.

7.1 Process Discussion

We are satisfied with the way we worked and split up our work, both in sections and between us. In a general sense everything went according to our time-plan. Throughout our process we made three different timelines, each one more detailed than the last, as we understood more about this project. We were able to keep to and follow our timeline almost perfectly. Something that helped with this is how we set clear goals for ourselves throughout the project.

From the start we knew that we wanted to split our process into three different stages, the prestudy, the game development phase, and the finishing phase.

7.1.1 Prestudy

During the prestudy we focused on learning as much as we could about our subject. This let us easily look through and gather sources for the report. The prestudy phase lasted until the planning report was finished, as intended.

While reading we made sure to save all the relevant sources in a shared document. This way we could dig deeper into different aspects of spatial audio and then share the knowledge between us. The idea for a shared document also helped encourage us to keep finding additional sources, something which helped us read one more source after we were exhausted.

What we found here included previous attempts at making audio games as well as noted areas that had to be properly investigated. We found a few sources that have attempted to make a spatial audio simulation. These were mostly focused on navigation through audio and using the medium of a digital computer to remove outside variables in their experiments. One of the reports, *Efficient 2D Sound Propagation in Video Games*, actually focuses on how sound moves throughout a space instead of the sound propagation being used as a means to another end.

Even then the research space for exploring spatial audio combined with VR was a fairly unexplored area. We came to this conclusion because of the lack of research papers found when looking for sources. This made our idea feel novel and pushed us further along this specific path. Which also influenced our focus during the prestudy. Not in the sense of what types of sources we were looking for, but we did keep an eye out for useful techniques or guidelines that could help us during the development phase.

During this phase we also set aside a piece of time each week for ideation. This time could be anything from a few minutes to a few hours but it allowed us to start the ideation process and cut down on the time needed for the development by a lot. Since we had already talked about what type of game we wanted to develop it allowed us to seamlessly switch over to implementation when the development phase started.

7.1.2 Development

This was the largest section of our timeline by far and split into its own little sprints using the Scrum framework. At this point we had already worked iteratively through the ideation phase, letting us get a feel for it. This way we had a starting point for how much structure we would need for our continued iterative process.

We made sure to write our tasks using the “user story” format. Something that might have not been as needed since we were only two developers and worked on a smaller project overall. On the other hand, we never had to question what any of the cards meant or when they would be considered finished. Weighing the comparatively small time it takes to write a user story, compared to the potential problems that can occur down the line from miscommunication, we are satisfied with our choice. Knowing how to write useful user stories is also a skill that will only help us in the future.

During the development process we mostly drifted into two different directions according to our strengths. This allowed us to continuously develop the game’s storyline in tandem with its technical development. While this is not something that we would encourage because it locks the game onto a predetermined path since changing the story is easier than reprogramming the technical side. We were lucky, both in the way that we had done most of the ideation for the game and the gameplay during the prestudy phase, but also because we already knew that the story would come second to the gameplay. This made it so that we could almost develop the game’s story independently from the game’s technical development for a few weeks. While also having beneficial influences between the two parallel tasks, in the form of ideas from the manuscript and knowledge of what could be implemented from the MVP.

To make sure that we kept consistent contact we had meetings twice a week, most of them in-person. Outside of this we also had regular check-ups where we just talked about further ideas and wants. Meaning that we had, more or less, constant contact between ourselves where we kept the other up to date on what we had done. As well as allowing us to throw around ideas before they were implemented. This constant contact and ability to talk to each other definitely helped us resolve issues before they could grow into larger problems.

While this was going on we also spent hours finding and recording different sounds for our project. While it could have been much easier to simply find “free to use” sounds online for our game, the fact that we recorded everything ourselves does give a larger feel of accomplishment. It also allowed us to not have to worry about copyright restrictions.

7.1.3 Level Design

The level design, as stated previously in *Process*, was heavily effected by our task, to explore how spatial audio can be used to help the player map and navigate the room. The hope was to test different room configurations and see how well the player could navigate. The two more interesting parts to briefly mention here is level 2 and level 5.

Level 2 got the most interesting result out of the levels we tested. It was the reason for us learning that the *Raytraced audio* add-on did not work as we expected. In hindsight, it would have been more appropriate for the level 2 player difficulty if the level was a simple U shape instead of an S shape. We are glad that we did not make that change though because we got our useful information from the fact that level 2 had more corners and extra complexity.

Level 5 was originally meant to highlight how sound is effected by the player standing close to a wall. Since this functionality was not implemented in the *Raytraced Audio* add-on and we did not have the time necessary to develop it ourselves, we had to scrap that idea. The idea itself was lovingly called “Shrek ears” during development and there is more written about this in *Section 7.6 Future Work*.

Much of this functionality never got the chance to be tested. Not only was the sound effects for the later levels not fully integrated, but the spatial audio did not even work as intended. It was not until almost the end of the development phase that we even realised that the *Raytraced Audio* add-on did not deliver on what we were told. As mentioned above, this problem was found out at level 2, which is why it was good that we stress tested the audio with that many corners.

7.1.4 Raytraced Audio Add-on

The *Raytraced Audio* add-on had many smaller problems that were all related to each other. Not only did it indiscriminately muffle sounds if they did not have a clear path to the player, uncaring if the sound was just around the corner. To clarify, the muffling effect was added when a sound source is blocked by an obstacle (e.g. a wall). This muffling is created by using a low-pass filter, which decreases the amplitude of all sounds above a certain frequency threshold. The threshold is decided by a ratio of how many rays reach the sound versus how many rays are emitted from the player. This ratio is applied linearly to logarithmically transformed frequencies to get a frequency from 250 - 20 000 Hz (by default).

```
var ratio: float = float(_lowpass_rays_count) / float(rays_count)
var lowpass: AudioEffectLowPassFilter = AudioServer.get_bus_effect(idx, 0)
var log_t: float = lerpf(LOG_MIN_HZ, LOG_MAX_HZ, ratio)
var log_hz: float = log(lowpass.cutoff_hz) / LOG2
log_hz = lerpf(log_hz, log_t, interpolation)
lowpass.cutoff_hz = pow(2, log_hz)
```

If there is a wall between the player and the sound source, the ratio becomes 0, resulting in a threshold of 250 Hz. Since the music created for this game consists of notes with a higher frequency than 250 Hz, the entire musical piece gets affected by the low-pass filter, rendering it almost inaudible.

To fix this issue, the minimum frequency for the low-pass filter was changed from 250 Hz to 2 500 Hz, which kept some of the muffling effect while still allowing the lower frequencies of the music to reach the player.

The reason for why the issue with the add-on was discovered so late was that it seemed to work during testing of the MVP. The hypothesis is that the shape of the MVP level along with the music-box's placeholder music made it a lot more difficult to discover the issue. The placeholder music contained a wider range of frequencies, especially lower frequencies, when compared to the final music. This allowed it to still be heard when it became muffled because of walls, as intended. The shape of the MVP level, specifically the fact that it was a long corridor with the player-spawn in one end and music box in the other, also made it so the music came from the correct direction. This, along with the fact that the playtesters successfully completed the level, falsely led us to the conclusion that the add-on's functionality would be sufficient for the project.

Furthermore, when we tested the finalised levels and music during development, our familiarity with each level's layout appeared to have subconsciously aided our navigation, allowing us to locate the music box with ease each time, reinforcing the belief that the sound was functioning as intended. This issue highlights the importance of conducting playtests – even if just informal ones – with unacquainted participants, and not only while using placeholder sounds and levels.

7.1.5 Godot Steam Audio Add-on

It was not until after the latest evaluation that we had a run-in with another game developer who recommended that we check out the *Godot Steam Audio* add-on. Even though the time we set aside for working on *Forest Mole* is officially over, we wanted to know if we were too hasty at the start of our project and if we could have functional spatial audio if we had just looked a bit further.

Godot Steam Audio is an open-source extension to Godot that intends to implement *Steam Audio*[20], a free plugin currently available for game engines such as Unity and Unreal Engine, but not for Godot. *Steam Audio* uses for example reverb and HRTF effects and claims to "significantly improves immersion in VR"[20]. Of note is that the *Godot Steam Audio* plugin is not complete. Below is a quote from the creator of the plugin:

> "This extension has been created and maintained by me (@stechyo), but due to a lack of time/interest in game development in the past year this is not really being maintained/developed at the rate it could be."
>
> — @stechyo (2025-03-05)
> GitHub, https://github.com/stechyo/godot-steam-audio, accessed 2026-05-16.

With a fairly quick read through of the GitHub repository for *Godot Steam Audio* and a look at their demo video we can say that it would not have made a difference. *Godot Steam Audio* focuses on occlusion and reverb just as the *Raytraced Audio* add-on through its muffle and echo abilities. None of its features state that *Godot Steam Audio* is meant to contain anything close to the ability to have sound bend around corners. We have not had the time to actually test this add-on and therefore cannot be completely sure, but with the lack of diffraction having been mentioned in its current or future features it does not seem likely. When looking over the demo video it becomes more and more clear that *Godot Steam Audio* does not have any functionality which allows sound to diffract around corners.

7.1.6 Report

The last phase we had was the finalisation phase, which primarily focused on completing the report. This included documenting the results of both the project itself and the evaluations that were conducted. These findings were then analysed in relation to our research question of what is important to consider when designing an audio-based VR game.

7.1.7 AI

When writing and debugging the game's code, a large language model (LLM) was used as a tool to aid in and speed up the development process. It allowed us to spend less time ironing out kinks, and instead focus on what we needed to implement in order to make the game work as intended. The LLM used was ChatGPT[61]. Throughout the project (January-June 2026), different versions of ChatGPT may have been used as the service evolved; we relied on the version available for free on the ChatGPT website at the time of use.

The LLM was useful for identifying solutions to small implementation problems, such as modifying action a certain button press would do. Additionally, it was particularly effective at generating and restructuring larger code segments involving repetitive code. One example of this was the creation of multiple similar objects in Godot. Manually duplicating these objects can be time consuming, as copied instances may require additional modifications to ensure unique identifiers and correct functionality. By generating the required code through an LLM, this process became a lot quicker.

When debugging, we prompted the LLM with a section of the code, along with an explanation of the intended behaviour and the current behaviour. This helped narrow down the possible causes of the issue. Although the LLM suggested potential solutions, these rarely worked without modification. This was either because the suggestions were incorrect or incomplete, in which case we could prompt it again with additional context, or because the proposed solutions did not integrate well with the rest of the code due to the LLM lacking access to the full project context. Furthermore, providing too much context often led to confusion or cause the LLM to overlook important aspects of the prompt in favour of less relevant details.

When it comes to the creation of our game's assets and manuscript, we did not use AI. While it could have made the development quicker, or enabled us to spend more time on other parts of the product, we wanted all creative elements to be made by ourselves. Firstly, we believe it makes for a more genuine experience when it is curated by the designers, which includes everything the player hears when playing the game. Secondly, to us personally, it gives us a stronger sense of accomplishment when we can feel like we did everything ourselves.

Another use case for the LLM was when writing this thesis. Here it was used to check for spelling and grammatical errors and to help enhance the flow of the text. All ideas, opinions and arguments are our own, however, and not generated by AI. This includes, but is not limited to, our research question, the aspects identified for the SWOT analysis, and our ideas for future work in the field of audio-based VR games.

7.2 Forest Mole Discussion

Creating *Forest Mole* was a valuable experience. Because it had no visuals, we had to truly focus in the audio as compared to games like *Scanner Sombre* and *Perception* which do use visuals to help show the environment. It felt like it had more freedom of movement than *A Blind Legend* where the player moves short distances between checkpoints. We took heed of what was learnt in Falk's study where an *Audio Game Prototype* was made and tested, which influenced us in our choice of not utilising elevation for finding the goal in *Forest Mole*.

Return of the Obra Dinn while being mainly a sighted game did help inspire us about ambient sounds and how a lack of visuals helps make the player focus on what they are hearing. *Portal* was a clear inspiration, both when it comes to how the Handler mimics *GLaDOS* in functionality but also in the idea of barren test chambers as the setting where the player solves puzzles, even if ours are significantly simpler. In *Half-Life: Alyx* we got a feel for how different movement types in VR affect the player. It was fun to play around and learn first hand just how motion sick one could get from using the continuos movement function. This knowledge helped emphasise the contrast of no motion sickness in *Forest Mole* because of its lack of visuals.

We were positively surprised at how far we got into realising the idea of *Forest Mole*. With all its levels being playable as well as a complete storyline with all its recorded voice lines. The thing that held it back from feeling far more finished than it did at this stage was the blunder with the *Raytraced Audio* add-on. We were disappointed when we realised that this add-on did not hold up to our expectations, despite testing it during the MVP. Since this problem only became known to us during the last week, we did not have enough time to fix it or find another plugin that would work better.

Another thing that we could have done better was to second guess our choice of controls. This is connected to another thing which we did not learn until much later, and which in hindsight feels obvious. One of our fairly early implemented mechanics is the ability to “clap”. This was meant as a way for the player to make a sound and either listen to the reverb and get a feel for the room size, or to later use it to make noise to call over the Robot. Clapping is an action that can easily become problematic in VR and that becomes obvious when remembering that the player cannot actually see the real world. This combined with attempting to slam their hands together when they cannot see, will end up with broken controllers for at least some players. We could have easily switched out this action to something one handed, such as pushing a button on the controller to whistle to snap their fingers. This has its own problems, especially when it comes to how to give the instruction in the same diegetic format that we tried to keep throughout the gameplay. It can also be seen as something that breaks immersion, either because the player cannot to that action in real life or because the action no longer requires a physical movement to trigger. The more high pitched noise of a whistle or snap is also not as useful for checking a room’s reverb as the deep undertones of a hollow clap.

For this paper we managed to create an audio game that could be tested. It also gave us valuable input for our SWOT analysis, both during its development as well as during its evaluation. This was in part what we wanted to accomplish with going through the process of creating an audio game. We still did expect to have a functional spatial audio system to test at the end of our development process. The hope was to get feedback pertaining to the more detailed part of spatial audio functionality. Since the spatial audio we used did not work, we instead got feedback that was expected for an earlier prototype in the development chain. This feedback will still be useful but it might also mean that our resulting SWOT analysis focuses more on the basic concepts of spatial audio than it would have done otherwise.

7.3 SWOT Analysis Discussion

This section focuses on discussing the result of the SWOT analysis. It is split into separate parts to emulate its counter part in the result chapter. The hope is to make it easier for the reader to follow along and understand what aspects are discussed where.

7.3.1 Audio Games

Audio games usually have a lack of visuals. If they are not completely absent they are at least not the main way for the player to gain information from the game. While that is the point, it also means that much of the input that can be gained through visuals instead has to be transmitted through sound.

Therefore it becomes essential for the player to understand the spoken content as well as get enough usable feedback from the world around them to make up for their lack of visuals. This does not mean that everything has to be conveyed, for example the colour of the walls or items is useless to convey unless it is relevant to the gameplay.

Either way it makes it harder to convey a narrative when emotion can only be expressed through sound. Without visuals the player cannot use facial expressions or body language to help indicate other characters' emotions. Since all of this have to be conveyed through the sound it also puts much more pressure on the voice actors to get the voice lines right. Something which we noticed during our own tests.

The person who recorded the voice lines for the menu was chosen because of their ability to speak clearly and with a distinct accent. This provided a further indication that the menu is outside the scope of the gameplay space (non-diegetic). Even with their ability to speak clearly and the first voice line in the game both welcoming the player and explaining how to navigate the menu, many of the playtesters mentioned how they did not know that they could navigate the menu in the first place. We have three theories as to why this issue occurred. The first is that the problem could stem from the fact that the game's lack of polish may have led participants to assume that the menu was not operational, despite the voice lines existing. The second is that the participants may not have listened carefully, as they were primarily focused on completing the playtest as instructed. The last theory is that the voice lines themselves were not clear enough for the players to internalise the information being conveyed.

Another thing that can disrupt the player's ability to take in the information is a possible language barrier. While this was not a factor in our tests, it is a factor that becomes important when considering a larger user base.

In a regular video game there is usually a setting to change the current language of the game. Depending on the game, some only have a choice in the menu while other games have the option to choose before the game even fully opens. In either case the action of choosing a language is very visually focused with having to find where to change it and then locate the correct language in the menu.

We mentioned in passing a few different ideas but the three that were actually thought over are stated below. The first and simplest is to just use the same style as a video game with the difference being that all the options are said out loud. While this would work, it still has the problem that the player needs to be passingly familiar with the starting language to be able to navigate to the settings menu.

A different idea is to have a language prompt the first time the game is opened. Our symbolic idea here is to have the game prompt the player into an action using different languages each time. This idea comes with the caveat that it is expected of the player to not move around or push buttons while waiting for a language they know to pop up. Something which can easily end up with players stuck with a language they do not know because they thought to try and skip the intro for the game.

The third idea is to find a way for the audio game to locate the set language for the system it is running on. This would hopefully solve the potential problem of the player not even being passably familiar with the language. But it is still not a guarantee, there is never a 100% guarantee that the player will know the language set, and while a combination of all three of these ideas would help minimise the risk it could also make the introduction to the game appear taxing or annoying.

7.3.2 VR

It was a good idea to use VR for this project, not only were we able to make use of the player's movements in real life through the head tracking it also helped focus the player on the digital world. The head tracking from VR made players able to rotate more precisely than what a controller or a mouse would allow, giving them more control and helping them get a feel for the space. It also placed them inside the game space in a way that is not the same as sitting in front of a computer. Since the visual input form the VR was just a black screen it removed visual stimuli and helped the players focus on what they heard.

As mentioned in the *Result*, an audio game is able to use continuous movement and rotation in a way that a VR game with visuals cannot. VR games usually stick to in-game movements where the visuals either fade in and out or it just snaps the player's position to the new coordinates. This is because of the disorientation that happens when the in-game visuals do not match the physical movement. An example of the confusion is how we naturally lean forward when walking, something which can lead to people feeling like they are falling because they are either leaning into or away from the in-game movement. A more well known example is how the in-game visuals try to move around the player as they walk, which is the thing that the motion sickness is usually attributed to.

From this we hypothesised that removing visual stimuli could mitigate the issue, a hypothesis that was supported by our playtesting results. None of the participants had any mention of motion sickness or other negative effects from spending half an hour in VR. Even with an audio clip of footsteps that did not match their walking speed, participants did not experience motion sickness. Instead, they reported frustration caused by being unable to accurately judge their walking speed at any given moment. Addressing this issue would likely improve the player's spatial awareness and understanding of their position within the virtual environment.

While we did learn that motion sickness is not something that we have to worry about in the current iteration of our game, that still does not mean that our hypothesis has to be correct. We have not tested if it will come back if we add multiple sound sources that all come from different directions. If this does happen then a new hypothesis can be made concerning either the delay in feedback compared to the action taken or simply that the virtual world is not imitating the real world well enough to fool all our senses properly, causing the motion sickness. Either way it is something we are unable to test currently.

When it comes to bugs in VR, the issue we encountered most often was falling through the level. We are fairly certain that the main reason for this was that the character collided with the environment as the next level was loading. Fixing this did lessen the amount of times the player fell through the floor when a new level spawned, but the problem did still occur occasionally. Therefore, we also made sure that if the player dropped below a given y-value they would be teleported back to the spawn point.

This did highlight the problem of the player not noticing when a problem has occurred. The bug was invisible on the player's side and was only noticed because of clipping planes and the spectator camera. In a video game, players will usually notice if they get stuck on the environment or flung into the sky because they are able to see it. With audio games like *Forest Mole* that is not an option, putting greater weight on making sure that the audio works as intended. If the room itself has many small background noises, then the player will hear if they get suddenly flung away from the map. While this is only one example it does give a good overview of the problem, as well as the solution we would want to implement. This helped show us that it is not just important to solve bugs but also to build the game in a way that helps indicate to the player when a bug has occurred.

7.3.3 Accessibility

When it comes to the accessibility of an audio game the most obvious one is the result of the reliance on audio. Meaning it is inclusive to people who are visually impaired while being more elusive to those that are hard of hearing. This is a fact of the genre and while audio can be visualised to help include this group, it does cause the risk of the game no longer belonging to the category of audio game.

Another way is to overstate the natural differences in spatial audio, which in this context mainly refers to over-exaggerating the audio panning. Being able to hear which direction the audio came from was a hard task even for our playtesters with normal hearing. If this was a result of our players being unused to navigating through audio or another thing that the *Raytraced Audio* add-on did badly is debatable. On one side, we as the developers could clearly hear where the music was located, and all the players did eventually find the music sources. On the other, many of them complained specifically about the inaccuracy of the audio panning. Combining these facts we can come to two theories.

The first is that accuracy increases with practice. Both because with time the player starts to understand the trick to using the virtual panning to locate the sound source, but also because they might start to understand and remember the levels. The latter of which is something that happened to the developers.

The other theory is that the *Raytraced Audio* add-on did not have good enough panning. This means that it was not created to have the required precision that we would need for a game like *Forest Mole*, where audio does not just supplement the visuals but is the main tool used for navigation.

The idea of using VR's room-scale to integrate height levels to different obstacles and sound sources is something that can hopefully give the world more life. These could for example be secrets being heard through air ducts close to the floor, bushes that the player can walk through, or simple square obstacles that just reach the collarbone. However, if not integrated properly it can alienate people who might not be in their prime or as able-bodied as the game requires. This can be solved through adding accessibility options in the menu, such as clicking a button to have the player crouch. The same function could be utilised for the current controls to hide the functionality to turn using the controller. If we implemented this, it might have helped indicate to our playtesters that they should actually rotate their body.

When it comes to monetary accessibility we do not have enough information to tell how the market will develop. While it might end up with spatial audio being able to run on any normal pair of headphones, it might also end up taking enough computing power that all computers will require a processing unit similar to the GPU to run the audio. From our one instance of this research we simply do not have enough information to know for sure.

7.3.4 Miscellaneous

This section contains the most about the potential larger consequences of audio games taking off on a larger scale. This means that much of this is the result of speculation.

Concerning audio games themselves, if audio gets to take more space for everyday games it might lead to more people learning about the existence of audio games overall. While players are aware on some level that games that focus on audio exists, it might not be something that they actively think of. Leading to assumptions such as audio games only existing as an alternative to video games for blind people. Video games is such a broad term and so very integrated into our language that it is not surprising if the term *audio games* feels like an inferior alternative to the “normal” video games.

We, who have gotten this far, know that audio games is its own genre and for everyone, with its own opportunities for novel gameplay. If spatial audio becomes more normalised in games it might also help elevate the larger audio games genre into something that feels more acceptable to play for everyone. This would also hopefully help get rid of the stigma that the term audio games is inferior just because it is less known, and the term video games became accepted first.

If audio games become more mainstream that would also mean an increased need for the parts to make audio games. Jobs such as voice actors, sound effect artists, audio designers, sound engineers, and probably many more working with audio could hopefully see an up-tick in job offers.

Taking a step away from games, if spatial audio itself becomes big then it can be used for so much more. Being able to practise audio-only navigation in a safe and simple way would be fantastic. While something probably already exists for emergency workers such as firefighters, more alternatives that all train different aspects can only help. Even if it is not useful for training firefighters to navigate smoke filled buildings, or workers through factories that lose power, it can still be useful for the general population to learn to use their hearing more.

7.4 Validity and Generalisability

The SWOT analysis was iteratively developed throughout the project, both prior to, during, and after development. New aspects were continuously identified and refined, and later organised into categories, which were then reviewed and discussed. This iterative approach helped ensure that relevant aspects of audio-based VR games were progressively incorporated as they emerged, allowing earlier findings to inform subsequent observations. This process strengthens the validity of the analysis.

However, the relatively small number of participants involved in the evaluation, combined with the limitation of testing exclusively with sighted participants, may have restricted the range of perspectives captured. Furthermore, the inability to test certain parts of the game due to the audio not functioning as intended may have limited the identification of additional aspects. As a result, certain aspects may not have been identified, limiting the completeness of the analysis.

The development and evaluation process used in this project could likely be reproduced by other teams within similar audio-based VR game projects. Since the evaluations were conducted in a controlled and quiet environment, the testing conditions are relatively feasible to replicate in future studies. Applying the same process to a broader range of projects and participant groups could contribute to a more comprehensive understanding of player needs and challenges within audio-focused VR experiences.

Furthermore, although certain spatial audio systems did not function fully as intended during testing, the evaluation was conducted with the intended audio design in mind. As a result, parts of the findings may still be relevant for future audio game projects implementing functional spatial audio systems.

7.5 Ethics

Most aspects regarding ethics when it comes to audio-based VR games have already been discussed as part of the SWOT analysis. This section will therefore mostly focus on ethics regarding the process of developing the game and writing this thesis.

7.5.1 Development

All assets, including sound effects and music, were created by us specifically for this project. During development, however, a commercially released copyrighted song was temporarily used as placeholder music for the music box. This track was only intended for internal testing purposes while the final music was being composed and produced. Before the final evaluation of the project, the placeholder track had been completely replaced with original music created by us, hence no copyrighted material was used for the music.

The ambient forest sound heard at the beginning and end of the game was recorded in a secluded forest area away from other people. This was done to ensure that no individuals were unknowingly recorded, meaning the audio only contained natural environmental sounds such as wind, birds, and the occasional distant aeroplane engine. While the primary motivation for choosing this location was to avoid unwanted sounds such as voices or traffic noise in the recording, it also had the positive ethical outcome of ensuring that no uninformed individuals were captured in the audio.

The manuscript was written by us, but we had help when recording voice lines. Specifically, four friends and family members were kind enough to voice act in our game. They are all mentioned in the acknowledgement of this thesis, and will also be given credit in a potential wider release of *Forest Mole*.

Ethical concerns regarding games and their narratives include how different characters and groups are portrayed. In the game developed for this thesis, the inclusion of a blind protagonist introduces ethical considerations related to disability representation and ableism. Early in the game, one character states: "[...] I suppose that is the best you can expect from someone blind." This line is intended to emphasise the character's antagonistic nature and prejudiced worldview. However, including discriminatory language in a narrative still requires careful consideration, as representations of this kind may reinforce harmful stereotypes or affect players in unintended ways.

7.5.2 Evaluation

When it comes to evaluation, specifically conducting playtests, there are several important considerations to keep in mind. First, participants must be informed that they are partaking in a playtest and understand the extent to which their actions, comments, and behaviours will be recorded or documented. Second, any personal information collected – such as age or gender – should be clearly communicated to participants beforehand, allowing them to decide whether they are comfortable providing that information.

In the case of the playtesting sessions for *Forest Mole*, these considerations were taken into account. Although there was no formal consent form or contract, participants were informed beforehand that they were testing a game as part of a masters thesis project. Before each playtest, participants were told that their actions and comments during the playtest would be observed and noted. At the end of the session, participants were also asked questions regarding their experience, such as what aspects they found difficult and what changes they would like to see implemented.

No personal information was collected during the playtests, primarily because it was not considered relevant to either the game itself or the purpose of the thesis. The evaluation focused on how effectively the audio design functioned and how participants used VR head-tracking to locate sound sources. The intention was not to compare performance across demographic groups, such as differences between age groups in the ability to localise sound. The only personal questions asked concerned whether participants had previous experience with VR and/or audio games.

However, this information was gathered solely to determine how much guidance participants might require before playing, rather than to serve as evaluation data.

One aspect that we did not look into, which could have helped us in the development of the game, is consulting and doing playtests with visually impaired individuals. This limitation in our process could potentially make the game less accessible to these people, since we do not know how they prefer playing games, nor how they use sound to navigate (both physically and in-game). It is likely that we do not notice subtle sounds and changes by the environment the same way that visually impaired people do, and therefore miss or overlook audio cues that we could have added to the game to make it easier to play.

7.6 Future Work

The research into spatial audio is still fairly unexplored, especially concerning VR, meaning there is much more that can be researched. Not only do we need different people doing the same tests to corroborate the answers, we also need future innovation in the genre of spatial audio simulations. There will not be one solution that fits all games and having more people build programs for more realistic in-game audio will increase the use of better spatial audio programs in games, as well as the general draw to research the subject. It will make sure no single solution becomes dominant across all use cases, increasing the need for continuous development and innovation of in-game audio.

To get a more complete picture and more solid evidence of the results of different audio simulations we need more people and more tests. By letting future researchers know where we left off and the problems we had, the hope is to help them not stumble as we did and actually be able to get the more detailed answers about gameplay and user experience which we had to give up on.

When it comes to future development of games that require more accurate sound, it would be advantageous to have existing add-ons or plugins to aid in this. Since the *Raytraced Audio* add-on for Godot did not live up to expectations and the *Godot Steam Audio* add-on did not look any better, the next best plan is to create one that does. Either way this is something that would be needed eventually when testing the more taxing audio situations. With it being a surprise just how far behind virtual audio is compared to its visual counterpart, this has only gotten more important.

What we already know is that for any future audio simulations to function with enough precision for audio game navigation, it needs to include audio diffraction. In *Section 7.6.1 The Future of Forest Mole* we explain more about how we plan to solve this, but our current idea may not necessarily be the best one. There are other ways that diffraction can be implemented. These include both sound maps that are dynamically generated during runtime, and faking diffraction by hardcoding extra sound sources for each map. The first one would be able to function and update without constant alterations to each level, while the hard coded version will most likely load much faster because the game does not have to run the calculations to determine where each source for the diffracted sound should be placed.

Another one of the sound aspects that can be explored refers back to *Chapter 3 Theory*, more specifically *Section 3.3.1 Collision.* A sound in real life almost never sounds exactly the same, be it because the collision was harder, faster, using different materials, or it occurred in a different location on the given objects. All of these are aspects that we unconsciously take heed of when doing an action. Attempting to create a program for simulating this would probably help increase awareness and open up the floor for many more questions about virtual audio down the line.

When it comes to audio games there are differences in what can be done depending on the platform it is made for. A phone game might appear more like a puzzle or an audiobook, while a computer game can utilise the keyboard and mouse. An audio game using the VR platform can have a larger focus on spatial awareness thanks to the more detailed head movements. It is a completely different way to play games. We believe that there is much that can be discovered about and developed for audio games by researching it for VR. We believe that it forces the innovation of game audio to be better than its computer counterparts. This is mainly because of how VR places the player inside the world, and therefore has the perfect setup to allow developers to imitate reality.

The possibilities for new and innovative gameplay in audio games has yet to be explored to its fullest. An easy way to make an audio game is to simply take a relatively simple video game and make it functional through mainly auditory feedback. We believe this path does a disservice to what more audio games could be. What ideas, what possibilities for gameplay can we create that would not work for traditional video games? What unexplored possibilities for innovation are we missing because we feel settled with video games as they are right now?

One way to explore these possibilities is to talk to people with visual impairments. How do their lives differ from sighted individuals? What is normal for them? What games do they already like or what do they do in their free time? Are they used to the auditory cognitive load and how does that affect how difficult they find audio games? We do not know what potential ideas or knowledge can be found through simply consulting people that are a lot more used to utilising their other senses to make up for their lack of vision. In the same way there is probably a lot that can be learnt from having playtests with visually impaired people and watch how they tackle the same problems as the people with vision do. This goes for more than just games. The programs used to simulate spatial audio can be used for more than just games, and it starts with asking those that could gain, or that have the largest insight, into the subject.

7.6.1 The Future of Forest Mole

Our hope is to continue development on this game, with the final end goal being to have a fully functional game at a later date. To accomplish this we will most likely have to create our own spatial audio simulation. To make a fully functional, and fun, audio game we also plan to expand the concept with more puzzles.

The first step, as stated above, is to create a Godot add-on with sound diffraction. To implement diffraction we want to take inspiration from how diffraction was modelled in the paper *Efficient 2D Sound Propagation in Video Games*. This includes how each sound source sends out rays to look for corners where a new sound source should be placed until the whole map has been covered. All the resulting sound sources can then be added to a list where one source is chosen from each tick. The criteria for the choice will be related to which sound source is visible to the player and has the largest amplitude. This chosen source is then the only source to make sound that tick. After we have the diffraction functional we can start looking into adding muffling through walls and sound reflection, both of which will help make the sound feel more real. We will have to make sure though that the realism does not take away from the functionality of the spatial audio tracking.

When we do get the audio functioning it will open up for a much broader range of puzzles we can implement. We can have hazardous terrain that bubble and pop like lava, or enemies that patrol the area. Puzzles can include leading them away using sound or hearing secrets through a low sitting ventilation access. During the testing we did get quite a few ideas from our participants. they ranged from features they wanted to see added to *Forest Mole*, to concepts that they think would work for audio games in general.

One of the functionalities that we want to add is what we call “Shrek ears”. This is the current term we are using because we have yet to find an easy way to summarise the concept. The idea is to emulate how sound is affected when someone stands close to a wall. When standing perpendicular to a wall, usually with one ear less than half a meter from it, some frequencies in the sound starts appearing muted, or at least clearly affected by the proximity. We want to emulate this effect to give the player an instinctual warning when they are standing close to a wall. Our hope is for this to allow players to follow a wall after they have located it without having to constantly use the haptic feedback to feel for it. This idea gained the name “Shrek ears” after one of the ideation sessions ended with the mental image of two rays sticking out of the player’s ears. These rays would be half a meter long and if they hit an obstacle they would mute high-pitched sounds according to the distance from the object to the ear. This way we can hopefully handle situations when the player is not standing completely perpendicular to the wall, as well as give a more smooth transition for if they turn their head while the “Shrek ears” muting is in effect.

When it comes to improvements inside the game itself, we have mainly two points. These include the settings menu and giving the game re-playability. The settings menu is mostly something we want to add so we can let players customise their experience without taking away from the original spirit of the game. Some of the settings we want in here include: subtitles, the debug visuals, and accessibility controls.

The only one of these options that are not immediately obvious from its name is the accessibility controls. Right now we have the left and right controls doing different player movements. This easily lead to confusion during development, which then resulted in clearer instructions during the playtest. At the moment the player has the option to either turn using their feet or using the right joystick, right out of the gate. Our plan is to add different ways for the game to handle input for movement. This way we can have the “Turning inside the game” as an alternative for players that require it.

To make the game feel repayable we want to add a speed-run feature. By this we mean that we want to add a timer to each level which gives the player the time said level took before the Handler starts talking at the end of a level. That way we can also add the ability to skip the Handler’s voice lines. Something that might also be added as a setting by either turning off the game narrative, or turning on the speed-running game-state.

Our current hope is to get far enough that we can hold another playtest and get us one step closer to being able to release *Forest Mole* as a fully-fledged game.

8

Conclusion

This thesis follows the design and implementation of an audio-based VR game. The game was created as part of a process using the *Research Through Design* method, in order to answer the following **Research Question**:

What are important aspects to consider when designing audio games for VR?

The resulting game was *Forest Mole*, an audio game for VR in which the player is supposed to navigate levels through sound. The game is set in a facility where the player does tests overseen by their Handler. These tests consist of the player locating what has been called the "music box" during development. The game was created with the intention of having a narrative and lore. The hope was to make a good enough game that the players could give valuable feedback on the actual gameplay without having to worry about the hiccups that could break their immersion.

Outside of this the game development itself also helped give insight into the matter of what goes into creating an audio game, what is important and what aspects might go overlooked when just playing the game normally.

Based on the knowledge gained during the prestudy and development phases, and the data acquired during evaluation, the research question was answered with a SWOT analysis. The most important aspects of the SWOT analysis are:

Strengths

- Audio games are playable by the visually impaired.
- The player does not get motion sickness from continuous movement and smooth rotations, as they might from VR games with visuals.

Weaknesses

- Audio games are less inclusive to people who are hard of hearing.
- Audio games require a higher quality of both the audio clips and the audio programs used, compared to video games.

Opportunities

- More job-openings in the audio sector of game creation, such as voice actors and sound engineers.
- Increased awareness about the existence of audio games.

Threats

- Games with more accurate audio becoming more computationally expensive, requiring more powerful and expensive hardware.
- Games requiring a larger team of developers in the audio sector, leading to more expensive games.

There is so much more to explore and research about audio games and spatial audio simulation in general. This paper found a lack of spatial audio simulations that were of a quality functional for audio games. Not only were they not mapping diffraction they also simply lacked the precision needed to be the primary source of feedback in a game. The field remains vast and relatively under explored, presenting significant opportunities for future research and development. With increased interest in this area, there is strong potential for advancements that could improve not only audio games, but spatial audio in games overall.

Bibliography

[1] R. Arnott, *Robin Arnott, "Sense Deprivation and Immersion"*, Taken from 23 minutes into the video, Accessed: 2026-06-09, 2011. [Online]. Available: https://vimeo.com/26992553?fl=pl&fe=cm.

[2] J. Nuora, "Introduction to sound design for virtual reality games : a look into 3D sound, spatializer plugins and their implementation in Unity game engine," Accessed: 2026-02-13, M.S. thesis, Tampere University of Applied Sciences, 2018. [Online]. Available: https://urn.fi/URN:NBN:fi:amk-201803213645.

[3] T. G. Project, *Git*, Accessed: 2026-05-08, 2026. [Online]. Available: https://git-scm.com/.

[4] I. GitHub, *Github*, Accessed: 2026-05-08, 2026. [Online]. Available: https://github.com/.

[5] Trello, *Trello*, [Accessed 2026-05-08], 2026. [Online]. Available: https://trello.com/.

[6] O. Shatokhin, A. Dzedzickis, M. Peiulien, and V. Buinskas, "Extended reality: Types and applications," *Applied Sciences*, vol. 15, no. 6, p. 3282, 2025. DOI: 10.3390/app15063282. Accessed: May 22, 2026. [Online]. Available: https://www.mdpi.com/2076-3417/15/6/3282.

[7] Online Etymology Dictionary. "Etymology." Online Etymology Dictionary, Accessed: May 22, 2026. [Online]. Available: https://www.etymonline.com/word/virtual.

[8] Wikipedia contributors, *Virtual reality*. Accessed: May 21, 2026. [Online]. Available: https://en.wikipedia.org/wiki/Virtual_reality.

[9] Wikipedia contributors, *Room-scale*. Accessed: May 21, 2026. [Online]. Available: https://en.wikipedia.org/wiki/Room-scale.

[10] HTC VIVE, *Vive vr, ar, and mr headsets, glasses, experiences*, Accessed: 2026-05-11, 2026. [Online]. Available: https://www.vive.com/us/.

[11] Meta Platforms, Inc., *Meta quest 2*, Accessed: 2026-05-11, 2026. [Online]. Available: https://www.meta.com/quest/.

[12] Epic games, *Unreal engine*, English, Accessed: 2026-02-11. [Online]. Available: https://www.unrealengine.com/en-US.

[13] Unity Technologies, *Unity*, English, Accessed: 2026-02-11. [Online]. Available: https://unity.com/.

[14] Godot Foundation, *Godot*, English, Accessed: 2026-02-11. [Online]. Available: https://godotengine.org/.

[15] GameFromScratch, *Game Engine Popularity in 2024*, Accessed: 2026-05-25, Jan. 2024. [Online]. Available: https://gamefromscratch.com/game-engine-popularity-in-2024/.

[16] Oliver Carson, *"Comparing Popular Game Engines*, Accessed: 2026-02-11, Aug. 2024. [Online]. Available: https://www.pubnub.com/blog/comparing-popular-game-engines/.

[17] GodotVR, *godot-xr-tools*, GitHub repository, accessed 2026-05-13. [Online]. Available: https://github.com/GodotVR/godot-xr-tools.

[18] WhoStoleMyCoffee, *raytraced-audio*, GitHub repository, accessed 2026-05-13. [Online]. Available: https://github.com/WhoStoleMyCoffee/raytraced-audio/.

[19] stechyo, *godot-steam-audio*, GitHub repository, accessed 2026-05-16. [Online]. Available: https://github.com/stechyo/godot-steam-audio.

[20] Valve Corporation, *Steam Audio*, Accessed: 2026-05-16, 2025. [Online]. Available: https://valvesoftware.github.io/steam-audio/.

[21] Wikipedia, *Audio Game*, Accessed: 2026-06-09, 2026. [Online]. Available: https://en.wikipedia.org/wiki/Audio_game.

[22] The Deep End Games, *Perception*, PC video game (Steam), 2017. Accessed: Feb. 10, 2026. [Online]. Available: https://store.steampowered.com/app/426310/Perception/.

[23] Cambridge Dictionary, *Echolocation*, English, Accessed: 2026-06-18. [Online]. Available: https://dictionary.cambridge.org/dictionary/english/echolocation.

[24] Introversion Software, *Scanner Sombre*, PC video game (Steam), 2017. Accessed: Jun. 10, 2026. [Online]. Available: https://store.steampowered.com/app/475190/Scanner_Sombre/.

[25] Dowino, *A Blind Legend*, PC video game (Steam), 2016. Accessed: Feb. 3, 2026. [Online]. Available: https://store.steampowered.com/app/437530/A_Blind_Legend/.

[26] R. Falk, *Engaging gameplay for audio games*, 2020. [Online]. Available: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1642651&dswid=-755.

[27] Lucas Pope, *Return of the Obra Dinn*, PC video game (Steam), 2018. Accessed: May 5, 2026. [Online]. Available: https://store.steampowered.com/app/653530/Return_of_the_Obra_Dinn/.

[28] Valve, *Portal*, PC video game (Steam), 2007. Accessed: Feb. 10, 2026. [Online]. Available: https://store.steampowered.com/app/400/Portal/.

[29] Valve, *Half-Life: Alyx*, PC video game (Steam), 2020. Accessed: May 11, 2026. [Online]. Available: https://store.steampowered.com/app/546560/HalfLife_Alyx/.

[30] F. E. Garcia and V. P. de Almeida Neris, “Design Guidelines for Audio Games,” in *Human-Computer Interaction. Applications and Services*, M. Kurosu, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 229–238, ISBN: 978-3-642-39262-7.

[31] F. Rumsey, *Spatial Audio.* London, United Kingdom: Taylor & Francis / Routledge, 2012, ISBN: 9780080498195. DOI: 10.4324/9780080498195. [Online]. Available: https://doi.org/10.4324/9780080498195.

[32] F. L. Wightman and D. J. Kistler, "Resolution of front-back ambiguity in spatial hearing by listener and source movement," *The Journal of the Acoustical Society of America*, vol. 105, no. 5, pp. 2841–2853, May 1999. DOI: 10.1121/1.426899.

[33] C. I. Cheng and G. H. Wakefield, "Introduction to head-related transfer functions (HRTFs): Representations of HRTFs in time, frequency, and space," *journal of the Audio Engineering Society*, vol. 49, no. 4, pp. 231–249, 2001.

[34] Olorama Technology, *How to achieve an immersive game experience?* English, Accessed: 2026-01-29, Feb. 2025. [Online]. Available: https://olorama.com/immersive-game/.

[35] Cambridge Dictionary, *Immersion*, English, Accessed: 2026-01-29. [Online]. Available: https://dictionary.cambridge.org/dictionary/english/immersion.

[36] M. Haggis-Burridge, "Four categories for meaningful discussion of immersion in video games.," English, *Published on Researchgate*, Apr. 2020. [Online]. Available: https://www.researchgate.net/publication/340686774_Four_categories_for_meaningful_discussion_of_immersion_in_video_games.

[37] Webmobril Admin, *The psychology of immersion in video games: How developers craft realistic virtual worlds*, English, Accessed: 2026-01-29, Aug. 2023. [Online]. Available: https://www.webmobril.com/psychology-of-immersion-in-video-games/.

[38] N. Davari and C. Verbrugge, "Efficient 2D Sound Propagation in Video Games," *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 17, no. 1, pp. 132–139, Oct. 2021. DOI: 10.1609/aiide.v17i1.18900. [Online]. Available: https://ojs.aaai.org/index.php/AIIDE/article/view/18900.

[39] C. P. v. d. Doel, "Sound synthesis for virtual reality and computer games," Accessed: 2026-02-05, Ph.D. dissertation, University of British Columbia, 1998. DOI: http://dx.doi.org/10.14288/1.0051161. [Online]. Available: https://open.library.ubc.ca/collections/ubctheses/831/items/1.0051161.

[40] M. A. Baytas. "What is research through design?" Design Discipline, Accessed: May 20, 2026. [Online]. Available: https://www.designdisciplin.com/p/rtd-origin.

[41] H. W. Rittel and M. M. Webber, "Dilemmas in a general theory of planning," *Policy sciences*, vol. 4, no. 2, pp. 155–169, 1973. [Online]. Available: https://www.jstor.org/stable/pdf/4531523.pdf?refreqid=fastly-default%3A81ad36187191bd2cf6e136f7c7bce58e&ab_segments=0%2Fbasic_search_gsv2%2Fcontrol&initiator=search-results&acceptTC=1.

[42] Design Council, *The Double Diamond*, Accessed: 2026-02-10. [Online]. Available: https://www.designcouncil.org.uk/our-resources/the-double-diamond/.

[43] K. Beck et al., *Manifesto for Agile Software Development*, Accessed: 2026-02-10, 2001. [Online]. Available: https://agilemanifesto.org/.

[44] K. Schwaber and J. Sutherland, *The Scrum Guide*, Accessed: 2026-02-10, 2020. [Online]. Available: https://scrumguides.org/scrum-guide.html.

[45] A. Salley, *Vertical Slicing: A Game-Changer for Agile Development*, Accessed: 2026-02-13, 2024. [Online]. Available: https://aaronsalley.medium.com/vertical-slicing-a-game-changer-for-agile-development-98aac11fb05b.

[46] Teravision - Marketing Team, *Advantages of Vertical Slice over Horizontal Slice in Agile Software Development.* Accessed: 2026-02-13, 2018. [Online]. Available: https://www.teravisiontech.com/blog/advantages-of-vertical-slice-over-horizontal-slice-in-agile-software-development.

[47] G. Lucassen, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper, "The Use and Effectiveness of User Stories in Practice," in *Requirements Engineering: Foundation for Software Quality*, ser. Lecture Notes in Computer Science, vol. 9619, Springer, Cham, 2016, pp. 205–222, ISBN: 978-3-319-30281-2. DOI: 10.1007/978-3-319-30282-9_14.

[48] J. W. Creswell and J. D. Creswell, *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2017.

[49] M. Scriven, "The Methodology of Evaluation," in *Perspectives of Curriculum Evaluation*, ser. AERA Monograph Series on Curriculum Evaluation 1, R. W. Tyler, R. M. Gagné, and M. Scriven, Eds., Chicago, IL: Rand McNally, 1967, pp. 39–83.

[50] T. Sammut-Bonnici and D. Galea, "SWOT analysis," *Wiley Encyclopedia of Management*, vol. 12, no. 1, pp. 1–8, 2015.

[51] Godot Engine contributors, *XR: Heavy flickering on Meta Quest Air Link*, GitHub issue, 2026. [Online]. Available: https://github.com/godotengine/godot/issues/114987.

[52] Wikipedia contributors, *Foley (sound design)*. Accessed: Jun. 8, 2026. [Online]. Available: https://en.wikipedia.org/wiki/Foley_(sound_design).

[53] Audacity Team, *Audacity*, Accessed: 2026-05-18. [Online]. Available: https://www.audacityteam.org/.

[54] Noteflight, *Noteflight*, Accessed: 2026-05-18. [Online]. Available: https://www.noteflight.com/.

[55] Cakewalk, *Cakewalk Sonar*, Accessed: 2026-05-18. [Online]. Available: https://www.cakewalk.com/sonar/.

[56] Plugins4Free, *Plugins4Free*, Accessed: 2026-05-18. [Online]. Available: https://plugins4free.com/.

[57] Malaclypse the Younger, *Breezy Day*, Accessed: 2026-05-18. [Online]. Available: https://plugins4free.com/plugin/3188/.

[58] Quilcom, *Christmas Belly*, Accessed: 2026-05-18. [Online]. Available: https://plugins4free.com/plugin/3305/.

[59] ToneCarver, *Wind Chimes*, Accessed: 2026-05-18. [Online]. Available: https://plugins4free.com/plugin/1555/.

[60] Innersloth, *Among Us*, PC video game (Steam), 2018. Accessed: May 24, 2026. [Online]. Available: https://store.steampowered.com/app/945360/Among_Us/.

[61] OpenAI, *ChatGPT*, 2026. Accessed: Jun. 9, 2026. [Online]. Available: https://chatgpt.com/.