Aalborg University Copenhagen

Aalborg University Copenhagen
Frederikskaj 12,
DK-2450 Copenhagen SV
Semester Coordinator: Stefania Serafin
Secretary: Judi Stærk Poulsen

Semester: 9th

Title: Connecting games to better spatialized audio

Project Period:
1st of September - 18th of December, 2020

Semester Theme: Master’s Thesis

Supervisor(s):
Stefania Serafin
Simone Spagnol

Members:
Jonas Siim Andersen

Abstract:
This project aims to establish if an interactive game experience can both be evaluated and enhanced by using an individualized Head Related Transfer Function (HRTF).
Relevant literature concerning frameworks for spatial audio in games using HRTF has been analyzed. The State of the Art examines research papers and commercial games to see how individualized HRTFs could improve a spatialized experience. Afterwards, the design process started. Here, a set of requirements were established and tested using iterative steps.
Once ready to be evaluated, the game was implemented using the Unity Engine and Steam Audio API. The implementation took player through three instances - an instance without an HRTF, an instance with a generic HRTF and an instance with an individualized HRTF.
Results found that the individualized HRTF performed best in the localization trial and users were able to describe the benefits as well through a QoE (Quality of Experience). Further enhancements could be made, such as testing this game for competitive gamers that would have their game performance improved by an individualized HRTF.

AALBORG UNIVERSITET

Master's Thesis

- Connecting games to better spatialized audio

Jonas Siim Andersen

Supervisors: Stefania Serafin & Simone Spagnol

Contents

1

Introduction

Current advancements in hardware technology provide enough processing power to start crafting realistic experiences in both the domains of Virtual Reality (VR) and Video Games. The jump to 8-core processors and above [1] not only provides the user capabilities to run games and VR experiences in larger and more expansive virtual worlds than ever before, it also provide the computational power to start crafting realistic audio experiences that abides by the characteristics of the virtual spaces. The result is that developers are looking for methods to enhance the realism of virtual audio. For example, the Sony PlayStation 5 has not only been focusing their marketing campaigns on providing more expansive and prettier visual experiences, but they have also put a lot focus on their tempest audio engine promising for more realistic rendering of 3D Audio than ever. [2] This includes systems for picking best-fit Head-Related Transfer Function (HRTF) profiles for the user [3]. This opens the door for many game and VR developers to start using binaural models to give their games a more realistic auditory feedback, but since the model of an HRTF is quite personal to the user, methods trying to use individualized HRTFs for games remain to be further investigated.

The problem statement that this thesis sets out to explore is,

"Can an interactive game experience evaluate if an individualized HRTF would improve player performance and auditory quality?"

To find a possible answer, this thesis will explore relevant literature in order to craft a framework on how to understand spatial audio and the binaural rendering of it. Additionally, assessments of spatial audio will also be explored. Hereafter, a set of design requirements will be defined in order to craft a prototype through iterative steps. In the implementation, the final prototype will be documented, whereas in the evaluation and discussion, the outcome of the implementation will be analyzed in order to try and answer the problem statement.

[1] Next-generation hardware: `https://www.ign.com/wikis/playstation-5/PS5_vs._Xbox_Series_X_Comparison_Chart`

[2] Tempest Audio Engine: `https://bit.ly/3ntnflM`

[3] The Road to PS5 `https://youtu.be/ph8LyNIT9sg?t=2561`

2

Literature Review

2.1 Spatial Audio

The human hearing system localizes sounds in our surroundings by judging their distance and direction in respect to our current position. A popular theory behind understanding how we localize sound sources is known as the Duplex Theory. According to this theory, we analyze the time differences (Interaural Time Delay) as well as the sound level differences (Interaural Level Difference) of the incoming sounds at our eardrums. To translate this to a digital domain, we make use of Spatial Audio which provides a third dimension to sound. It allows for the listener to localize sounds that exist in a 3D space. [1]

Especially in recent years, the popularity for VR has boomed among consumers and developers. This results in a greater need for crafting both realistic and efficient spatial audio. [2]

2.1.1 Spatial Audio in Games and VR

Spatial audio in games does not only provide a more immersive experience for players, it also aids the users to navigate the in-game environments. [3] In a 3D environment, a mix of visual and auditory localization would prove to be the strongest localization method due to the auditory cue giving the players a reference with the visual stimuli to confirm their observation.

Additionally, spatial audio in games can also further enhance the overall sonification process. The information stored within the game can be visually disconnected by instead presenting it through a series of explicit sounds that influences the player's decision making. It could for example be that the system is pinging the player through sounds that they are low on health, but they hear something dangerous closing in on them from the distance prompting them to play defensively. [4]

2.1.2 Methods for modelling Spatial Audio into Virtual Experiences

When it comes to implementing Spatial Audio, some different enhancements can be applied depending on platform and system specifications.

2.1.2.1 Distance-based attenuation

For spatial audio, a model of distance-based attenuation curve, as well as reverberation, is commonly seen [1]. It includes measuring the player's position and through a curve that attenuates the sound source depending on the distance. Also, the direction of the sound gets panned using the left and right audio channels. If trying to emulate the feeling of sound propagation through space, an amount of reverberation can be applied as well to simulate vast open environments or specific rooms. Plus, a low pass filter can be applied to simulate high frequencies being absorbed over distance. Most noticeably, the loudness gets lowered the further away the player moves from the source giving a sense of distance, and depending on the player's rotation, the sound source is also panned accordingly giving a sense of direction. To further enhance the immersive listening experience, the model can be supplied with the use of the doppler effect [1] that adjusts the sound depending on the player's or the sound source's velocity. The type of curve can also be changed to a linear attenuation or a logarithmic attenuation depending on how the sound best fits in with the virtual environment [2].

2.1.2.2 HRTF filtering

It is possible to further enhance the distance-based attenuation model using HRTF filtering for both ears, which in result enhances the localization of front-back sound sources, as well as sound sources that are elevated. Front-back and elevated sound sources often poses a problem to localize accurately when the spatial audio is not being filtered by an HRTF.[1][4] The filter is expensive to compute in real-time depending on the filtering applied, so it is often seen as an option for systems with stronger specifications where weaker hardware tends to use a standard distance-based curve with no filtering. The theory behind HRTF will be more thoroughly explain in later section 2.2.

2.1.2.3 Parametric/Convolution Reverb & Sound Propagation

There exist two known approaches when enhancing the reverberation of the spatial audio model called parametric and convolution reverb profiles [1]. Parametric reverb can help create a rough representation of the space within the virtual environment. The implementation of parametric reverbs are very similar to digital effects you for example find in digital plugins, where a dry signal is filtered with certain parameters. In the case of virtual experiences, these parameters could be controlled by the player's position. For example, whether they are inside a tunnel or not, the wetness of the reverberation will be added accordingly to the dry signal of their footsteps and other actions taken.

To get realistic representations of space within the virtual world, a convolution profile can be used. Each space has a unique impulse response (IR) profile, that can be convolved onto the signal to emulate the sound source being played within that same space. However, in virtual environments, the IRs would realistically change consistently depending on where the active listener is positioned since the IRs are recorded from a fixed position. To model this change accordingly, it will be necessary to simulate sound propagation by using ray-tracing.

2.2 Head-Related Transfer Functions

As stated previously, the human hearing system is capable of locating sound in a third dimension. However, this feat is quite a personal process, since the brain is capable of analyzing sounds concerning the disparity in for example time and frequency. Physical proportions such as the shoulders, head and ears also influence the perceived sound. All

[1] The Doppler Effect: https://web.archive.org/web/20170914003837/http://www.einstein-online.info/spotlights/doppler

[2] Types of roll-off: https://docs.unity3d.com/Manual/class-AudioSource.html

of these parameters are measured using an HRTF when applied to a digital system, gives the effect of localizing the virtual sound as they would in real life, giving the improved perceptual features.

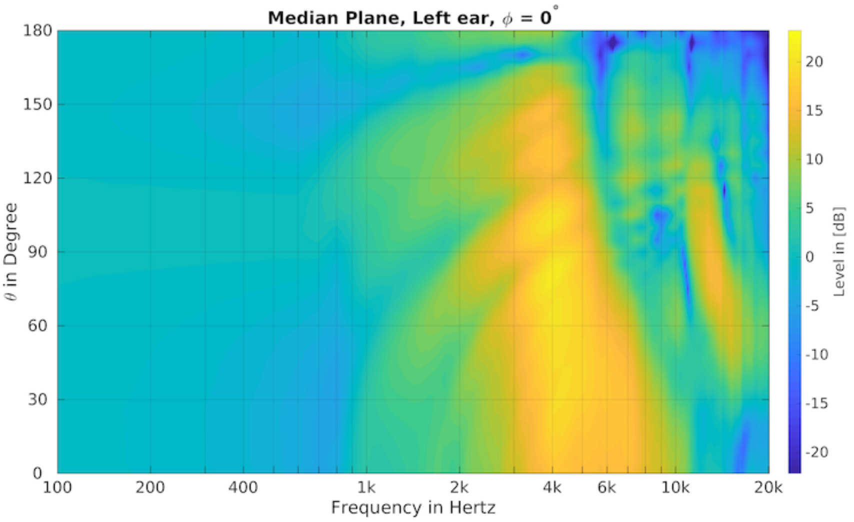


Figure 2.1: A measurement of a HRTF on the left ear, where the level of sound depends on frequencies and degrees in the median plane (elevation of the sound) [3]

Left and right ear HRTFs are measured using a common approach where both ears have probe tube microphones partially inserted into a subject's ears [5]. The microphones picks up sound played from an array of loud speakers that are positioned at a specified azimuth, elevation and distance with respect to the subjects head as the origin. It is therefore possible to measure a transfer function that quantifies the spectral and temporal cues for spatial hearing. It can take quite a long time to retrieve accurate measurements for a subject [6], so often artificial heads are used to also create "approximate" HRTF that becomes generalized HRTFs as opposed to an individual HRTF of a human test subject.

[3] Retrieved from `https://www.akustik.rwth-aachen.de/go/id/pein/lidx/1`

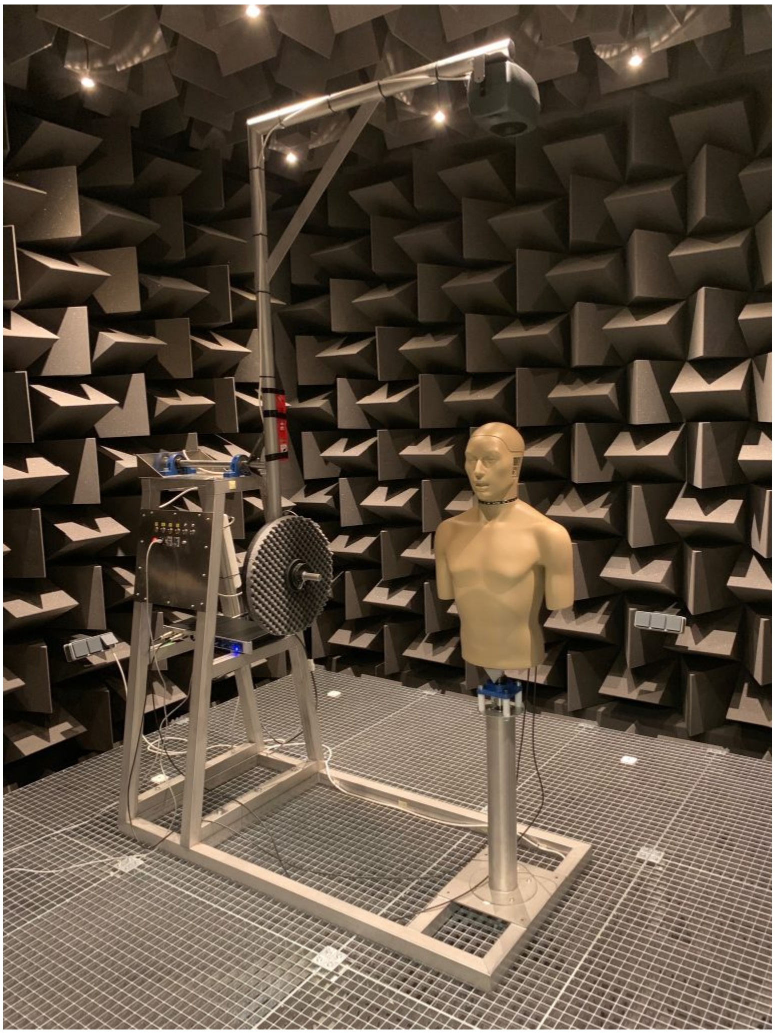


Figure 2.2: A mannequin being set up for a HRTF measurement session [4]

Furthermore, the setup for the HRTF measurements has to be done in a designated area with quite a lot of space available as seen from figure 2.2 above. This does not exactly give the privilege to anyone to start measuring their own HRTFs with ease, so the next section is going to dwell further into how generic and individualized HRTFs differ.

2.2.1 Comparison between a generalized and individual HRTF

As derived from above, the rather complicated process of measuring an individual HRTF leaves most binaural implementations to use a model that covers "general" good measurements of spatial hearing. One might ask, it is is even necessary to implement individualized HRTFs for binaural rendering and what benefits do they hold over a generalized HRTF?

Møller et. al. [7] carried out a research project where the test participants got exposed to individual binaural records and non-individual binaural records. From their data, they could prove an increased number of error in the non-individual binaural recordings of both nearby and distant sound sources played from the median plane, suggesting that individualized binaural profile can improve the performance for sound localization. Wenzel et.

[4] Retrieved from `https://itsadive.create.aau.dk/index.php/news/`

al. [8] also found distortions for front-back and elevation locations when using generalized HRTF, though she argues that her test participants still had generally a good idea of directional information with the generalized HRTF.

Research suggests then, that use of a different HRTF profiles can be an anchor to measure the accuracy of localizing sound sources.

While a generalized and an individualized HRTF shows interesting results in comparisons, considerations should also be made into how the quality of the synthesized sounds are perceived.

2.3 Assessing Spatial Audio

Rozeen Nicol et al. [9] argue that the multitudes of approaches to simulate real-life audio listening within virtual experiences using spatial audio calls for a more in-depth look on how to assess the difference in the models, especially for binaural audio that has quite subjective results depending on the HRTF set used.

In many cases, HRTFs also tend to only be weighed by the accuracy of localization and not other categories such as the timbre. In such cases, it would seem ideal then to measure the timbre by using the Basic Audio Quality (BAQ) model, where a *decline* in the given signal can be measured in comparison to a certain reference. However, BAQ was not meant for the assessments of HRTFs and was instead developed for measuring audio codecs.

Additionally, the traditional BAQ metrics don't account for errors or inaccuracies that can be introduced by the binaural content, for example, small errors in measuring individualized HRTFs. There is no steadfast theoretical proof either, that an individualized HRTF should provide the best possible audio experience for the user.

What's proposed instead is to find a means to evaluate binaural rendering systems that does not measure localization accuracy alone.

What Rozeen Nicol et al. propose instead for the binaural content, is using Quality of Experience (QoE) to measure the user's experiences within the system. Simply put, it is the user's subjective stimuli and experiences within the system.

An example could cover room-related attributes, which the space that the sound source is played from has a large effect on the perceptual attributes.

Other subjective data on the perceived quality of spatial sound reproduction also goes into the idea of of having attributes that can be ranked. One frequently used attribute being that the spatial sound reproduction can be either experienced naturally or artificially among listeners. [10]

Other anchors that describe the attributes of spatial quality could also be (but not limited to) brightness, richness, externalisation and preference. [6]

3

State of the Art

3.1 Research utilizing interactive evaluation of Binaural Spatial Rendering

This following section will dig into some research papers that are relevant to the problem statement proposed in the introduction chapter.

3.1.1 Impact of HRTF individualization on player performance in a VR shooter game II

David Poirier-Quinot and Brian F.G. Katz [11] tried to research and characterize the impact of binaural rendering quality using different setups in a VR application. The hypothesis of their research is that using individualized HRTFs will increase the participant's performance measured in reaction time and movement efficiency as the difficulty in the game escalates.

The HRTF database applied for the experiment was acquired from the LISTEN database [12] and a model for individual profile selection [13].

Their results showed that there was no impact of significance when applying an individualized HRTF to their VR game, however, presentation order of the HRTF had a significant difference on HRTF performance.

The VR shooter game only examines performance on localisation in respect to time and movement. As covered in the previous Chapter, there are more dimensions to measure in order to let the user evaluate the quality and preference of a given HRTF. For a *complete* assessment of HRTF quality - a QoE to also evaluate the spectral content of the HRTF could be evaluated as well.

3.1.2 Audio Quality Evaluation in Virtual Reality: Multiple Stimulus Ranking with Behavior Tracking

To assess the spatial audio quality of a VR experience, Olli Rummukainen et. al. [14] crafted an implementation which uses a stimulus ranking system inspired by the Multiple Stimulus with Hidden Reference and Anchors (MUSHRA). The implementation also utilized behavioral tracking to gather more data on their test participant's stimuli. They implemented three different "renderers", that the test participants were subjugated to. The first renderer applied a structural model which is able to synthesize HRTFs in real time. The two other renderers were generalized or 'generic' HRTFs acquired from mannequin heads.

Furthermore, they conducted the experiment for both VR and a desktop environment.

Their findings showed that the use of VR had more discriminability over the desktop environment. They argue that audio quality evaluation for VR experiences has to be done with VR.

3.1.3 User selection of optimal HRTF sets via holistic comparative evaluation

The idea to have users evaluate a set of generic HRTFs has been explored by Rishi Shukla et. al. [15]

Their methodology is implemented as such, that their participants have to compare pairs of songs that are binaurally rendered. For each pair, the same songs plays but convolved with different HRTFs. The participants are given the option through their software to rotate the song around their head and rate their preferred HRTF.

From the outcome of the selection process, the participants has to complete a music browsing task where they navigate a 2D binaural auditory scene that contains 15 songs using the HRTFs assigned to them.

The best-fit HRTF is given to the user through a selection tournament using a robin tournament structure. The participants are asked to choose the set of HRTFs that references the best realism of the spatialisation.

From the results gathered, they could see that their system shows results in consistently identifying optimal HRTFs among the participants. Moreover, they did not seem to find any patterns that suggests a big difference between expert and novice participants, making it a great method for both groups.

However, future work must assess if this method is also viable for 3D binaural renders, since their system only used a 2D display. Additionally, more data must also be required in terms of examining the user's performance. Examples could be time taken, or selection strength and success rate.

Overall, the research shows promising results giving novice and expert users control ranking a holistic set of HRTFs.

3.2 Games utilizing a focused attention on Binaural Spatial Rendering

As stated previously, games and VR experiences makes great use of spatial rendering for their audio solution. Here some games that focuses on the player's perceptual hearing ability will be examined, as well as highlighting certain gameplay elements that better HRTF quality could improve.

3.2.1 Half Life: Alyx

Due to the PC gaming VR market not being as immense compared to more traditional games [1], it is not often to see large development studios spend an enormous budget on crafting a VR experience with a focus on content and polish as Valve has done with Half

[1] Steam Hardware Survey `https://bit.ly/3ralhcg`

Life: Alyx. Half Life: Alyx uses the Steam Audio API to craft a more immersive binaural experience, since the game doesn't rely on competitive online play but instead a single player experience exploring the narrative further of the acclaimed Half Life universe.

In an interivew, the developers explained that they went to great length to spatialize every element in the spaces of the environment as their internal testing found that with VR players tend to spend more time observing a space than more traditional games [2]. They also mentioned that bringing in the iconic traditional sounds from the older Half Life games were hard to spatialize due to the very limited frequency spectrum of these sounds. They had to carefully craft every sound with a larger frequency spectrum in order to suit the binaural rendering.

They also mentioned that the game has a segment, where the player is guided towards the direction by a very distinct sound playing from a certain location, fully exploiting the idea of localizing sound as a navigational gameplay element.

3.2.2 Hunt: The Showdown

Hunt: The Showdown is a competitive online shooter game where players explore a big terrain while trying to solve personalized objectives. Players are giving no indication of where each other are, and at times they will cross each other's path that results in a player vs. player shoot out. The game has received a lot of praise from critics, especially for how realistic the audio feels, and how important the audio systems also play into the gameplay. Many small details in the environment will give off sound, like branches and leaves, if another player is trying to sneak up on you. In many similar scenarios you will hear the enemy player before seeing them.

Crytek, who are the developers behind the game are using their own generalized HRTF [3], which they explain they've spent quite some time on tweaking for the game. The plugin they use is called CrySpatial and was developed alongside the company's VR games prior to Hunt.

The developers described themselves that they want the player's ears to be an 'extra' set of eyes, showing the potential for experiences relying on more dimensionality for localization gameplay than just visuals.

3.2.3 Escape from Tarkov

Escape from Tarkov [4] is yet another popular online competitive shooter that incorporates a lot of similar elements as Hunt: The Showdown. However, what makes it worth mentioning is that the game is built using Unity and incorporates the Steam Audio API for its binaural rendering. This actually opens up the possibility of using customized HRTFs, if the game would explore this opportunity in the future. Steam Audio will be introduced later on in section 5.2.1.

Interestingly, the game also allows the player to 'tweak' the frequency response of the binaural audio by wearing different "combat" headsets. The choice of headset for the player is completely subjective to what they feel like they perform best with in terms of hearing close and distant enemy players lurking about.

[2] Half-Life Alyx Interview `https://www.asoundeffect.com/half-life-alyx-vr-sound/`

[3] Binaural Audio in Hunt - The Showdown: `https://www.huntshowdown.com/news/binaural-audio-in-hunt-showdown`

[4] Escape from Tarkov: `https://www.escapefromtarkov.com/`

Ideas such as different HRTF profiles becoming objects in the game in the form of headsets or ear plugs is also an interesting idea that could be exploited for games.

3.2.4 Counter Strike: Global Offensive

Counter Strike: Global Offensive (CSGO) is a worldwide phenomenon that has existed for a long time and is a big player in the competitive e-sports field. This game also gives very discrete information about where enemy players are. The top competitive players rely on extreme perceptual skills to make out footsteps and gunshots in order to locate the enemy team and surprise them.

The game in the past used a basic model of Spatial Audio, up until in 2017 when the game was updated to include binaural audio in the model as well, starting the motivation to develop the Steam Audio API, (as discussed in later section 5.2.1). The early implementation of Steam Audio had users complaining that toggling on the use of HRTF in the game gave it a significant drop in sound quality [5]. Valve Software has since then been able to take in player feedback in order to construct a better generalized HRTF and binaural implementation for their players.

Like with Tarkov, this game through the API would also support custom HRTF data quite easily, but this is something that the developers have not touched upon as of yet.

[5] Feedback regarding HRTF and Sound Occlusion: https://www.reddit.com/r/GlobalOffensive/comments/7a3ec7/feedback_regarding_hrtf_and_sound_occlusion_311017/

4

Design

4.1 Requirements

In the previous chapters, spatial audio rendering for games was explored in further depth. To craft a prototype that can test the problem statement, the following requirements should be considered:

1. The game should employ a strong focus on the dimensionality of spatial audio - in this case, spatial localization and spatial content.
    - The game should fetch metrical data that can give an indication of the player's performance with accurate sound localization
    - The game should allow the user to rate the quality of the experience. Both in terms of how they perceived the accuracy of localization and quality of timbre
2. The game should focus on modelling sound that is accurate to its space in order to hold realistic expectations as a reference
3. Since the game might subject the user to repetitive tasks spanning multiple HRTF renderers - the game should include gamified elements that makes it fun and pleasing to play and thereby upholds the player's attention span.
4. The game should be easily accessible and not test the player's skill on understanding the game, but it should test the skill in accurately locating sound sources within the HRTF renderers.
5. Due to the expensive cost of rendering binaural spatial audio, the game should also focus on optimization in case that powerful hardware specifications aren't available to the user.

With the requirements laid out, the design process can start.

4.2 Methodology for testing the design

To end up with a solid prototype that adheres to the requirements set for the design, the method of Agile Software Development [16] will be applied to this project. Early stages of the prototype will be sent to a small amount of participants. Their feedback will assessed and taken into considerations for developing the next iteration of the prototype. In this case, a set of participants will get to try out the game and leave feedback for it.

Fullerton [17] also recommends this approach of a continuous, "iterative process of playtesting, evaluating, and revising", and get started as early as possible. The longer the game gets developed, the harder it will be to change fundamental gameplay structures that the developers have built when starting out.

The best approach to commence a successful play test is by warming players up for a discussion by previous games they have played. Hereafter, you simply let them play your prototype for 15 - 20 minutes and take observations. This is followed by a discussion of the experience of the game with a set of sample questions that can be used for each player. A method for this type of play testing could also be to host it remotely with a feedback form which is sometimes done by the likes of Microsoft Game Studios. Moreover, methods like recording player metrics are also a normal process of the playtest to assess certain features with the game.

4.3 Design Ideas

In this section, ideas gathered from the requirements that lead to the first iterative design of the prototype are presented.

4.3.1 Establishing the genre

As explored in the previous section about applying spatial audio to virtual experiences, VR in particular strikes out since the head-mounted device (HMD) allows to user to observe the virtual world through head tracking. It is possible to use the medium to create fun gamified tasks that measure the user's accuracy for localization tasks of spatial audio as explored from earlier section 3.1.1. However, if the purpose of the project is to compare multiple references in the form of HRTF renderers, the measurements of spectral content (timbre) and the overall quality of experience is equally of importance (see section 2.3).

Crafting an experience that can be properly assessed for its binaural qualities needs to have more content than just simplified tasks. It needs to model real-life space and the coherence within, which can be used as a reference to how "life-like" the experience felt for the user.

Examples could include an experience similar to the first-person perspective games explored in chapter 3. Notable mentions in the realm of VR experiences that both translates and advances the more traditional first-person shooter genre (FPS) to VR are, for example, Half-Life: Alyx (see section 3.2.1). The concept of an FPS applies well to assessing binaural localization due to the semantic nature of 'locating' the enemy target and shooting it, which simplifies the localization task to a level that everyone understands. The first person perspective also requires believable graphical quality and sound quality to fully make the player immerse themselves as the player character, adding semantic to assess the spectral content of the binaural quality.

For these reasons, the project should be modeled close to a more traditional FPS experience. However, due to the current novel coronavirus impacting the world at large during the writing of this paper, the design of the genre needs to be scalable for both VR and a traditional method of control (Keyboard & Mouse) in case that lab testing is prohibited. A way to solve this is to start designing the game as a traditional FPS experience like the games seen from section 3.2, which can then be translated to VR if the opportunity is possible during the design process.

4.3.2 Modeling the experience

For the experience to qualify for assessing the binaural space to real life references, the virtual world is inspired by the criteria set up by Rozeen Nicole et at. (section 2.3). For example, the room related attributes should impact how the entire level is designed. It shouldn't be a holistic virtual world, instead, it should focus on taking the player through different environments that impact the timbre of the sound. More specifically, modeling the reverberation accordingly to the space the player is in as explored from section 2.1.2.3.

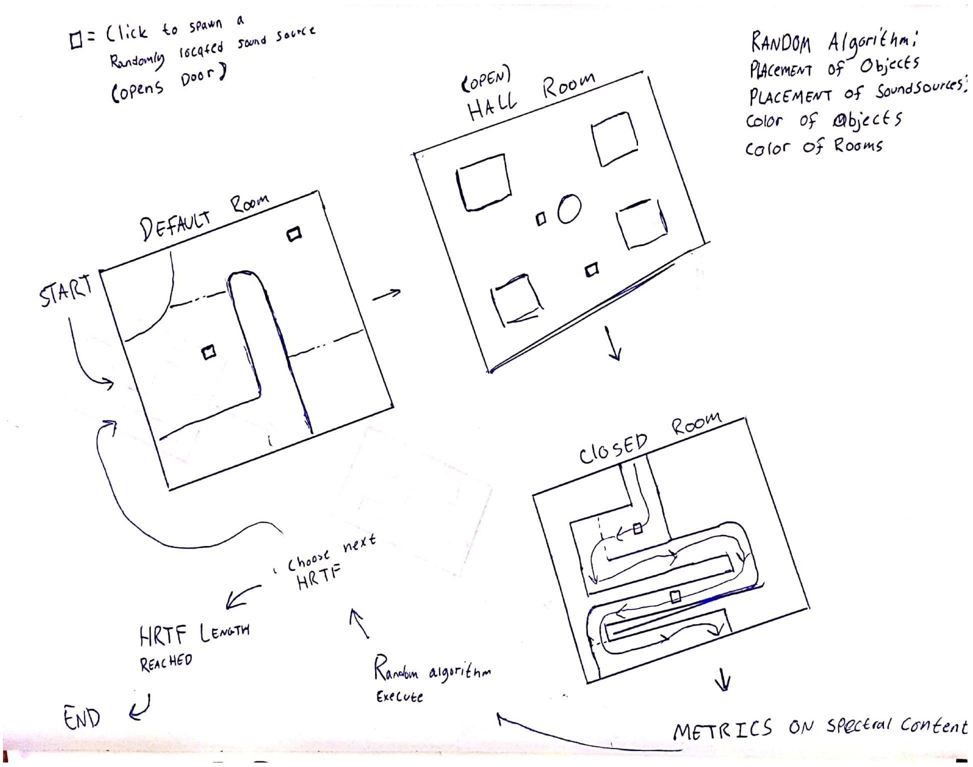


Figure 4.1: Early design sketch of the three auditory spaces.

Examples of how different spaces could be designed can be seen in figure 4.1. The sound propagation in each space would be separately distributed. In the first space, the player starts in an open area with a lot of objects scattered about to absorb the reflection of the sound using approximately a mix of dry and wet reverberation. Hereafter, the experience takes them into an open hall with a lot of wet reverberation. The last area, which is meant to be a small and closed space, will subject them to fully dry reverberation.

These spaces should also be designed in such a way that allows the user to localize specific sound sources by 'tagging' them. Both assessing the coherence of the sound propagation within the space, but also how localization is affected by each space.

Further ideas were also explored such as, randomizing the spaces a bit each time the player went through them to not make the environment feel too repetitive. The reason for this is that to fairly compare the HRTFs with each other, the player would have to go through the same level/instance of the game again using a new HRTF that can be compared with the previous reference. With many HRTFs loaded into the system to assess, concerns started to raise for a very repetitive and boring experience, losing the player's interest by the last set of HRTF profiles.

4.3.3 Modeling the sound output

In the world of film theory, sounds can be split into diegetic sound and non-diegetic sound. Diegetic sound is described as sound belonging to the fictional world and thereby the sound source exists in that world. On the other hand non-diegetic sounds could for example be a musical score that exists externally. Same can be applied to games, but with some changes to not lean too heavily on terminology that is meant to describe films. For example, sounds can be split into proactive and reactive sounds. [18]

Proactive sounds uses the sonification element of game sound to give a message to the player, which requires action from the player. They can be measured in urgency, where high urgency sounds such as the player being hit or taking damage requires an immediate action taken. Medium urgency sounds are still important, but allows the player to deal with it later, whereas low urgency sounds can be information that does not influence the player much.

For reactive sounds, they play according to the action taken by the player. Examples could be firing a gun, or sounds of fast footsteps that lets the player know they are moving at a quicker speed. Overall, both proactive and reactive sounds should be well balanced by the sound designers to create a solid informative system to the player using purely auditory spaces.

For the prototype, each sound should correspond to the space it is in, and thereby apply an abundance of diegetic sounds that references real world explicit sounds. Localization tasks should exploit the use of high urgency proactive sound, whereas the reactive sounds committed by the player should stay coherent to the space the sound is in.

4.3.4 Optimizing the User Experience

The game needs to have simplified and easy to use systems since the user should be exclusively measured on how they perform the tasks given to them and not struggling to figure out how the game should be controlled and played.

One way to achieve this is by familiarizing the players with the genre. The game should deploy the same layout of controls that people are used to for first-person experiences, such as using the mouse to control the player's head orientation and clicking the mouse button to fire a projectile towards their orientation. Additionally, limited use of key bindings should be enforced to make the user learn the control quickly so they can focus on the task given to them.

It is also important to convey information to the player through the proper use of a User Interface (UI), which shows 2D images and text overlay on top of the game screen. Through the UI, such as a pause/configuration menu, the user gets to access the configuration of the system such as volume level, if they feel the sounds are too quiet. UI elements such as waypoints can also guide the player towards their objective, so they don't get lost and frustrated within the virtual space.

Proper design and use of UI is should receive extra focus in this project since, in the case of remotely testing the participants, different systems and configurations are going to be running the game that the designers of the experiment cannot optimize for specifically; especially with how expensive rendering binaural audio can be. Therefore, it is necessary to make sure different configurations for how the game is rendered is presented to the users. For example, lowering the effects produced by the graphical rendering pipeline to make the game perform at a higher frame rate. In the case of binaural audio, the method of filtering can also be changed to allow for weaker CPUs to render binaural audio.

4.3.5 Synthesizing individualized HRTF profiles with Deep Learning

In section 2.2, it was mentioned how measurements of human test subjects was not exactly a feasible task. Thereby, different methods to try and optimize the process of retrieving personalized HRTF measurements exists, especially with the application of Machine Learning (ML). [19][20][21]

This project applies an algorithm created by Riccardo Miccini and Simone Spagnol, which uses deep learning to synthesize HRTF profiles from user antropometric data [22]. The synthesized HRTF profiles are based off of the Viking data set, which covers a range of HRTF measurements that were recorded using a KEMAR mannequin with probe microphone tubes [23]. One of the test subjects from the viking data set has its pinnae removed and replaced with flat baffles to obtain a response that only show the effect that the head, shoulders, and torso has. That response is combined with a custom pinnae response that is synthesized using images of the user's pinnae by using deep learning algorithms [22], which then substitutes the original ITD by applying a method for selecting the best-match ITD from head and should parameters [24][25]. What type of anthropometric data is needed for the algorithm to work can be seen from the anthropometric data guide in appendix B.

With the possibility to generate individualized HRTFs using the method described above, the game has access to render an individual HRTF for each test participant for the final evaluation.

4.4 Design Process

This section will explore the iterations that the design process went through. Feedback from each player test will also be gathered to analyze how the design could further be improved

4.4.1 First Iteration

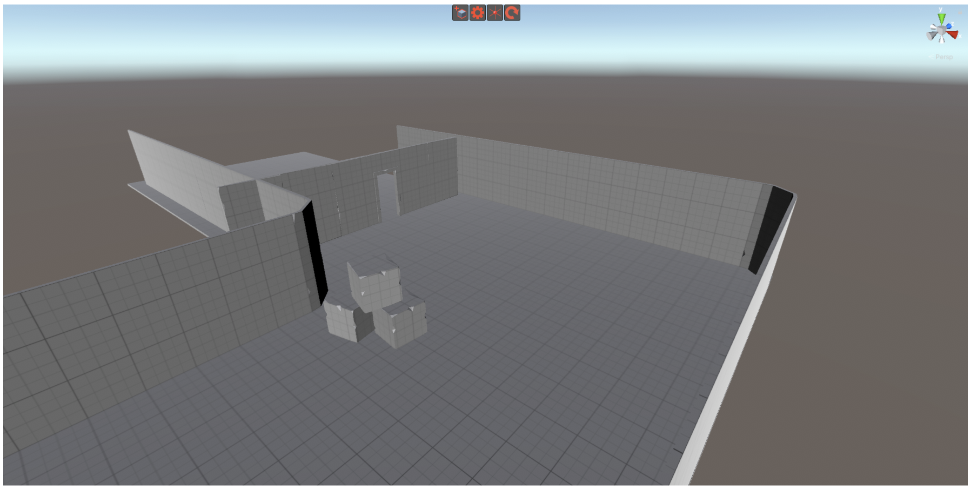


Figure 4.2: Very first prototype with a simple stage to test the in game mechanics.

Before the first iteration of the game could be sent out, an early prototype was developed. The early prototype started with testing some of the ideas gathered from the previous section. Most importantly, the player movement was designed to get the feel of first-person navigation right, along with adding mechanics for shooting a firearm to hit the targets with. Also, the binaural sound was implemented and tweaked to get good results to set up the later test. Plus, adding a framework on how to serialize the player's accuracy in locating the sound sources. This stage of the prototype also involved using free degrees of movement where the player could freely walk anywhere, and the accuracy of hitting the target would be measured using a dot product. However, errors in the validity of the data started to become a concern since there is no fixed path to measure the player from and some players could perhaps find ways to exploit the task by pure accident. Also, measuring the accuracy of localizing the targets using a dot product turned out not to be a valid data of measuring the localizing of targets either. This is because it depended on the player having a very accurate aim to hit the target which would measure the player on shooting skills, not hearing ability.

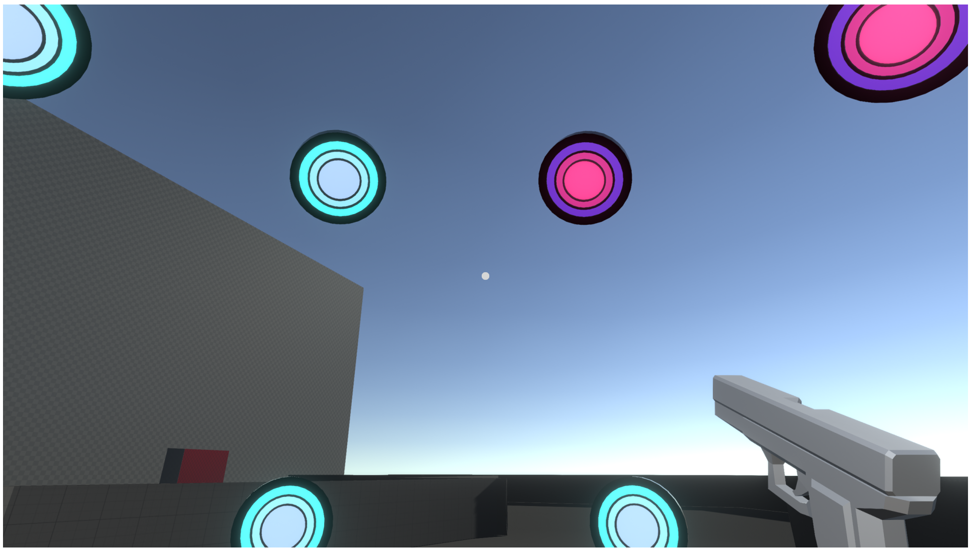


Figure 4.3: In the first iteration of the game, the targets would appear around the player with exact spherical coordinates.

Once the concerns raised from the early prototype was addressed, the game started to take shape. Most notably, the control of the player was changed to be more of an on-rail path instead, like an amusement ride where the player would be sitting in a cart. Thereby, each test participant would move on the same fixed path through each instance and HRTF profile. Targets the player had to shoot would randomly appear along the player's path, playing the sound of a firework (see figure 4.3). Primitive versions of the three auditory 'spaces' (section 4.3.2) that the player would navigate in was added, keeping equal grid sizes across the board to not lengthen the experience to a certain space.

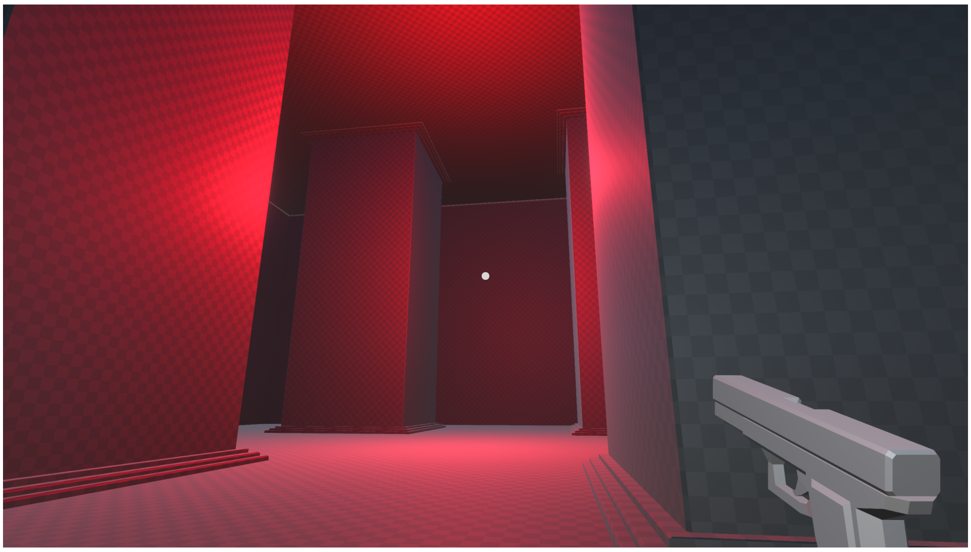


Figure 4.4: First iteration of the second space, which would be a grand hall with a lot of reverberation added in for a later stage of the project.

Once the player had gone through the level, it would restart with a new HRTF loaded in from the RIEC database [26][1] order to make a comparison between each profile. The RIEC database was only used as a temporal placeholder to test the design of the game, rather than the actual experiment.

A few ambient sounds were added into the first space, simulating a city. This included sounds such as birds flying around and cars driving in the distance.

4.4.2 Player Test Feedback on the First Iteration

For this smaller test, 5 participants were sampled. The participants were sampled randomly online through convenience sampling [27]. Due to conducting the test remotely, it was not possible to observe the players over their shoulders and ask them questions throughout the gameplay session as suggested by the theory presented in previous section 4.2. Instead, the players were invited to a Google Forms link, where an install of the game resided. Here there were also some questions they had to answer before starting the game mostly related to their background with games. All of the participants turned out to quite an avid audience of video games, which is perfect for getting more expert opinions on the early stage of the design.

Once they had answered the background questions, an instruction for the game was presented explaining the task they had to do. When they were done playing, they were met with the following questions, with some of them inspired by Fullerton's suggestions:

1. Overall, what are your thoughts about the game?
2. Is there any information that would have been useful to you before starting?
3. Were you able to learn to play quickly?
4. What were the objectives of this game?

[1] RIEC accessed from: `http://www.riec.tohoku.ac.jp/pub/hrtf/index.html`

5. Did you find the objectives easy or difficult, and why?
6. What would you describe happened to the sound in the game everytime you restarted the level?
7. Is there anything that you did not like about the game? If so, what?
8. What would you add or change to this game to make it more fun and engaging to play?

From the first question, the users fully understood the idea behind the game and the task they had to do. However, opinions were mixed with some being confused about what to do exactly in the beginning due the lack of more clear instructions. One user also found the given task fun, but that it was too repetitive to play by the last instance. This was also rooted in the second questions, with one user feeling all the needed information was supplied, however, the rest needed more clear instructions. For the third question, they all felt it was easy to play and got into the gameplay loop quite fast which is very positive when trying to meet the design goal presented in section 4.3.4.

For the fourth question, all of the participants gave a clear understanding of what they had to do which was locate the correct target by sound, and shoot it. For the fifth question, very interesting results started to appear as one user had issues locating targets on the elevation:

"Relatively easy, although the most difficult part for me was to figure out where the target is vertically(i.e. first or second from the ground)" - Participant #1.

Showing inaccuracies with the HRTFs being able to represent the elevation positions accurately.

Also, an answer that is also worth showcasing is,

"The sound changed quite drastically between each restart, which sometimes made it very difficult to locate the correct target." - Participant #3.

This shows the user was sensitive to hear the difference in each instance of the game when presented with a new HRTF profile. It is worth mentioning that none of the users were given clear information about what happened at each instance of the game. They were only told to play through the entire experience. This answer shows that there is indeed an effect happening in each instance.

For the 6th question, participant #3 said that on their 4th instance (fourth HRTF profile), they were practically guessing. The rest of the participants except one started to explain how some sounds could not be heard anymore in certain instances of the game. That the sound of the city and birds could not be heard anymore, showing an effect happening in both the amplitude and frequencies of the HRTF profiles.

Feedback from the 5th question showed that three of the participants said that the cart moved way too slow and especially that got boring during the later instances. They would love more control over it, or for it to move faster at least. Also again, there was a lack of more clear instructions presented to the user, as well as some issues with the color-coding of the targets that were shot.

As for the final question, the participants felt some aspects could be improved. Including a scoring system to make it more exciting, a difficulty curve to make it more exciting for users that are performing well, and more sounds added to the environment in general.

4.5 Second Iteration

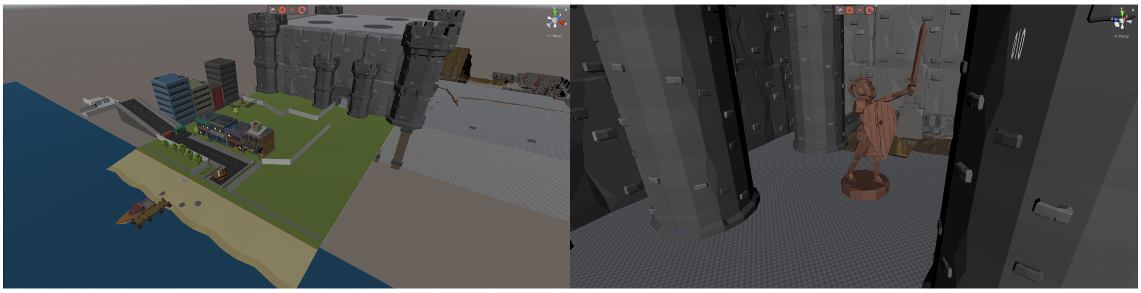


Figure 4.5: Early design of the two first spaces, as the second iteration was being slowly implemented

For this iteration, previous concerns were addressed.

Instead of having the targets appear as a sphere around the player and stopping the cart completely, the map design started to get fleshed out with buildings and small key areas around the three spaces. Also so it would be easier for the users to refer to exactly which areas they had a difficult or easy time with locating the sound sources. Each space got its own identity to stand out.

On these buildings, the targets were attached to various surfaces to make the task feel more organic to the spaces. The issues with the game feeling like a drag due to the slow cart speed were also addressed by designing a method to let people toggle the speed themselves. The cart also did not stop anymore when a localization task would being - instead, it would move at a very slow speed for the player to feel the effect of the localized sound source as the cart is moving.

The UI also started receiving more elements as well. Information about the current instance of the HRTF data set the player was on was now informed visually as level 1/3, etc.

Graphics were also tweaked to make the game look more polished. Higher detail of sky maps, lights, emission, and materials were added to put in more vibrant colors in the game.

4.5.1 Player Test Feedback on the Second Iteration

For the feedback in the second iteration which was running a lot similar to the first iteration, however, an addition of 5 point likert scales in the web questionnaire were also included for this iteration.

A random sample of 6 participants who were different players from the first iteration test was chosen.

This time impressions were much more positive. The participants did not complain about the slow speed of the cart anymore, which showed that the design allowing the participants more control over their speed aided the overall reception of the game. There was some room for further improvements as well.

Some the participants were confused whether they hit the right target or not. They suggested to add a form of sound or visual feedback to point out which target they shot, and which target contained the sound source. Furthermore, the color-coding got a bit confusing since only the colors red and green were used to distinguish the correct and wrong targets.

A question was also added to ask the participants,

"Did you like the placement of the targets you had to shoot? If not, which targets were difficult to locate and where would you place them instead?"

All the participants mentioned that the statue that had targets placed vertically was very difficult for them to locate the source, perhaps indicating the placements are a bit too high on the difficulty curve which could be fixed in later iterations.

For the new likert scales that were added, they asked the following questions

1. What was your first impression?
2. Did the game feel repetitive?
3. Did the game feel too long, or too short?
4. How did the controls feel? Did they make sense?
5. Could you find the information you needed on the interface?

For the ratings the player gave, it left a high first impression. The participants also felt it was repetitive, suggesting to be careful about how many instances they have to go through to not lose their attention.

There were mixed opinions about whether the game felt too long or too short, so the popular opinion is a balance of both.

The participants felt that the control made great sense, fulfilling the design goal of intuition. Most could find the information needed, but results where rather mixed suggesting perhaps a form of an interactive tutorial stage teaching them the task might be best to include.

5

Implementation

5.1 Choice of Game Engine

There exists an assortment of free-to-use game engines nowadays, to code all sorts of experiences or games. Unreal Engine[1] by Epic Games boasts incredible state of the art visuals for real-time rendering. This feature of the engine also started to make it more popular for film production[2]. Due to the Unreal Engine being marketed as a tool for crafting realistic experiences, it's also worth taking a look into its implementation of audio. The newest Unreal Engine 5 includes additions to the audio system[3], including implementation of rendering the spatialized audio with convolutional reverb as introduced in section 2.1.2.3. Unreal Engine 5 isn't released to the public yet and requires modern/next generational hardware that's only being released now (as of writing this paper). It could be an interesting solution in the future for assessing spatial audio due to these huge advancements in audio rendering.

Unity[4] is another popular engine among the game development community. It's often considered the go-to engine for smaller teams and projects due to its great flexibility and third party support. The engine also supports a lot of platforms that it can build games for, allowing the developers to easily port their games to cross-platform functionality. It's especially this feature that made it popular for use with crafting VR experiences, although it's worth mentioning that Unreal Engine also supports VR. Unity utilizes the C# programming language, for implementing logic and custom systems as the scripting API[5].

Due to being mostly skilled in Unity and having more easy access to build the game for either Windows, Mac or translating all of the components to VR, and having support for relevant third party tools needed to realize the implementation, Unity is used as the game engine for this prototype.

5.2 Third-party packages used

The following section will explain and give proper credit to all the third party packages applied for this project.

[1] Unreal Engine: `https://www.unrealengine.com/en-US/`

[2] UE Film Production: `https://www.unrealengine.com/en-US/industry/film-television`

[3] Unreal Engine 5 Introduction: `https://youtu.be/qC5KtatMcUw?t=185`

[4] Unity: `https://unity.com/`

[5] Unity Scripting API: `https://docs.unity3d.com/ScriptReference/`

5.2.1 Steam Audio API

Steam Audio[6] is a free-source Audio API made available by Valve Software that is compatible with both Unity and Unreal Engine. It enhances the simple distance-based attenuation curve model (see section 2.1.2.1) used by Unity by allowing binaural rendering of both a built-in generalized HRTF or custom HRTF to both ears of the player. Furthermore, it also allows for sound occlusion of spaces by using a ray-tracing method, however, this method, as mentioned from section 2.1.2.3, is computationally expensive for the CPU[7] and should be used for hardware with stronger specifications. Enhancing the model with this sound propagation method should be carefully assessed. Valve Software developed the Steam Audio primarily for their competitive game Counter-Strike: Global Offensive, which depends on auditory localization cues for the player to locate the enemy team as mentioned in previous section 3.2.4.

Steam Audio API is also recommended by Beig Mirza et. al (see section 2.1.2) in their analysis of current spatial sound renders, when it comes to using custom HRTF data. For these reasons, the Steam Audio API was imported into this prototype.

5.2.2 List of Third-party Assets used

Other third-party tools used, was mostly in the form of 3D asset packs to construct the three different spaces, without actually having to spend time on 3D modeling all of this one self.

- UModeler by Tripolygon, inc. [8]
  - ∗ This asset was used to construct primitive shapes for the three different spaces during the early iterations of the prototype. It's a 3D Modeler software that exists within Unity, so there are no compatibility issues when importing 3D assets from external programs into Unity. It helped to carve more complex paths, like the underground tunnel that leads into the last space (The cave).
- Polygon Prototype, Simple City, and Knights by Synty Studios. [9]
  - ∗ For creating the overall 3D environment and adding 3D models to the three spaces. The simple city asset pack was used to create the ambience of the city, with shops, cafes, and cars all over to assist with the coherency of the soundscape. The knights asset was used to create the second space - the castle. Finally, the dungeon asset was used to create the graveyard and the tunnel connecting the second and third space.
- Databox for Unity by doorfourtyfour [10]
  - ∗ This asset helps setting up all the serializations inside of Unity, so information on objects can be stored in real-time. It's popular for creating a save game system for games, where items, health, and progress for the player can be serialized and stored in a file. Tweaking it to record metrics performed by the player also proved very useful for this prototype. Also due to the fact, that it supported

[6] Steam Audio: https://valvesoftware.github.io/steam-audio/

[7] Steam Audio Occlusion: https://valvesoftware.github.io/steam-audio/doc/phonon_unity.html#occlusion

[8] UModeler: https://assetstore.unity.com/packages/tools/modeling/umodeler-80868

[9] Polygon assets: https://www.syntystudios.com/

[10] Databox for Unity: http://databox.doorfortyfour.com/

multiple platforms assisting the idea of making the prototype more accessible to participants.

- Universial Sound FX by Imphenzia [11]

  ∗ This huge sound pack contains over 5000+ free sound samples, which is great for finding suitable sound samples for each space. However, the quality of each sample can wary with some being quite low-quality recordings, while others are great, so careful assessments had to be made if the sound sample used was of acceptable quality. This asset was primarily used when placing sounds all over each space.

- Ultimate Game Music Collection by John Leonard French [12]

  ∗ This pack contains a large assortment of royalty-free music as well. Music from this asset was used twice in the game - the first time when the user had to adjust the volume to a comfortable level and the second time when they were navigating through the first space and a song played from a car radio.

[11] Universial Sound FX: `https://assetstore.unity.com/packages/audio/sound-fx/universal-sound-fx-17256`

[12] Ultimate Game Music Collection: `https://assetstore.unity.com/packages/audio/music/orchestral/ultimate-game-music-collection-37351`

5.3 Implementing the Final Prototype

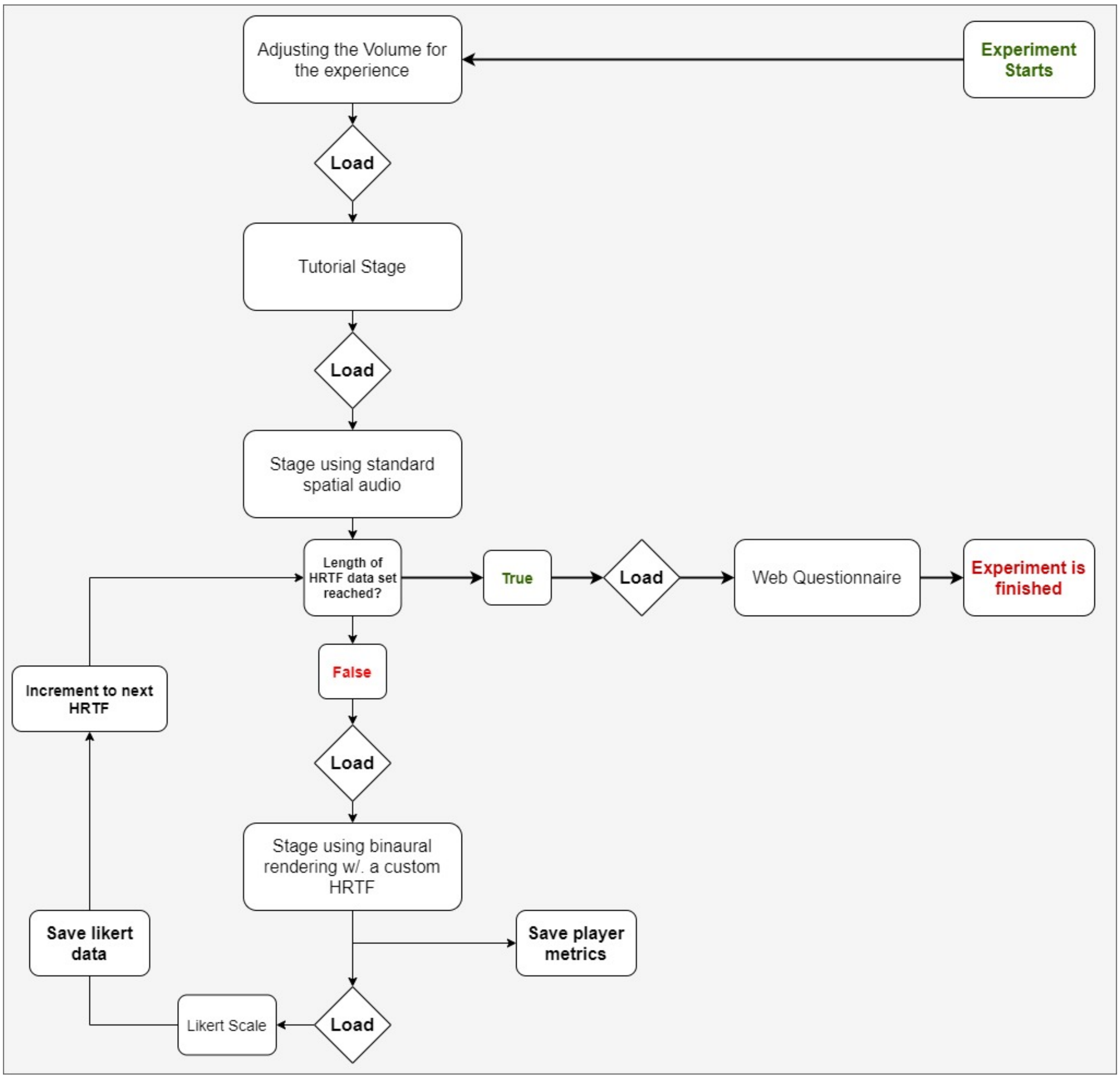
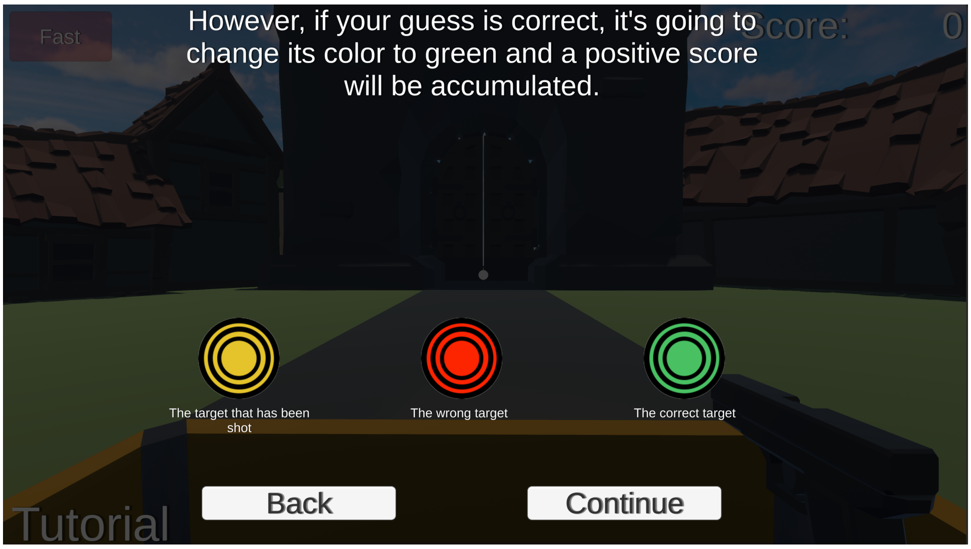


Figure 5.1: Flowchart that shows a holistic view over the entire system for the final prototype.

The final system for the prototype can be seen from figure 5.1, applying the final design goals from the last iteration.

Once the experiment starts, the player is loaded into a 2D program that plays a song. Here the player is asked to wear their best possible headphones, and adjust the volume to a comfortable level making sure they aren't playing the game at a low volume which can impact the data.



However, if your guess is correct, it's going to change its color to green and a positive score will be accumulated.

Figure 5.2: The Tutorial Stage. Here the player is being explained the sound localization task.

Hereafter, they are introduced to the Tutorial Stage (see figure 5.2 above) which is more interactive this time around. The player is given instructions through a heads up display on how to control the speed of the cart they're situated in, as well as the instructions for how the controls work. They're also told what the task of the game is, while they get to try and shoot a dummy target emitting a sound using the basic distance-based attenuation curve.

Once the simple task of the tutorial stage is completed, they get exposed to the first instance of the game, where they drive through the three spaces - City, Castle, and Dungeon locating targets, while the spatial audio model is only using distance-based attenuation like in the tutorial stage. This stage also familiarizes them with the actual task of the experiment, which starts once this stage has finished.

The first HRTF element in the data set is loaded, and the game now enhances the distance-based attenuation curve with a binaural renderer for both ears of the player. For each target the player shoots, the player's vector as well as the target vector is stored in the database, along with the time it took for the player to locate the target (using Time.deltaTime).

Upon finishing the stage, the player loads to a new 2D program which asks them to rate the difficulty of localizing the target as well as the timbre of the sound, on a 7-point Likert scale. The likerts can be seen in section 5.3.4.

The feedback is stored upon submitting the Likert where the system increments to the next HRTF element in the dataset and starts the stage all over again, depending on if the length of the HRTF data set has been reached or not.

If the length of the HRTF data set has been reached, the game opens up a Google Form questionnaire for them instead, where they're asked to give their feedback on the overall experience. The game stores the data and quits the game as a background task. Once they've submitted the final questionnaire they're asked to send the data file which recorded all the metrics and Likert feedback to the experimenter.

Once the experimenter has received the data from the test participant - the experiment is concluded. The entire experiment setup can be read in chapter 6.

The next following sections will go into more depth with how the different parts of the system were implemented

5.3.1 Reading the HRTF data

The HRTFs measurements use the SOFA[13] convention on digital systems for reading and storing HRTF measurements of both individualized and generic HRTFs. The Steam Audio Manager, which is a class that stems from the Steam Audio API (5.2.1), reads the StreamingAssets path of the Unity project to search for HRTFs using the .sofa file format. They are not automatically added to the array of SOFA files in the audio manager, so some custom serialization tweaking needs to be implemented. One example could include creating an array that stores the names of all the sofa files in the StreamingAssets directory:

```
        private void Awake()
        {
            if (!_intialize) return;
            _audioManager = GetComponent<SteamAudioManager>();

            var filePaths = Directory.GetFiles(Application.streamingAssetsPath,
                "*.sofa", SearchOption.AllDirectories).Select(f =>
                Path.GetFileNameWithoutExtension(f));

            _audioManager.sofaFiles = filePaths.ToArray();
        }
```

Listing 5.1: Code to stream SOFA files to the Steam Audio API

In a separate class, static variables are initialized and coupled to the variables stored by the Steam Audio Manager class. These static variables are important for the flow as seen in figure 5.1.

It's important to note that only two HRTFs are loaded for the final implementation. A generic HRTF and an individualized HRTF.

[13] Spatially Oriented Format for Acoustics: https://www.sofaconventions.org/mediawiki/index.php/SOFA_(Spatially_Oriented_Format_for_Acoustics)

```
 1      public class HrtfManager : MonoBehaviour
 2  {
 3      private SteamAudioManager _audioManager;
 4      public static int incrSofa;
 5      public static int sofaLength;
 6
 7      // Start is called before the first frame update
 8          void Start()
 9          {
10              incrSofa++;
11          }
12
13      // Update is called once per frame
14          void Update()
15          {
16              _audioManager.currentSOFAFile = incrSofa;
17              sofaLength = _audioManager.sofaFiles.Length;
18          }
19  }
20
21      public class StageHandler : MonoBehaviour
22  {
23          private void OnTriggerEnter(Collider other)
24          {
25              if(other.CompareTag("Player"))
26                  SceneManager.LoadScene("LikertScale");
27          }
28  }
29
30      public class SubmitButtonAction : MonoBehaviour
31  {
32        { ... }
33
34          IEnumerator DataEventChain(float secs)
35          {
36            { ... }
37
38              if(_lastButton){
39                  yield return new WaitForSeconds(secs);
40                  SceneManager.LoadScene(HrtfSwitch.incrSofa ==
                        HrtfSwitch.sofaLength ? "FinishExperiment" : "FirstStage");
41              }
42          }
43  }
```

Listing 5.2: How variables read from the Steam Audio Manager class can keep track of the current sofa file and use that input to exploit which stage of the system to load

Line #23 introduces the physics component of the game, where the game engine can detect when game objects collide with each other. Generally, this collision logic is used throughout the game to fire off certain events. Such as changing the level seen from the code snippet above, but also triggering audio and the targets the player has to shoot in the other components of the game.

5.3.2 Target Logic

A key area of the prototype was also to implement the localization task, where the player would be presented with different targets throughout the three spaces. The following section will highlight the code related to this task.

```

public class FiringLogic : MonoBehaviour
{

{...}

void DrawRay()
    {
        RaycastHit hitInfo;
        Ray _ray = new Ray(Camera.main.transform.position,
            Camera.main.transform.forward);
        if (Physics.Raycast(_ray, out hitInfo, Mathf.Infinity))
        {
            if (hitInfo.collider.gameObject.CompareTag("SoundSource"))
            {
                //Correct target was hit
                StartCoroutine(_targetManager.HitTarget(true));
            }

            else if (hitInfo.collider.gameObject.CompareTag("Lamp"))
            {
                //Wrong target was hit
                StartCoroutine(_targetManager.HitTarget(false));
            }

            if (hitInfo.collider.gameObject.CompareTag("SoundSource") ||
                hitInfo.collider.gameObject.CompareTag("Lamp"))
            {
                hitInfo.collider.gameObject.SendMessage("TargetFeedback",
                    Color.yellow);
                StringType nameOfObject = new StringType();
                Vector3Type objTransform = new Vector3Type();
                nameOfObject.Value = hitInfo.collider.gameObject.name;
                objTransform.Value =
                    hitInfo.collider.gameObject.transform.position;
                data.AddData(HrtfSwitch.currentSofa, $"Sound Source
                    {_targetManager.indexOfSource}", "Hit Target Name",
                    nameOfObject);
                data.AddData(HrtfSwitch.currentSofa, $"Sound Source
                    {_targetManager.indexOfSource}", "Hit Target Vector",
                    objTransform);
            }
        }
    }
}
```

Listing 5.3: Code which shows how shooting a projectile casts a ray to check for the sound source and save the data

For this class, attached to the weapon object the player holds, a function is defined that draws a ray cast. Ray casts are physics objects that can trace colliders in the Unity environment[14]. In this case, from the code snippet above, the ray is created from the center of the camera view, and out towards the Z-axis. The physics engine checks an infinite amount of distance, for any information retrieved that it stores in the hitInfo variable. This variable is then used to compare the string the object is tagged with, and if it's the correct sound source, it executes the shot registered function accordingly (True if the player hit the sound source, false they did not hit the sound source). To make sure the code isn't checking constantly for irrelevant colliders (such as 3D models in the spaces), the last check makes sure to change the color-coding of the object to yellow to give the player feedback on that a target has been hit, as well as storing the information to the metrics database.

[14] Unity Ray Casts: `https://docs.unity3d.com/ScriptReference/Physics.Raycast.html`

```
public class TargetManager : MonoBehaviour
{

{ ... }

        public IEnumerator HitTarget(bool shot)
        {
            _dataParser.startCounter = false;
            _dataParser.TargetHit = shot ? 1 : 0;
            if (shot) _spawnedSource.Stop();
            StartCoroutine(AddDataDelay(1f));
            foreach (var t in lampObjs)
            {
                t.gameObject.GetComponent<SphereCollider>().enabled = false;
            }

            yield return new WaitForSeconds(3f);
            _waypointMarker.toggleRender = true;
            if (shot)
            {
                _spawnedSource.Stop();
                StartCoroutine(AddScoreHandler.AddScore());
            }
            else
            {
                StartCoroutine(AddScoreHandler.DetractScore());
            }
            foreach (var t in lampObjs)
            {
                switch (t.tag)
                {
                    case "Lamp":
                        t.gameObject.GetComponent<RandomizeColor>().SendMessage("TargetFeedback",
                            Color.red);
                        break;
                    case "SoundSource":
                        t.gameObject.GetComponent<RandomizeColor>().SendMessage("TargetFeedback",
                            Color.green);
                        break;
                }
            }
            SpeedToggle.toggleSpeed = true;
            yield return new WaitForSeconds(5f);
            GetComponent<SpawnHandler>().SendMessage("DespawnObjects");
        }

        IEnumerator AddDataDelay(float _secs)
        {
            yield return new WaitForSeconds(_secs);
            _spawnedObject.SendMessage("AddData");
            yield return new WaitForSeconds(_secs);
            _spawnedObject.SendMessage("SaveDB");
        }
    }
}
```

Listing 5.4: The events that fires off once a target has been hit by the player

In the snippet above, the function call happening inside the DrawRay() function is examined. It's a coroutine function, which executes the code on a frame-by-frame basis, by being able to pause certain executions and resume again[15]. The delay in execution can be seen by the WaitForSeconds return calls. Once the coroutine is active, the time counted on the sound source object is stopped immediately from the DataParser class. In this class (DataParser), the sound source is also registered as a hit depending if the player hit the right target or not.

From here (Line 11) on the metric data is safely added to the database, by using a nested coroutine. There were issues with storing the data properly in the earlier iterations if the code execution happened without using some method to delay the calls. Using a coroutine fixed this unwanted behavior.

Hereafter, the colliders on the targets are immediately switched off using a for each call. This is done to prevent the player from shooting additional targets once their shot was registered.

After a 3 second delay, which gives the player some time to observe their action, the waypoint marker is rendered. Additional executions happen to the UI as well with the score either being added up or detracted for the player. Finally, the targets get color-coded whether to player hit the correct or incorrect target. The cart resumes to normal speed and all of the targets are removed from the Unity scene with a despawn function call.

5.3.3 Serializing the C# objects to a JSON file

```
Public class DataParser: Monobehavior
{
{...}
       private void AddData()
       {
           Vector3Type playerPosition = new Vector3Type();
           playerPosition.Value = playerPos.position;
           Vector3Type _ObjPos = new Vector3Type();
           _ObjPos.Value = transform.parent.position;
           StringType _nameOfParent = new StringType();
           _nameOfParent.Value = nameOfParent;
           FloatType _getCurrentTime = new FloatType();
           _getCurrentTime.Value = getCurrentTime;

           //Target Missed data is added in WeaponSystem.cs
           data.AddData(sofaFile, objectName, "Correct Target Name",
               _nameOfParent);
           data.AddData(sofaFile, objectName, "Correct Target Vector", _ObjPos);
           data.AddData(sofaFile, objectName, "Player Vector", playerPosition);
           data.AddData(sofaFile, objectName, "Time Taken", _getCurrentTime);
       }
```

[15] Unity Manual - Coroutines: `https://docs.unity3d.com/Manual/Coroutines.html`

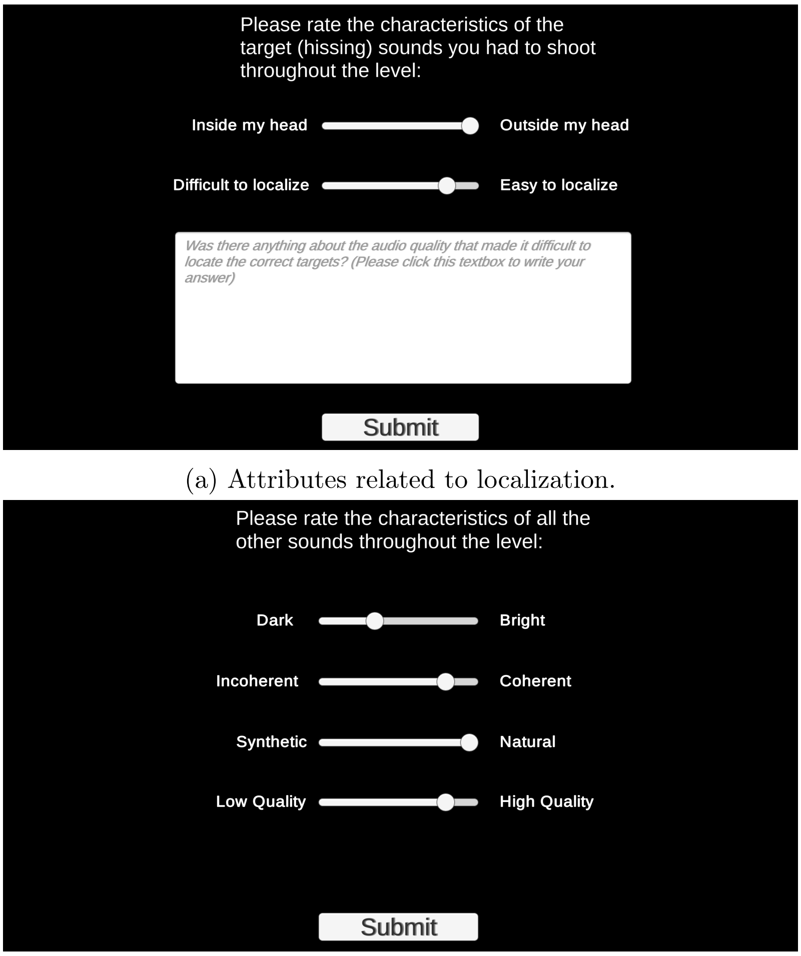
Listing 5.5: Code implementation of how the player metrics were stored for statistical analysis

From previous code snippets, an add data function can be seen called in certain areas of the code implementation. The AddData() function uses the databox library (see section 5.2.2). The Data.Add function creates a table ID, an entry ID, and a value ID, which is great for constructing JSON files since the information can be appended to which HRTF renderer is currently being measured, which sound source is currently being evaluated, and what metrics stored as variables should be named.

This JSON file can then be read and easily exported for further analysis using statistical software, to measure the player's performance. In this case, for the final prototype, the target vector, player vector, and sound source vector are stored to craft an error per participant. Time is saved as well to see if the player reacts fast or not to the sound sources.

For the Likert scale program, the user's written input and ratings are also stored per HRTF renderer.

5.3.4 Likert Scale for rating timbre quality



Please rate the characteristics of the target (hissing) sounds you had to shoot throughout the level:

Was there anything about the audio quality that made it difficult to locate the correct targets? (Please click this textbox to write your answer)

(a) Attributes related to localization.

Please rate the characteristics of all the other sounds throughout the level:

(b) Attributes related to spatial timbre and quality

Figure 5.3: In-game likert scale program where players rate attributes related to spatial localization, timbre, and quality

Figure 5.3 above gives a look into how the likert program of the prototype was implemented.

Each attribute was implemented as a 2D UI Game Object, that has an interactive slider that simulates an anchor related to the attribute. Additionally, a descriptive text box is implemented (see figure 5.3b) in the first screen related to spatial localization, where user can optionally write a description about their experience with the localization task for the current instance. Once the player presses the submit button, the code snippet below runs which saves the user's choices to the json database under the table of the current .sofa file that they are instanced to.

```
foreach (var x in slideObjects)
{
    x.SendMessage("AddAnchorData");
}
yield return new WaitForSeconds(secs);
data.SaveDatabase();
```

Listing 5.6: Code implementation of storing the Likert items

5.3.5 Synthesising the short noise burst

Some participants complained about the sound effect of fireworks used in the earlier iterations of the design. It could make it sound like 'multiple' sound sources playing at once. It was decided for both the second iteration and the final prototype to go for a more dry and monaural signal. To achieve this end, a White Gaussian Noise burst was designed using Matlab. The inspiration came from similar research for sound localization [28].

```
1  fs = 48000;
2  hz = 440;
3  msWinLength = 100;
4  msAudioLength = 351;
5  winLength = msWinLength * fs/1000;
6  w = hann(winLength);
7  t = 0:1/fs:msAudioLength/1000;
8  
9  ySine = sin(2*pi*hz*t);
10 
11 yGaus = awgn(ySine,10);
12 
13 WGN = yGaus - ySine;
14 
15 index = 1;
16 windowSequence = [zeros(1,index-1) w' zeros(1,length(WGN)-index-length(w)+1)];
17 output = WGN.*windowSequence;
18 
19 audiowrite('WGN100ms.wav',output,fs);
```

Listing 5.7: Matlab implementation of the short noise burst used for the targets

Parameters are set up to model both a sinusoid and a window. In this case, a Hann window is used to model the burst effect.

Once the sinusoid is generated, the white gaussian noise is added using the awgn() function. However, we need to subtract the sinusoid for the generated noise model in order to isolate the noise. This is done in line 14 by defining the WGN.

The window sequence is then modeled onto the length of the WGN measured in samples. This is used for the output of the signal in the next line, where an element-wise operation defines the window as the samples. It's important to note that the 350ms sample length is on purpose, to give the noise burst a 250ms pause each time it loops inside of Unity.

The generated signal for 100 miliseconds can be seen below:

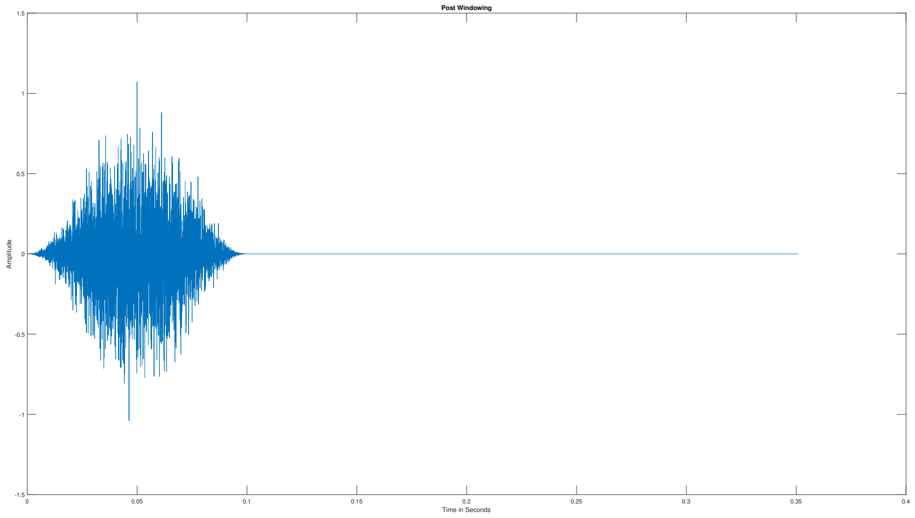
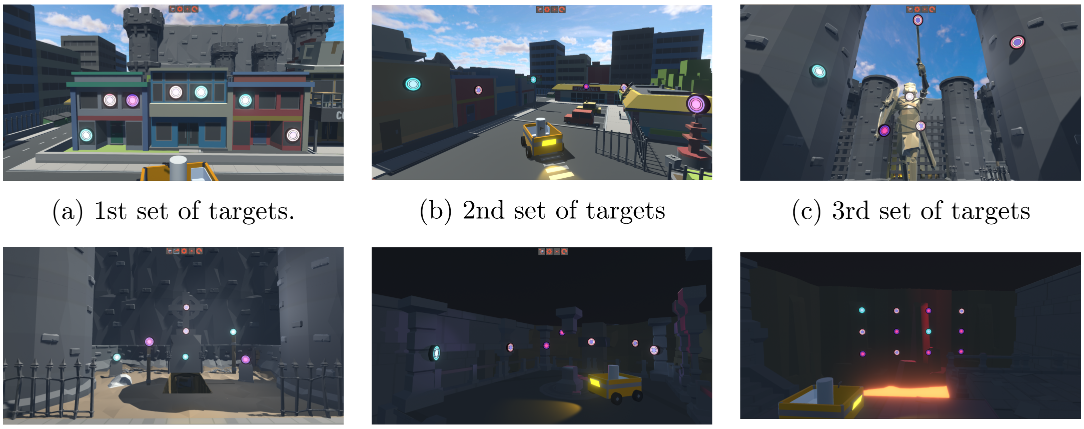


Figure 5.4: Graph of the noise burst at 100ms length and a 250ms pause

5.3.6 Building the environment

For building the environment, the fixed path the player moves on is exploited. Each space has a carefully modeled route that takes the player through all three spaces, with the targets spawning alongside the 3D models to make them blend more into the overall environment. In total there is 6 set of targets that spawns along the player's path. Once the player is done with the 6th target, they start driving backwards and have to shoot all the targets again in a reverse order. The total amount of sets with targets is 12.

Targets that are placed near each other, add some challenge to the localization task. Most notably are the 2nd and 11th set of targets (see figure 5.5b) that are placed in a horizontal circle around the player to test their ability to localize the correct sound source along the azimuth. For the 3rd and 9th set of targets (see figure 5.5c), they are placed vertically on a statue to test the player's ability to localize the correct target along the elevation. For the 6th and 7th set of targets (see figure 5.5f), a balance of both azimuth and elevation are highlighted as a 3x4 grid.



(a) 1st set of targets.

(b) 2nd set of targets

(c) 3rd set of targets

(d) 4th set of targets

(e) 5th set of targets

(f) 6th set of targets

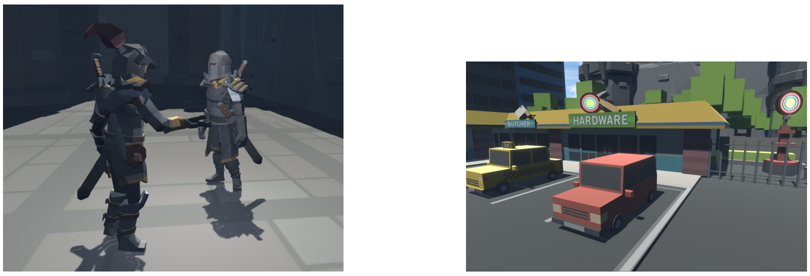
Figure 5.5: All of the target placements implemented in the game. A larger version of the images can be seen in appendix A

Along the fixed path, there are collision triggers scattered throughout. This includes a conversation between two knights that triggers when the player passes by (see figure 5.6a). There's also continuous sounds that plays without any triggers, like background ambience and a car that has the radio turned on which the player passes during the second set of targets. (see figure 5.6b).

Another important trigger along the path is the different parametric reverberation profiles that are stored in the game. A small reflective reverb for the City space, a medium reflective reverb for the Castle space, and a large reflective reverb for the Cave space.

These reverb effects only apply to the sound sources heard around the player, as well as the gunshot from their pistol and the motor engine from the cart they're positioned in. Initially, it was attempted to put the effect onto the short noise burst as well. But since the sound signal of the noise burst is dry, the reverberation did not end up having the desired effect.

The plugin used for the reverberation is the built-in reverb offered by Unity since they need to be more customized for games.



(a) Two knights striking up a conversation with each other the moment the player passes them by in the Castle space.

(b) A car radio that plays music with a low-pass filter applied to make the space of the sound interior.

Figure 5.6: Some of the spatial sound sources throughout that has a visual feedback as well.

5.3.7 Optimizations in scaling the game for lower-end hardware

Since the test had to be conducted remotely, it was important to scale back on the processing required to run the game since the participants would be randomly sampled - meaning not everyone is in a possession of a strong computer. The following optimizations were made to make sure the game could run fine on lower specifications as well,

- Since the sound sources played throughout each space are quite passive, there was no need to model any expensive real-time sound propagation. Instead, it was opted to adjust a parametric reverb profile for each space, until it had believable enough results attained to that space.
- All the sound sources that would be binaurally rendered were imported and exported as mono sound files, instead of the spatial sound engine needing to do this conversion at run time. This gave less processing to the audio threads.
- A graphics option was implemented in the Main Menu, where the user can click a preset of "low quality" to "ultra quality", if the game ran slow on their GPU (especially when running the game with integrated graphic cards). Through this menu, they could tone down the quality to a setting where the frame rate started to get more acceptable.
- Due to the passive nature of the environment also (no dynamic/moving targets except the player), the lighting renderer was changed to be a mixed solution of baked shadow maps for the static objects and real-time shadow maps for the player's cart.

6

Evaluation

This chapter displays the conclusive results of the prototype. Since a lot of data was gathered from all the test participant, each section will try to go through most of the data collected.

6.1 Methodology

Since the prototype was designed to be parallel in terms of testing (to be either conducted at a lab or remotely), it was decided to focus on doing the test remotely and for PC/Mac, because of the rising Covid-19 restrictions during the testing period. To meet that end, questionnaires were developed to try and form a way to use Convergent parallel mixed methods [27], which explains that qualitative and quantitative data collection and analysis are done separately during the same time frame. The results are then brought together for an interpretation using both methods. This approach is also similar to what was achieved during the design chapter, 4.

However, for this final stage of the prototype, the volume of quantitative data is quite expansive so more interesting results can be interpreted in parallel with the qualitative data in how the user expresses their feedback about the quality of the experience (QoE).

6.1.1 Experiment Design

The participants were sampled using convenience sampling [27] whereas if they were interested in joining the experiment, they would be sent a consent form (see appendix C) as well as a guide on how to send in the anthropometric data (see appendix B) needed for their individualized HRTF renderer to be synthesized (see section 4.3.5). Once the data was retrieved and the HRTF renderer was successfully synthesized and verified, a user would be sent a personalized build of the prototype using the same generic HRTF (from the MIT KEMAR dataset [29]) across all participants and their individualized HRTF. Since prior research from section 3.1.1 suggests that the HRTF order had a huge significance on their data, it was also decided that the participants would be split into an A and B grouping, with the A grouping experiencing their individualized HRTF first after having played the non-binaural instance. The B grouping would in return experience the generic HRTF first after the non-binaural instance.

For evaluating the dimension of sound localization and thereby the player's accuracy whether they could locate the sound source or not in each renderer, three vectors were stored each time the player shot a target. The first one being the vector of the player used as a point of origin, the second being the vector of the target they hit, and the last vector being the target containing the sound source. It was possible to calculate the angle difference as an error in both the azimuth and elevation from the target the player shot to the target containing the correct sound source. This means the error would be 0 if the player shot the vector containing the sound source.

To evaluate the spatial quality and timbre of each profile, the test participant would, during gameplay, be introduced to the Likert scale (see figure 5.1), with modified MUSHRA attributes and anchors inspired by previous research in regards to assessing spatial audio quality [6][30]. This would be rated for three renderers, the non-binaural renderer (using a distance-based attenuation curve without an HRTF), a binaural rendered generic HRTF, and lastly a binaural rendered personalized/individual HRTF. Most importantly are the comparisons between the binaural renderers, where the difference in mean-variance can be observed using a Wilcoxon signed-rank test, which is also recommended for statistical analysis of any likes of a MUSHRA evaluation [31].

Lastly, their feedback on the experience as well as their preferred profile would be evaluated using a modified version of the Google Questionaire previously used in the iteration testing (see section 4.2). In this form, background information such as age, sex, experience with video games as well as the model of their headphones was included as well. The remainder of the questionnaire had some questions about the overall experience that the participants could write their answers to.

All data analysis and calculation was done using Jupyter Notebook using a Python kernel [1] with numpy, pandas and scripy stats libraries.

6.2 Participant Information

In total the experiment was sent to 26 people but 4 did not manage to complete it by the assigned deadline. The final sample size ended up being 22 participants instead, with 72.7% being male and 22.7% female. One participant preferred to not specify. The age of the participants ranges from 22 years old to 30 years old. None of the participants reported having any hearing loss, which is crucial information to the experiment. Players were also asked to rate how often they played video games. Looking at figure 6.1 below, the two clusters between participants not playing games that often and participants tending to play games often is even, providing two clusters of participants who are less and more experienced with playing video games.

[1] Project Jupyter: `https://jupyter.org/`

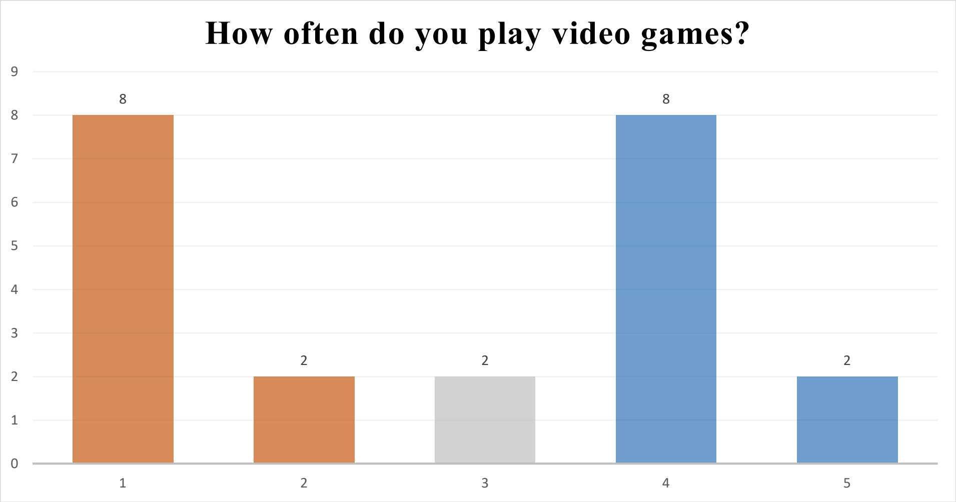


Figure 6.1: A 5-scaled likert where players ranks how often they play video games. 1 = Not at all, 5 = All the time.

6.3 Performance Metrics

As mentioned in the experiment design, the azimuth and elevation errors were calculated as angular differences between two vectors.

The vectors in the JSON database were stored as Cartesian coordinates. The 'Hit Target' and 'Correct Target' (Sound Source) vectors were both subtracted from the 'Player's' vector, their Cartesian coordinates were converted to spherical coordinates, using an arctan2 function call to extract the azimuth and elevation coordinate in radians. The calculation can be see in the python snippet below using numpy.

```python
def vectorToSpherical(v):
    x, y, z = v
    az = np.arctan2(z, x)
    el = np.arctan2(np.sqrt(x**2 + z**2), y)
    return el, az

def processTable(df):
    for i, val in df.iterrows():
        ht = val['Hit Target Vector'] - val['Player Vector']
        ct = val['Correct Target Vector'] - val['Player Vector']
        # convert vectors to spherical
        ht_el, ht_az = vectorToSpherical(ht)
        ct_el, ct_az = vectorToSpherical(ct)
        # store differences
        az_err = np.degrees(np.abs(ht_az - ct_az))
        el_err = np.degrees(np.abs(ht_el - ct_el))
        if az_err > 180:
            az_err = 360 - az_err
        df.loc[i, 'Horizontal Angle'] = az_err
        df.loc[i, 'Vertical Angle'] = el_err

    return df
```

Listing 6.1: Python code that shows how the performance metrics gathered from the JSON database was calculated to produce an error for the Azimuth and Elevation

The radians were calculated in degrees and the difference was stored by subtracting the absolute value from the 'Hit Target' azimuth with the 'Correct Target' azimuth, and the 'Hit Target' elevation with the 'Correct Target' elevation.

	Non-binaural				**Generic HRTF**				**Individual HRTF**			
Participant ID	Time Taken	Azimuth Error	Elevation Error	Targets Hit	Time Taken	Azimuth Error	Elevation Error	Targets Hit	Time Taken	Azimuth Error	Elevation Error	Targets Hit
1	3.00	0.46	3.94	9	1.86	0.40	3.83	10	2.26	0.00	0.74	11
2	11.56	17.14	6.84	3	11.04	5.23	5.95	6	13.17	2.29	4.15	7
3	6.88	3.07	3.99	8	4.94	0.64	4.55	10	7.29	0.00	3.52	10
4	12.14	6.41	5.43	6	18.21	2.92	5.23	6	12.77	3.17	6.67	6
5	5.92	2.89	4.70	6	4.42	2.05	1.87	9	5.14	0.00	0.82	11
6	12.77	6.04	3.78	6	8.46	2.44	2.73	9	8.66	1.96	4.62	9
7	3.27	0.45	3.16	10	3.30	0.96	1.43	10	2.59	0.02	1.51	10
8	6.62	1.75	4.24	7	3.74	0.05	2.84	9	3.27	0.37	2.39	10
9	4.46	0.06	5.17	8	3.64	0.45	2.58	9	4.09	0.04	4.18	9
10	9.62	9.18	10.65	4	7.18	5.21	7.86	7	8.74	8.46	7.37	5
11	12.13	42.48	12.00	2	18.51	10.13	9.52	4	18.49	4.12	4.48	7
12	21.92	12.33	9.11	5	13.31	13.04	2.18	6	14.09	9.97	5.89	4
13	8.28	11.16	2.32	6	11.49	20.92	5.86	7	11.46	1.68	4.54	9
14	12.40	8.06	7.02	5	10.74	7.36	4.94	3	7.47	2.62	3.67	9
15	6.49	6.16	5.52	7	8.09	1.77	9.31	7	9.42	1.61	4.13	7
16	12.86	4.29	4.53	7	12.51	4.76	3.94	6	9.83	3.10	3.41	8
17	6.79	17.37	7.40	6	8.12	2.41	2.91	9	7.95	0.00	0.81	11
18	9.04	11.28	6.76	6	11.25	16.05	6.75	2	11.27	1.10	5.94	7
19	15.59	0.92	3.98	8	12.92	4.69	5.12	7	15.83	4.03	1.98	8
20	10.82	19.43	5.02	5	9.05	7.72	4.15	7	10.22	12.35	8.10	6
21	2.63	43.94	13.27	0	6.75	17.45	6.25	5	8.21	11.04	4.19	5
22	5.53	0.00	0.82	11	4.10	1.37	4.96	8	4.66	1.52	1.72	9
Mean	**9.12**	**10.22**	**5.89**	**6.14**	**8.80**	**5.82**	**4.76**	**7.09**	**8.95**	**3.16**	**3.86**	**8.09**
Std dev	**4.54**	**11.91**	**3.01**	**2.45**	**4.53**	**5.92**	**2.20**	**2.19**	**4.22**	**3.72**	**2.05**	**2.02**

Table 6.1: Table over each participant's accuracy with localizing the sound source by the azimuth and elevation. The table is calculated by a mean of all 12 localizing trials per instance. 0 error means all 12 targets were hit in that category. For Targets Hit, a higher number means more correct targets were hit.

In the table above (6.1), the time taken for each participant to select and shoot a target is quite similar across each instance. However, the mean error for the azimuth tends to be quite high in the non-binaural instance, with a high standard deviation as well suggesting outliers. For the generic HRTF, this azimuth mean has decreased the most. Finally, for the individual HRTF, both the azimuth and elevation starts to have quite a significant low mean error for both angles as well as a standard deviation suggesting much fewer outliers in the data. Most targets were also successfully hit during the binaural instances, with the individual HRTF scoring highest.

In the next subsections, the category between the expert participants and the novice participants has been divided to calculate the error within both groups.

6.3.1 Expert participants

The *expert* participants are defined as such, since in the background information segment of the questionnaire, they both rated themselves as playing games frequently as well as being fully aware of how 3D audio works in games (with one participant being an exception), by rating it high in how they utilize it in their playstyle (either by finding objectives or enemy players in competitive matches)

	Non-binaural				**Generic HRTF**				**Individual HRTF**			
Participant ID	Time Taken	Azimuth Error	Elevation Error	Targets Hit	Time Taken	Azimuth Error	Elevation Error	Targets Hit	Time Taken	Azimuth Error	Elevation Error	Targets Hit
2	11.56	17.14	6.84	3	11.04	5.23	5.95	6	13.17	2.29	4.15	7
3	6.88	3.07	3.99	8	4.94	0.64	4.55	10	7.29	0.00	3.52	10
5	5.92	2.89	4.70	6	4.42	2.05	1.87	9	5.14	0.00	0.82	11
7	3.27	0.45	3.16	10	3.30	0.96	1.43	10	2.59	0.02	1.51	10
8	6.62	1.75	4.24	7	3.74	0.05	2.84	9	3.27	0.37	2.39	10
9	4.46	0.06	5.17	8	3.64	0.45	2.58	9	4.09	0.04	4.18	9
10	9.62	9.18	10.65	4	7.18	5.21	7.86	7	8.74	8.46	7.37	5
17	6.79	17.37	7.40	6	8.12	2.41	2.91	9	7.95	0.00	0.81	11
21	2.63	43.94	13.27	0	6.75	17.45	6.25	5	8.21	11.04	4.19	5
22	5.53	0.00	0.82	11	4.10	1.37	4.96	8	4.66	1.52	1.72	9
Mean	**6.33**	**9.58**	**6.03**	**6.30**	**5.72**	**3.58**	**4.12**	**8.20**	**6.51**	**2.37**	**3.07**	**8.70**
Std dev	**2.57**	**13.09**	**3.49**	**3.13**	**2.37**	**4.94**	**2.01**	**1.60**	**3.03**	**3.81**	**1.93**	**2.15**

Table 6.2: Time, error and successful shots calculated for the expert participants.

Again, a decrease can be observed with the individual HRTF having the lowest error and standard deviation.

6.3.2 Novice participants

Novice participants are the opposite to the expert participants. They rated as having played games very little or not at all and joined the experiment out of pure curiosity.

	Non-binaural				**Generic HRTF**				**Individual HRTF**			
Participant ID	Time Taken	Azimuth Error	Elevation Error	Targets Hit	Time Taken	Azimuth Error	Elevation Error	Targets Hit	Time Taken	Azimuth Error	Elevation Error	Targets Hit
4	12.14	6.41	5.43	6	18.21	2.92	5.23	6	12.77	3.17	6.67	6
11	12.13	42.48	12.00	2	18.51	10.13	9.52	4	18.49	4.12	4.48	7
12	21.92	12.33	9.11	5	13.31	13.04	2.18	6	14.09	9.97	5.89	4
13	8.28	11.16	2.32	6	11.49	20.92	5.86	7	11.46	1.68	4.54	9
14	12.40	8.06	7.02	5	10.74	7.36	4.94	3	7.47	2.62	3.67	9
15	6.49	6.16	5.52	7	8.09	1.77	9.31	7	9.42	1.61	4.13	7
16	12.86	4.29	4.53	7	12.51	4.76	3.94	6	9.83	3.10	3.41	8
18	9.04	11.28	6.76	6	11.25	16.05	6.75	2	11.27	1.10	5.94	7
19	15.59	0.92	3.98	8	12.92	4.69	5.12	7	15.83	4.03	1.98	8
20	10.82	19.43	5.02	5	9.05	7.72	4.15	7	10.22	12.35	8.10	6
Mean	**12.17**	**12.25**	**6.17**	**5.70**	**12.61**	**8.93**	**5.70**	**5.50**	**12.08**	**4.37**	**4.88**	**7.10**
Std dev	**4.08**	**11.17**	**2.61**	**1.55**	**3.26**	**5.83**	**2.19**	**1.75**	**3.12**	**3.56**	**1.69**	**1.45**

Table 6.3: Time, error and successful shots calculated for the novice participants

A big decrease again, in terms of performance from non-binaural to the HRTF renderers. However, the novice participants still have quite a high error still, especially when looking at the individual azimuth scores of the generic HRTF. The individualized HRTF also performed best for the novice players.

6.4 Likert Scale Data

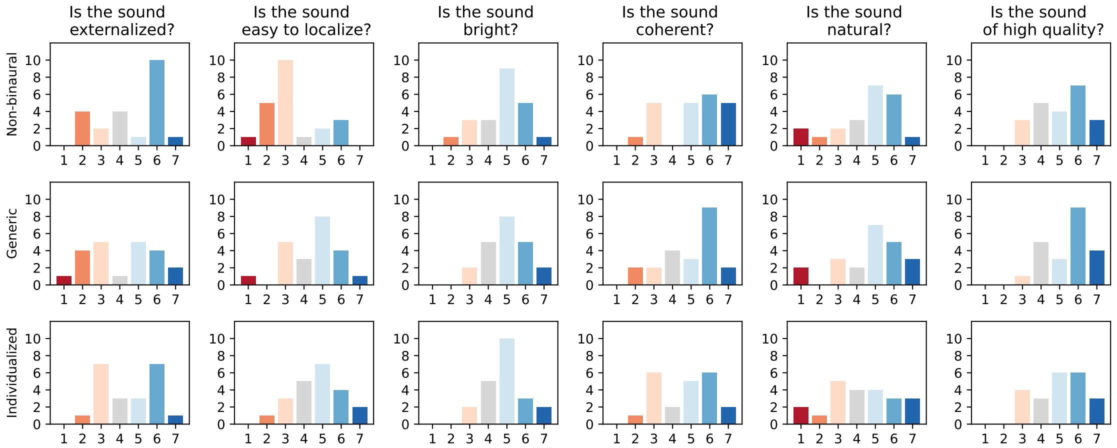


Figure 6.2: Histogram displaying the responses of each attribute across all three instances.

As stated previously in the chapter, the Likert scale was inspired by the attribute approach using MUSHRA anchors to assess audio quality. Rating the scales follows that selecting 1 disagrees with the characteristic of the sound which is presented at 7. In figure 6.2 above, starting from the left histogram the two first items corresponded to localization we see a higher sum of positive scores for the binaural profiles.

For the anchors related to more of the timbre quality of each profile, we see similar distributions across each item.

Generally, the profiles look to be closely distributed. A non-parametric test in form of the Wilcoxon signed-rank test would need to be conducted to better measure the difference between both binaural profiles.

6.4.1 Wilcoxon signed rank test

	pvalue
Questions	
Is the sound externalized?	0.09
Is the sound easy to localize?	0.74
Is the sound bright?	0.86
Is the sound coherent?	0.06
Is the sound natural?	0.02
Is the sound of high quality?	0.01

Table 6.4: Wilcoxon signed rank test performed on each attribute of the likert scale between the generic and individualized HRTF.

The Wilcoxon signed-rank test has been performed in Python using the library, Scipy [2]. We setup the follow hypothesis

[2] Scipy: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wilcoxon.html#r996422d5c98f-4

- H_0: The rating of the spatial attribute is equal between the individualized and generic HRTF.
- H_A: The rating of the spatial attribute is not equal between the individualized and generic HRTF

Here we see the two items, natural sound and sound of high quality having a p value less that 0.05, which means that for these two items, the null hypothesis gets rejected which says the generic HRTF performed better in those two cases since it has the highest mean, whereas for the rest of the scores we accept the null hypothesis.

Level	**Externalisation**	**Localization**	**Brightness**	**Coherence**	**Naturalness**	**Quality**	**Overall**
Generic HRTF	4.14	4.50	5.00	4.95	4.77	5.45	**4.80**
Individualized HRTF	4.50	4.73	4.91	4.68	4.27	5.05	**4.69**
Non-binaural	4.64	3.32	4.77	5.14	4.55	5.09	**4.58**

Table 6.5: Total mean for each spatial attribute

The generic HRTF is performing best with the highest overall mean score.

6.4.2 Responses from the Likert Scale

In the likert scale program, users were also able to describe their experience with the spatial localization throughout the various instances. In this section, some of the written descriptions will be highlighted. The entire table can be viewed in appendix E. Only around the half of the participants opted in to write a description, with a fewer writing a more detailed description for each instance they were subjugated to.

One of these participants that left a description for each instance was participant #5:

- *"It wasn't anything regarding the audio that made it difficult to find out the correct target. It worked pretty well when the targets were not in a vertical line. It was hard to determine which one it was when I had 3 of them in front of me in a vertical line. But maybe it would be the same in real life? Otherwise this works pretty good. I think it would have potential for horror games. You could have nice sounds coming from different angles to spook the player."* - Participant #5 during the non-binaural instance, highlighting the front/back and elevation issues of non-binaural rendered spatial audio.
- *"Again it was mostly the placement. Vertical placement is pretty hard to localize in my opinion."* - Participant #5 during the generic HRTF instance highlighting elevation issues still.
- *"Ok so this one was way easier to localize even in vertical lines. I also had most points by a long shot for this one. It was definitely easier to hear which target was making the sound."* - Participant #5 during the individualized HRTF instance, highlighting that it was easier to locate elevated targets this time around.

Participant #7 also noted that the non-binaural instance was difficult to locate sounds in, and it was still a bit tough during the generic HRTF instance. When they played the individual HRTF they describe that *"It was easier to locate vertically this time"*.

In general, the non-binaural and generic renderers would be described as being difficult to locate elevated targets.

6.5 Feedback on the experience

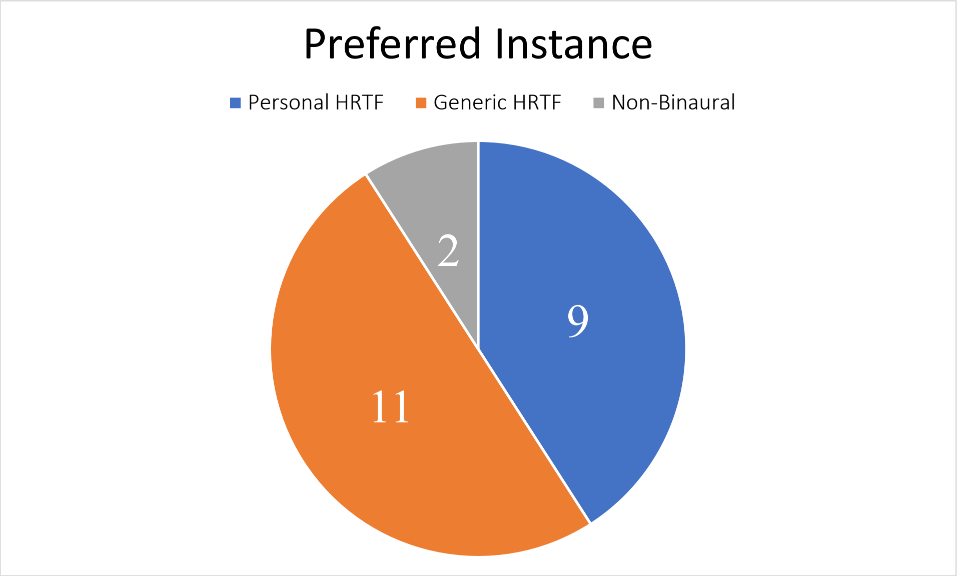


Figure 6.3: Pie chart over the instance the participants preferred. n = 22

The participants were asked, "In which level did you like the sound the most?" where they could pick either the instance of the non-binaural, generic or the individual HRTF renderer. With respect to the participant's sampling order, their preferred pick is visualized in figure 6.3 above.

The participants were hereafter also asked to explain in written words "why the sound was better in this level?"

Some of the responses highlighted explains that,

"It felt the most... natural? Things felt appropriately distant but not too far away, and the sounds blended in a believable way. In level 1 everything felt very distant, especially the hissing targets, and it was almost impossible to locate the targets both vertically and horizontally. In level 2 it felt like the opposite had happened instead, and now it felt like the targets were "inside my head". - Participant #08 (Picked their individual HRTF).

"I think that was the closest where I was able to discern between sounds that are up or down. That was my biggest struggle" - Participant #09 (Picked their individual HRTF).

"even if I preferred the quality of the surrounding sounds in the 2nd level, the target localization seemed clearer to me in the 3rd one. I wouldn´t exclude that, since I had already done the previous levels, this could be due to some "tricks" I thought up (e.g. to wait a little more before giving the answer thus allowing myself to explore the setting a little more)" - Participant #13 (Picked the generic HRTF)

"I don't really know which I liked more - I picked Level 3 as I found it the easiest for

localizing the target hissing sounds. But the soundscape also felt more 'empty' in this level than in the other 2" - Participant #07 (Picked the generic HRTF)

5 participants from the A group (individualized HRTF renderer first), picked their individual HRTF as their preferred instance, while 4 participants from the B group (generic HRTF renderer first) picked their individual HRTF as their preferred instance.

For the question, "Which placement of targets did you find most difficult throughout the entire game? (Coffee shop, parking lot, warrior statue, graveyard, underground altar, lava platform)" the majority of users picked the lava platform, with two picking another area (graveyard and underground altar).

For the last two questions that ask users to mention things they did not like and things they would want to add or change to make it a more fun experience. The participants were generally positive and satisfied about the game, however, the nature of conducting the experiment and playing the same instances over and over again with new profiles was boring for some.

Suggestions the participants made were to perhaps make the environments more exciting, branching paths to drive through and perhaps add more game mechanics as a boss or some enemies you have to fight by localizing sounds around the space.

Not many users reported any glitches or bugs with the game, those that did reported that if they waited in the cart too long, some of the sounds from the underground altar would appear in the lava platform as well for a short while which could be a bit distracting.

7

Discussion

7.1 Assessing the localization task

As gathered from table 6.1, we can see that each renderer has a significant impact on the participant's ability to localize the correct sound source. For example, participant #3, #5 and #17 managed to have no azimuth error with their individual HRTF, whereas #17 had quite a large error during the run of the non-binaural renderer. However, not much attention should be paid to the high error of the non-binaural renderer, since this is the first instance where the participants are introduced to the game. Learning the controls and understanding the task could lead to a lot of targets being missed as well as the poor localization qualities of non-binaural distance based attenuation curves not assisting much either. We can see this effect especially when calculating the high standard deviation which shows that there are quite a few outliers in the non-binaural renderer.

Comparing the generic HRTF and the individualized HRTF, we can also see that generally there is a significant improvement in locating the sound source in the individualized HRTF. The standard deviation is also quite low, so generally each participant had better results. It could perhaps be argued that the participants were able to understand and train better for the task when finally reaching the instance with their individualized HRTF, however, the majority who picked their individualized HRTF were from the sampling group A, who tried the experiment with their individualized HRTF first. The difference in the sampling group A order is only one participant smaller compared to sampling group B.

Interestingly, if we focus on the expert participants, we can see that the individualized HRTF starts to have an improvement again, suggesting that the switch of renderers did cause some a more significant effect on reducing their error on locating the sound sources throughout the level. Comparing the expert participants to the novice participants, we can definitely see that the elevation error has a difference between both groups.

To test whether the individualized HRTF has a larger effect on avid gamers. A focus group would have to be tested with a larger sample size.

Regarding the participants' own answers to describe how they felt the experience of localizing the sound sources, it's hard to draw any generalized conclusion since not many participants opted to describe their experience with the localization task to great length. Some of the participants could definitely tell that there was a problem with locating the "vertical" sound sources that got better when switching to the binaural HRTFs. Especially the individualized HRTF seemed to fix the issue of elevated targets for most.

Opinions here are very subjective and not super consistent, since some participants also noted that the non-binaural renderer sounded great, despite performing better with the

HRTF renderers. More participants would need to give a descriptive answer to start seeing any generalized patterns emerge between both HRTF renderers. Some participants also mentioned after the experiment they didn't feel knowledgeable enough to describe the details of the spatial localization.

In future iterations, perhaps something akin to card sorting could perhaps make it more easy on the user to describe their experience with the spatial localization.

7.2 Assessing the spectral content

When the participants were asked themselves to rate both the localization and the spatial attributes of the sound, it is hard to make out any striking patterns, except the non-binaural instance having the easiness of locating the sound source rated quite low. The Non-binaural instance also ranks quite high on the sound being externalized, however, this could be due to being the first instance that the user is subjugated to. If they have not really tried playing a game with 3D audio, or paid attention to it before, this would put the non-binaural renderer being the first instance at quite an advantage. For the item describing it the sound is easy to localize, we can see that the individualized HRTF scored highest, which also supports the measurements done in the performance metrics.

For the timbre, the anchor of how bright the sound sounded to the users is quite closely rated between the HRTF renderers. When talking to some of the test participants after the experiment, they noted that some of these anchors seemed very vague to them, and they were not used to the terminology in order to describe sounds. The MUSHRA anchors are also designed for more expert listeners of the sound, which could indicate that in this prototype, they failed to properly make the users aware of what they are rating. Suggestions here could be to perhaps for more non-expert listeners to play a sound from each anchor item, where they get a reference on how a sounds bright or dark, high quality or low quality.

In the Wilcoxon signed rank test, rather unexpected results where the generic HRTF is performing best generally. Here the opposite turn out was expected when reading the performance metrics. This could further indicate that the likert scale process failed to clearly communicate each attribute to the participant.

7.3 Validity of the Experiment

Despite the test being successfully conducted remotely for 22 users, that does not go without saying that doing this type of test remotely can cause some issues. Most notably being that each test participant had a different set of headphones and giving very little consistency across headphone equalization. In a worst case scenario, some of the headphones can be of poor quality and not really reflect the frequency spectrum of the HRTF renderers. For the most accurate and valid results, it would be best if the participants had the same equalized set of high quality headphones either for a physical or remote experiment session.

Another issue with doing the experiment remotely was that we lost quite a few participants along the way. Some participants might have felt that sending in the anthropometric data, required a lot of work on their side and interest simply faded. Of biggest concern was that on an operating system version for OSX, the JSON database wasn't saved properly. This happened to one participant that had to be excluded from the experiment due to invalid data. The builds using windows worked stable and without issues due to this also being the main platform the game was built for.

For these reasons above, more accurate and valid data can be acquired for doing this type of experiment physically at a lab using the same type of hardware and peripherals. This also puts the experimenter in charge of making sure that the anthropometric data gathered is accurate. If it were to be done remotely, it is better to assure that all the users have similar computers and headphones, which could be more troublesome.

8

Conclusion

The aim of this thesis was to evaluate whether or not an individualized HRTF could improve player performance and auditory

By doing so, related literature was analyzed and it was found that the realism of spatial audio in games is greatly improved by applying HRTFs. Also, there is plenty of room for further improvements in the field.

A framework on how to assess spatial audio was then investigated. Here it was discovered that HRTFs contain personal measurements, which can give considerable differences in how a person perceives an individualized and generalized HRTF. These differences were both in localization accuracy of sound sources and the quality of the spectral content that could both be measured and evaluated to find contrasts.

In state of the art, research that already attempted to assess spatial audio showed promising results, however there was potential to discover more. Relevant games that incorporated the use of HRTFs deep into their gameplay were analyzed. It was found that better localization accuracy could better guide players with navigational task whereas improved spectral content could enhance immersion.

A design was undertaken proposing a set of requirements to craft a prototype that could answer the research question. The game was designed with the familiarity of a First Person Shooter to make the task of accurately locating sound sources fun and rewarding. The prototype underwent two iterative steps which helped the final prototype improve.

Once the final prototype was ready, a remote experiment with 22 test participants could finally go underway. Performance metrics in accurately locating the sound source was recorded for both a generic and individualized HRTF, as well as the player's experience with their quality of sound.

The results showed that, overall, the participants performed best in the localization task with their individualized HRTF, but the participant's own opinion on which HRTF they preferred themselves differs.

To answer the problem statement,

An interactive game experience can indeed through performance metrics and user ranking find that an individualized HRTF improves the performance for a game that applies a first-person perspective.

However, further improvements can be made. Since some of the participants found the spatial attributes vague, a better communication regarding the spatial attribute rankings to the participants could be considered. For example, a more thorough explanation or an example could have been added.

It could be worth considering to only perform these experiments on a more focused target group that spends a great deal of time playing competitive games where sound localization is key. This target group could more easily see the personal benefits of an individualized HRTF. Furthermore, this would also eliminate any potential selection bias.

Bibliography

[1] M. Beig, B. Kapralos, K. Collins, and P. Mirza-Babaei, "An introduction to spatial sound rendering in virtual environments and games," *The Computer Games Journal*, vol. 8, 12 2019.

[2] S. Serafin, M. Geronazzo, C. Erkut, N. C. Nilsson, and R. Nordahl, "Sonic interactions in virtual reality: State of the art, current challenges, and future directions," *IEEE Computer Graphics and Applications*, vol. 38, no. 2, pp. 31–43, 2018.

[3] M. Gröhn, T. Lokki, and T. Takala, "Comparison of auditory, visual, and audiovisual navigation in a 3d space," *TAP*, vol. 2, pp. 564–570, 10 2005.

[4] M. Beig, B. Kapralos, K. Collins, and P. Mirza-Babaei, "An introduction to spatial sound rendering in virtual environments and games," *The Computer Games Journal*, vol. 8, 12 2019.

[5] C. Cheng and G. Wakefield, "Introduction to head-related transfer functions (hrtfs): Representations of hrtfs in time, frequency, and space," *AES: Journal of the Audio Engineering Society*, vol. 49, pp. 231–249, 04 2001.

[6] C. Armstrong, L. Thresh, D. Murphy, and G. Kearney, "A perceptual evaluation of individual and non-individual hrtfs : a case study of the sadie ii database," *Applied Sciences*, vol. 8, p. 2029, 10 2018.

[7] M. F. Møller, Henrik Sørensen, C. B. Jensen, and D. Hammershøi, "binaural technique: do we need individual recordings?" *journal of the audio engineering society*, vol. 44, no. 6, pp. 451–469, june 1996.

[8] E. Wenzel, M. Arruda, D. Kistler, and F. Wightman, "Localization using nonindividualized head-related transfer functions," *The Journal of the Acoustical Society of America*, vol. 94, pp. 111–23, 08 1993.

[9] R. Nicol, L. Gros, C. Colomes, M. Noisternig, O. Warusfel, H. Bahu, B. Katz, and L. Simon, "A roadmap for assessing the quality of experience of 3d audio binaural rendering," *Proceedings of the EAA Joint Symposium on Auralization and Ambisonics 2014*, 04 2014.

[10] J. Berg and F. Rumsey, "correlation between emotive, descriptive and naturalness attributes in subjective data relating to spatial sound reproduction," *journal of the audio engineering society*, september 2000.

[11] D. Poirier-Quinot and B. F. Katz, "impact of hrtf individualization on player performance in a vr shooter game ii," *journal of the audio engineering society*, august 2018.

[12] B. Katz and G. Parseihian, "Perceptually based head-related transfer function database optimization," *The Journal of the Acoustical Society of America*, vol. 131, pp. EL99–105, 02 2012.

[13] A. Andreopoulou and B. Katz, "Subjective hrtf evaluations for obtaining global similarity metrics of assessors and assessees," *Journal on Multimodal User Interfaces*, vol. 10, 03 2016.

[14] O. Rummukainen, T. Robotham, S. J. Schlecht, A. Plinge, J. Herre, and E. A. P. Habels, "audio quality evaluation in virtual reality: multiple stimulus ranking with behavior tracking," *journal of the audio engineering society*, august 2018.

[15] R. Shukla, R. Stewart, A. Roginska, and M. Sandler, "user selection of optimal hrtf sets via holistic comparative evaluation," *journal of the audio engineering society*, august 2018.

[16] A. Cockburn, *Agile Software Development*, 01 2002.

[17] T. Fullerton, *Game design workshop: a playcentric approach to creating innovative games.* CRC Press, 2019.

[18] A. Westerberg and H. Schoenau-Fog, "Categorizing video game audio: An exploration of auditory-psychological effects," in *Proceedings of the 19th International Academic Mindtrek Conference*, ser. AcademicMindTrek '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 47–54. [Online]. Available: https://doi.org/10.1145/2818187.2818295

[19] H. Hu, L. Zhou, H. Ma, and Z. Wu, "Hrtf personalization based on artificial neural network in individual virtual auditory space," *Applied Acoustics*, vol. 69, pp. 163–172, 02 2008.

[20] G. W. Lee and H. Kim, "Personalized hrtf modeling based on deep neural network using anthropometric measurements and images of the ear," *Applied Sciences*, vol. 8, p. 2180, 11 2018.

[21] M. Zhang, X. Wu, and T. Qu, "Individual distance-dependent hrtfs modeling through a few anthropometric measurements," 05 2020, pp. 401–405.

[22] R. Miccini and S. Spagnol, "HRTF individualization using deep learning," in *Proc. 2020 IEEE Conf. Virtual Reality and 3D User Interfaces Work. (VRW 2020)*, Athens, GA, USA, Mar. 2020, pp. 390–395.

[23] S. Spagnol, K. B. Purkhús, S. K. Björnsson, and R. Unnthórsson, "The Viking HRTF dataset," in *Proc. 16th Int. Conf. Sound and Music Computing (SMC 2019)*, Malaga, Spain, May 2019, pp. 55–60.

[24] S. Spagnol, "HRTF selection by anthropometric regression for improving horizontal localization accuracy," *IEEE Signal Process. Lett.*, vol. 27, pp. 590–594, Apr. 2020.

[25] M. G. Onofrei, R. Miccini, R. Unnthórsson, S. Serafin, and S. Spagnol, "3D ear shape as an estimator of HRTF notch frequency," in *Proc. 17th Int. Conf. Sound and Music Computing (SMC 2020)*, Torino, Italy, Jun. 2020, pp. 131–137.

[26] K. Watanabe, Y. Iwaya, Y. Suzuki, S. Takane, and S. Sato, "Dataset of head-related transfer functions measured with a circular loudspeaker array," *Acoustical Science and Technology*, vol. 35, no. 3, pp. 159–165, 2014.

[27] T. Bjørner, *Qualitative Methods for Consumer Research: The Value of the Qualitative Approach in Theory and Practice.* Gyldendal Akademisk, 2016. [Online]. Available: https://www.amazon.com/Qualitative-Methods-Consumer-Research-Approach/dp/8741258533?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=8741258533

[28] S. Spagnol, R. Hoffmann, F. Avanzini, and A. Kristjánsson, "Effects of stimulus order on auditory distance discrimination of virtual nearby sound sources," *J. Acoust. Soc. Am.*, vol. 141, no. 4, pp. EL375–EL380, April 2017.

[29] B. Gardner and K. Martin, "Hrtf measurements of a kemar dummy-head microphone," MIT Media Lab Perceptual Computing, Tech. Rep., 1994.

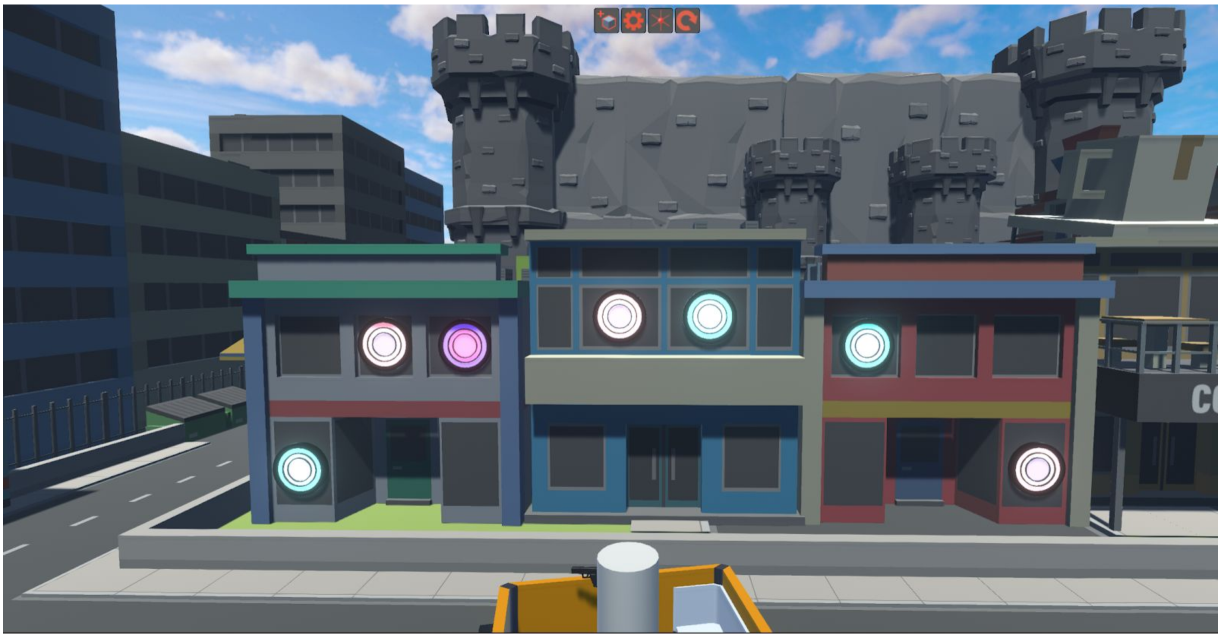
[30] S. Le Bagousse, M. Paquier, C. Colomes, and S. Moulin, "sound quality evaluation based on attributes - application to binaural contents," *journal of the audio engineering society*, october 2011.

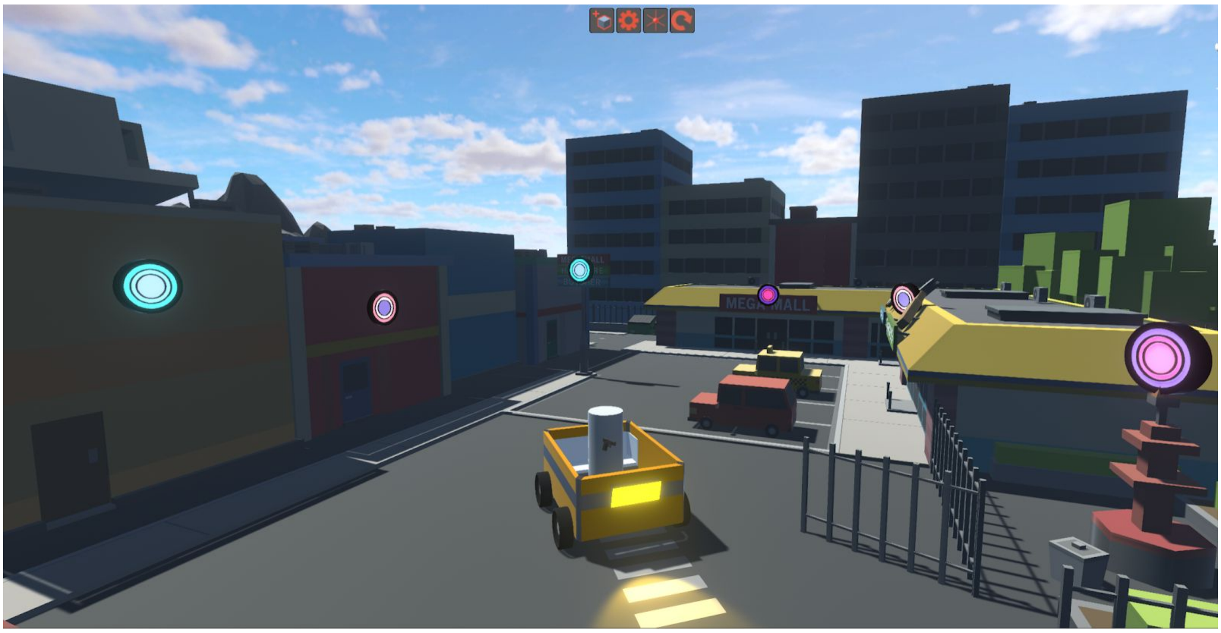
[31] C. Mendonça and S. Delikaris-Manias, "statistical tests with mushra data," *journal of the audio engineering society*, may 2018.

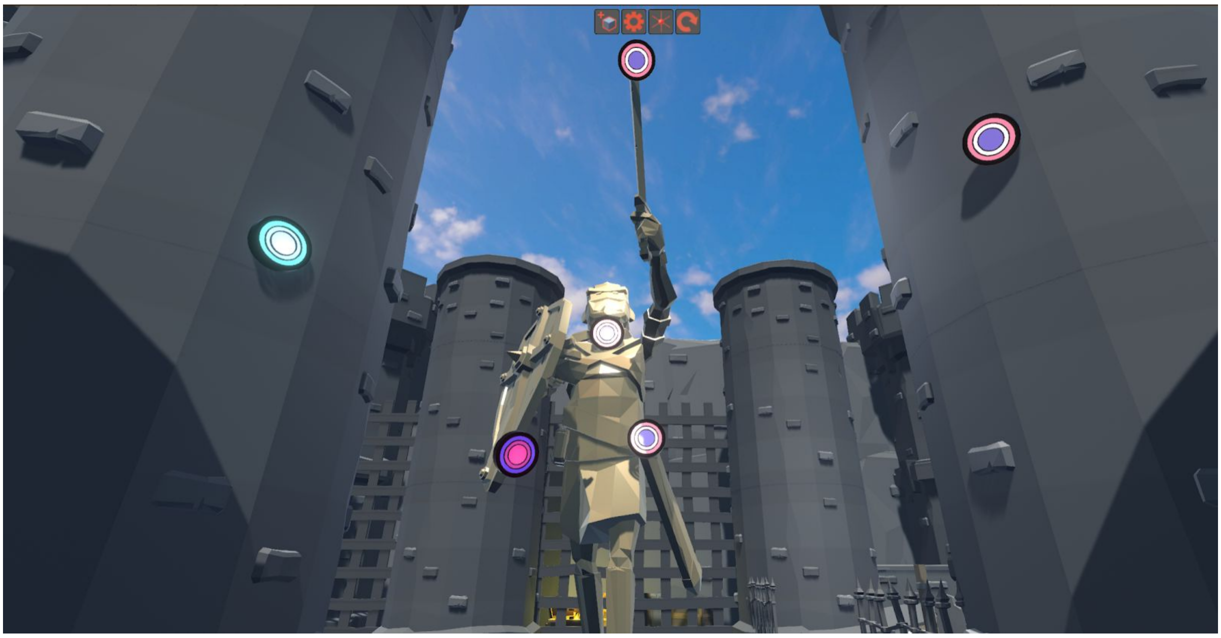
9

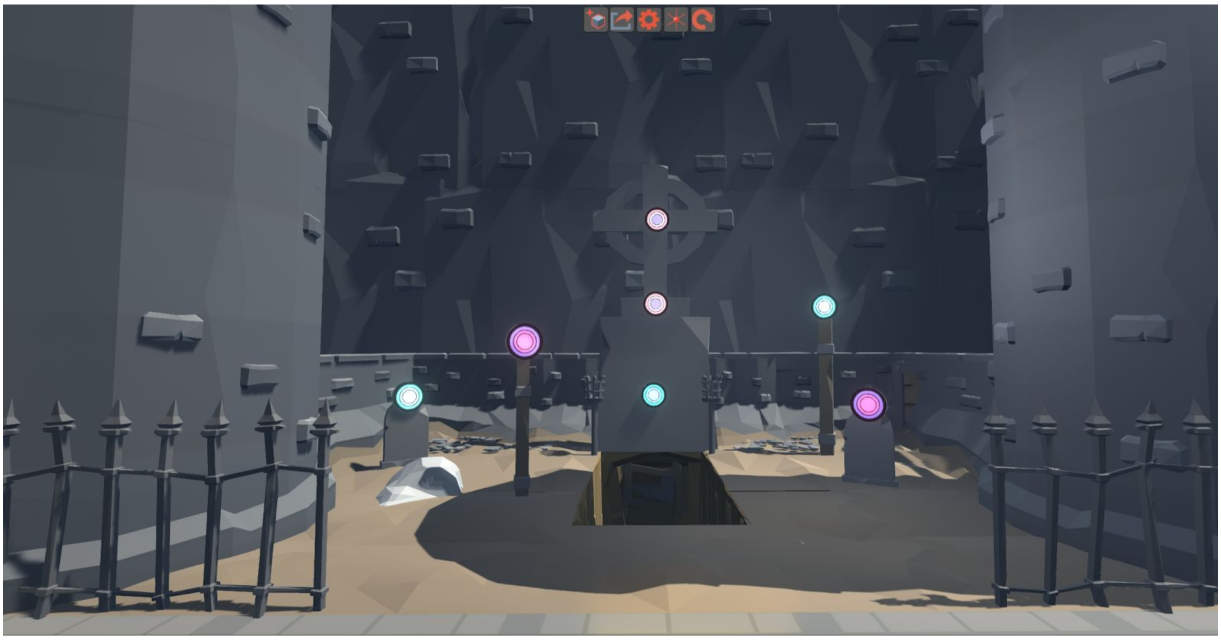
Appendices

Appendix A: Sets of Targets - Large Version


CO


MEGA MALL





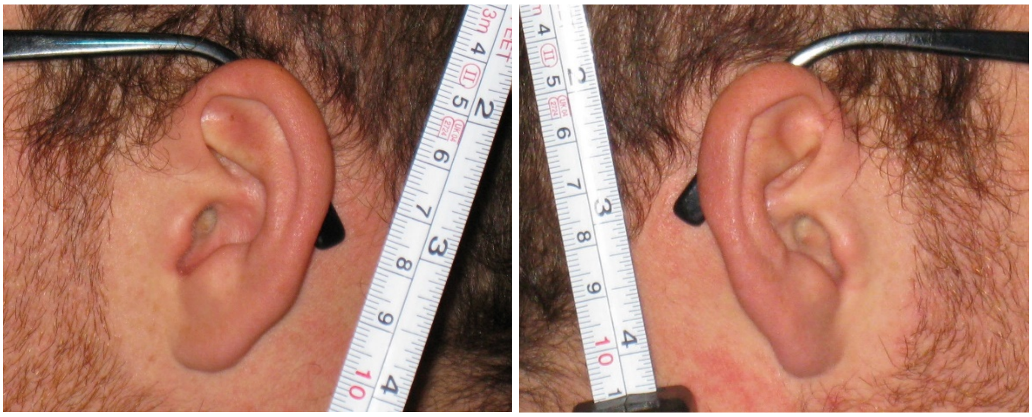
Appendix B: Anthropometric Data Guide

INSTRUCTIONS FOR COLLECTING YOUR ANTHROPOMETRIC DATA

In order to set up your customized 3D audio system, we need the following data from you ASAP.

Ear pictures:

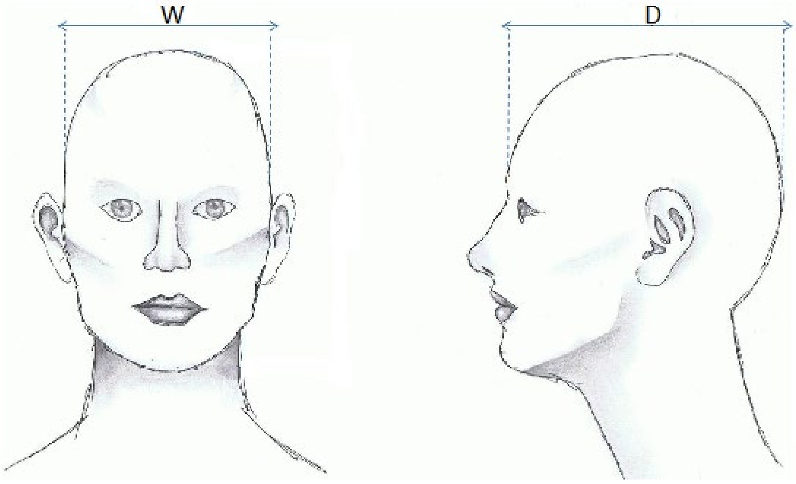
Please send us two side pictures of your ears, one for your left ear and one for your right ear. We need high quality pictures so please use the highest quality camera you have in good lighting conditions. As it might prove difficult to control the camera's angle of view, you can ask someone close to you to take the pictures.



As in the above pictures, we ask you to place an absolute reference (such as a ruler) behind the ear, so that we are able to get a very good estimate of the size of your ear. Taking earrings/glasses off would be a plus!

Head size measurements:

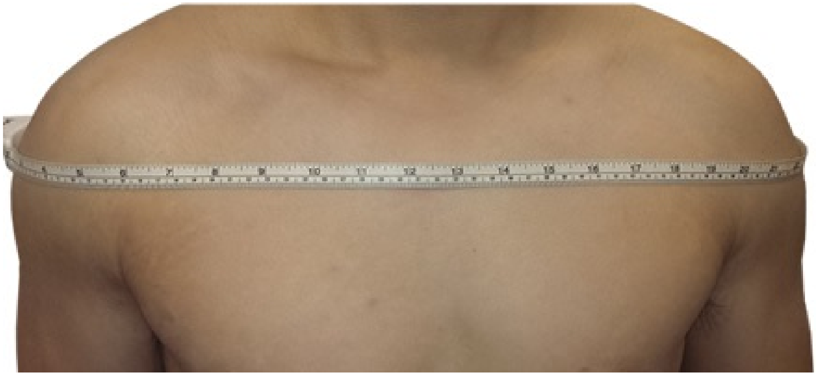
Please send us an estimate of the width (W) and depth (D) of your head in centimetres with one decimal place (e.g., W = 15.3 cm, D = 19.5 cm). W and D are defined according to the following representation:



Therefore W is the distance between your left and right temples, while D is the maximum distance between your forehead and back head. You can estimate these two measurements with a sliding caliper, if you have one, or by simple tricks. For instance, you can place two fingertips on the two reference points, move your head away while keeping the other parts of your body still, and ask someone close to you to measure the distance between your fingertips with a ruler.

Shoulder circumference:

Please send us an estimate of your shoulder circumference (SC) in centimetres with one decimal place (e.g., SC = 114.2 cm). A very simple way to get SC is to use a soft measuring tape, like in the following picture.



Simply ask someone close to you to take the tape, wrap it around your chest keeping it snug, and return back to the starting point. We recommend that you stand still, up straight, and stay completely relaxed to get the right measurement.

Appendix C: Consent Form (English)

Informed Consent to Participation in a Scientific Research Project

Title of the research project: IT'S A DIVE

Declaration by the Volunteer

I have received information about the research project both in writing and orally, and I have sufficient knowledge of the objective, method, advantages and disadvantages to confirm my participation.

I know that participation is voluntary and that I can always withdraw my consent without losing my present or future rights to treatment.

I hereby give my consent to participation in the research project and confirm that I have received a copy of this form and of all written information for my own use.

Name of the Volunteer: ____

Date: _____________ Signature: ____

Would you like to be informed of the results of the research project and of the consequences for you, if any?

Yes_____ No _____ (tick the appropriate field)

Declaration by the Person giving Information

I hereby declare that the Volunteer has received information both in writing and orally about the research project.

I believe that the information given is sufficient for making a decision on participation in the research project.

Name of the person giving the information: ____

Date: _____________ Signature: ____

Project identification: (e.g. project ID of the Committee, EudraCT No., version No./date etc.)

____

Standard declaration of consent issued by Den Nationale Videnskabsetiske Komité, December 2011.
Translation into English made by Center for Sensory-Motor Interaction, Aalborg University

Appendix D: Experiment Information Sheet

IT’S A DIVE – INFORMATION SHEET

Thank you for your interest in our research! Please take time to read the following information carefully.

If anything is unclear or if you would like more information, please ask.

What is the project about?

IT’S A DIVE is a European project based in Denmark whose aim is to develop original techniques for realistic headphone listening. By the analysis of ear shapes, we are developing technologies that reproduce through headphones our own and unique way of hearing space, and in particular the direction of virtual sound sources in the space around us. Thanks to the IT’S A DIVE technology, you will be able to perceive virtual sounds as if they were real, allowing you to “dive deep” into what you are hearing. The IT’S A DIVE research program is expected to represent an innovative breakthrough for a significant number of applications, including for instance computer games and navigation aids for the blind.

What will it involve for me?

The present study includes a series of virtual acoustic localization tasks in a game running on a computer. You will be playing a game with a simple interface and control scheme using a keyboard and mouse peripheral and listening to spatialized sounds via headphones. Your task is to recognize the direction from where you think the sound came, by aiming the in-game camera at the source and confirming your selection with a click. Participation in the study typically takes half an hour.

We will collect the following data:

(a) *prior to the experiment*: pictures of your ears collected through a standard camera and a few simple parameters of your head and torso (head width, depth, and shoulder circumference);

(b) *during the experiment*: acoustic localization data. These data will include measures of how well you localize virtual sounds in the 3D space around you. This will mainly depend on the quality of the presented virtual sounds and their fit to your own ears, therefore they do not represent a measure of your localization ability;

(c) *after the experiment*: questionnaire data. These data will report your answer to questions about the realism of the presented sounds as well as a short list of demographic questions (sex and age). No sensitive personal questions will be asked (e.g. political opinions, religious conviction, sexual orientation).

Only data (b) and (c) will be stored after the end of your participation.

What are the risks associated with this project?

You will not be exposed to excess sound levels. Average levels will not exceed 80 dB and peak levels will not exceed 120 dB, as prescribed by national health regulations. We will ensure that you can take breaks when you need to and that all work will take place in a safe and suitable venue. No deception is involved, and the study involves no more than minimal risk to participants (i.e., the level of risk encountered in daily life).

What are the benefits of taking part?

You will have the chance to obtain for free a customized 3D audio software for your own use.

Confidentiality and data retention

The results of this study may be published in scientific journals, presented at academic conferences, or used for teaching purposes. Anything you say will only be attributed to you with your permission; otherwise the information will be reported in such a way as to make direct association with yourself impossible. The resulting data will be coded and stored in such a way as to make it impossible to identify them directly with any individual (they will be organized by number rather than by name). Anonymized data will be retained indefinitely and made available on a public database.

Right to withdraw

Your participation is entirely voluntary and you may choose to decline to answer any question or to withdraw at any point and for any reason from the project without explanation.

Contact information

If you have concerns or questions about this study, please contact Dr. Simone Spagnol, Department of Architecture, Design & Media Technology, Aalborg University Copenhagen. E-mail: ssp@create.aau.dk; telephone: +45 9356 2408, address: A.C. Meyers Vænge 15, 2450 København SV.

Appendix E: Descriptive respones from the in-game likert scale

Participant ID	Description	Instance
1	Not in quality - maybe depending on emission of high frequencies.. otherwise a bit difficult to decide on elevation/height of sound	Non-binaural
2	The background noise made it harder to localize sounds.	Non-binaural
3	I felt mostly like I could just not tell the difference in the vertical axis, had no clue if it was up or down in the same line. Also, the volume test in the start of the game was not representative for the game volume once I got to the shooting. The music in the start was loud enough on a very low volume, but in the game I had to bump the volume way up	Non-binaural
5	It wasn't anything regarding the audio that made it difficult to find out the correct target. It worked pretty well when the targets were not in a vertical line. It was hard to determine which one it was when I had 3 of them in front of me in a vertical line. But maybe it would be the same in real life? Otherwise this works pretty good. I think it would have potential for horror games. You could have nice sounds coming from different angles to spook the player.	Non-binaural
5	Again it was mostly the placement. Vertical placement is pretty hard to localize in my opinion.	Generic
5	Ok so this one was way easier to localize even in vertical lines. I also had most points by a long shot for this one. It was definitely easier to hear which target was making the sound.	Individualized
6	Closer the targets would appear to each other the harder it was to localize. Turning around helped sometimes, but not when the targets would rest close to each other.	Non-binaural
7	It is difficult to locate sounds vertically	Non-binaural
7	It didn't feel as precise as level 1	Generic
7	It was easier to locate vertically this time	Individualized
8	The rumbling from the cart muffled the soundstage a bit.	Non-binaural
8	It was a lot harder to locate targets vertically on this run compared to the previous one. The hissing overall was less distinguishable from the rest of the sounds, however, overall it sounded more pleasing to me than the last run did.	Generic
10	The audio quality was good	Non-binaural
14	no	Non-binaural
14	no	Generic
14	no	Individualized
15	When I turned the audio high (at the end) I got all targets right. BEfore it was too low to notice correctly the targets for me, althouth yet it was a pleasing volume	Non-binaural
15	When there are targets that are in the same direction, same distance, but different height, I-m confused	Individualized
17	I had some problems localizating the vertically space, but i think it's normal	Non-binaural
18	On the last one I accidentaly shot the wrong one, actually I was deciding between the right one and the nearest.	Non-binaural
19	it was difficult almost when i needed to localize the hizzing sound towards up and down, while it was easier to localize it when it was more shifted towards left or right	Non-binaural
19	Same as before: more difficult to localize when i needed to choose whether the sound was more up or down	Generic
19	Same as before	Individualized
21	the finals target in the castel are very difficult. Too close to each other	Generic
22	No	Non-binaural
22	No	Generic
22	No	Individualized