Bachelor’s Thesis

Information and Communications Technology

2020

Axel Lindberg

# IMMERSIVE SPATIAL AUDIO VERSUS BUILT-IN SPATIALIZATION SOLUTION AMONG VR USERS

## – Utilizing A/B Testing to Determine User Preferences

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR’S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

INFORMATION AND COMMUNICATIONS TECHNOLOGY

2020 | 55 PAGES

Axel Lindberg

# IMMERSIVE SPATIAL AUDIO VERSUS BUILT-IN SPATIALIZATION SOLUTION AMONG VR USERS

## - Utilizing A/B Testing to Determine User Preferences

This work involves utilizing two existing VR scenarios to compare the spatial audio features provided by the spatialization solution built into the Unity game engine and the Resonance Audio spatializer plugin. The aim was to determine the preferences of experienced VR users as well as regular consumers through the comparison. Everything is based on the findings gathered through A/B testing. The thesis also tries to find out the significance and usefulness of immersive spatial audio. In addition to these, it tries to find out if a plugin like Resonance Audio is worth the effort and whether it is a profitable investment for game companies focused on VR games and simulations. The purpose of this work is to serve a potential guideline for game companies and developers alike interested in the topic by providing comprehensive experiences on the subject.

An A/B test where test subjects had to listen to sounds one after another was conducted in both scenarios. Each sound was played twice and the participants had to choose between the two choices. The choices used the spatialization solutions provided by both Unity and Resonance Audio and the participants had to choose which one they preferred more. Overall the results were in line with the expectations and matched the amount of time and effort spent on work. The results also gave grounds to conclude that increasing the immersion of the spatial audio provided by Resonance Audio would significantly increase the difference in preferences between the two spatialization solutions.

KEYWORDS:

Spatial audio, virtual reality, sound processing, immersion, Unity, Resonance Audio

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

TIETO- JA VIESTINTÄTEKNIIKKA

2020 | 55 SIVUA

Axel Lindberg

# IMMERSIIVINEN TILAÄÄNI VASTAAN SISÄÄNRAKENNETTU SPATIALISOINTIRATKAISU VR-KÄYTTÄJIEN KESKUUDESSA

## - A/B-testauksen hyödyntäminen käyttäjien mieltymysten määrittämiseksi

Tähän työhön liittyy kahden olemassa olevan VR-skenaarion hyötykäyttö Unity-pelimoottoriin sisäänrakennetun spatialisointiratkaisun ja Resonance Audio -liitännäisen tarjoamien tilaääniominaisuuksien vertailemiseksi. Tarkoituksena on selvittää kokeneiden VR-käyttäjien kuin myös tavallisten kuluttajien mieltymyksiä kyseisen vertailun kautta. Kaikki perustuu A/B-testauksessa ilmi käyneisiin löydöksiin. Kyseisten löydösten kautta halutaan myös selvittää immersiivisen tilaäänen merkitys ja hyödyllisyys. Näiden lisäksi halutaan selvittää, onko sellainen liitännäinen, kuten Resonance Audio, vaivan arvoinen, ja onko se kannattava sijoitus VR-peleihin ja -simulaatioihin keskittyneiden peliyritysten kannalta. Tämän työn tarkoitus on olla mahdollinen suuntaa-antava ohje kaikille aiheesta kiinnostuneille peliyrityksille ja pelinkehittäjille, tarjoten kattavia kokemuksia aiheesta.

Molemmissa skenaarioissa suoritettiin A/B-testi, jossa koehenkilöiden oli kuunneltava ääniä peräkkäin. Jokainen ääni toistettiin kahdesti ja testaajien oli valittava näiden kahden vaihtoehdon välillä. Valinnoissa käytettiin sekä Unity:n että Resonance Audio:n tarjoamia spatialisointiratkaisuja, ja testaajien oli valittava kumpaa ratkaisua he suosivat. Kaiken kaikkiaan tulokset vastasivat odotuksia ja työhön käytettyä aikaa ja vaivaa. Tulokset antoivat myös syyn olettaa, että Resonance Audio:n tarjoaman tilaäänen immersion kasvattaminen lisäisi merkittävästi mieltymyseroja.

ASIASANAT:

Tilaääni, virtuaalitodellisuus, äänenkäsittely, Unity, Resonance Audio

# CONTENT

# FIGURES

# PICTURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| HRTF | Head-Related Transfer Function |
| ILD | Interaural Level Difference |
| ITD | Interaural Time Difference |
| TGL | Turku Game Lab |
| TUAS | Turku University of Applied Sciences |
| VR | Virtual Reality |

# 1 INTRODUCTION

Virtual reality (VR) has been on the rise for a few years now and this year we've seen one the first big-name titles for VR being released by Valve called Half-Life: Alyx, where spatial audio played a huge role. Usually, the purpose of VR is to provide the most realistic impression of a situation to which the player or user is to be presented. Therefore, audio solutions, in principle, should be as good as they can. The goal of these solutions would be to mimic real-life listening experiences. Although, virtual reality, where users strap on a headset and trackers that orient the body within a simulated world, is a medium that hasn't quite taken off yet among regular consumers. Perhaps it is due to the fact that some companies don't want to invest in providing a more realistic and immersive listening experience to their users. It may also be because they simply do not want to spend time on it and only focus on better graphics and interesting gameplay. Perhaps immersive spatial audio is feared to require a certain level of expertise to maintain it, let alone implement it. While not denying the importance of graphics, gimmicks, the overall stability of refresh rates and frame rates, or the gameplay itself, the realism of the spatial audio could be a major factor in the entirety of virtual reality.

Spatial audio, or 3D audio, is a set of techniques that make it possible to create an impression that sounds are coming from different locations around a listener. In other words, spatial audio adds to the sense of space within a virtual environment. While increasing the effectiveness of spatial audio is not a new concept, there still seem to be individuals out there who believe that investing in methods capable of doing that are not profitable in the long run and that it's best to go from where the fence is the lowest. If in virtual reality, whether they are games or something else, the goal is most often to provide the users with an overall immersive experience, why shouldn't the audio match everything else in the level of immersion?

Immersion is a term that will be used often in this thesis because of how important a factor it is in VR. In this VR context, immersion is something that can be more or less measured based on user experience. So why settle for a soundscape that provides feedback only roughly to match a situation in which the player or user is placed, when the listening experience could be nearly identical to what the situation would be in real life?

The most prominent and accessible game engines, such as Unity and Unreal, provide audio features with somewhat extensive customization settings and parameters. There are also plugins and middleware that provide an even larger variety of audio features for developers to fiddle around with. When using these features, it is important to know how much you need them and what parameters should be used. However, how does a beginner know what values to put if getting them right is so important in the first place? Fortunately for us, multiple research findings on spatial audio and its use present values to achieve optimal sound directivity and immersion despite the virtual environment. Obstacles and possible sound-reflecting surfaces are a separate issue that needs to be approached from a different perspective and are not part of this work.

In this work, Unity game engine is the platform of choice, which also happens to be the same engine that the game technologies side of Turku University of Applied Sciences is using to teach game development. Following in the footsteps of previous theses related to similar topics conducted under the wing of Turku Game Lab, the Resonance Audio spatializer plugin was chosen to enable immersive spatial audio. Coincidentally, Resonance Audio is free software that is available for both previously mentioned game engines, Unity and Unreal.

In this thesis, we had the opportunity to use two scenarios from a VR project called Simulandia. Simulandia is a virtual learning environment intended to train workers for tasks in many different fields. It provided the ideal environments to compare audio solutions and to perform an A/B test with industry experts and experienced VR users at the same time. The participants mostly consisted of employees working in the industry, other students studying game development, and laboratory engineers working at Turku Game Lab. Some of the participants were also ordinary consumers who, on average, did not have much experience with VR. A subtle A/B test was created for both scenarios, in which the participants stay in place and listen to sounds tailored for the scenarios. Each sound is played twice because it is necessary to see which sound the test takers prefer more. The first sound uses features provided by the Resonance Audio spatializer plugin, while the second uses features built into Unity. The sounds are played in a predetermined order so that the A/B test is the same for each participant, but the participants themselves are not told which sound is which.

In summary, the goal of this thesis is to provide answers to whether it is worthwhile to invest in advanced audio solutions and to provide the author’s own experiences with simple implementations of this technology. The work aims to prove the usefulness of immersive spatial audio by comparing the preferences of VR users between two audio solutions, Unity’s built-in solution and the solution provided by Resonance Audio. However, before we can latch onto the work and A/B testing itself, we have to go through certain viable data that differentiates spatial audio formats and techniques from each other and proves the existence of immersive spatial audio.

The next section of this thesis covers the theoretical background of spatial audio and its use while providing information about both Unity’s and Resonance Audio’s spatialization solutions. The third section explains how to implement both spatialization solutions to a Unity project and shortly goes through some customized programming related to Simulandia. The fourth section explains everything related to the chosen testing method and the sixth section goes through the results attained through testing. The final section shows the conclusion.

# 2 THEORETICAL BACKGROUND

## 2.1 Spatial Audio

Spatial audio is about creating a three-dimensional hearing experience. Just as having two eyes in slightly different positions allows us to perceive depth, so the fact that the ears are placed on either side of the head enables us to benefit from stereophonic, or three-dimensional, hearing [10]. Applications using spatial audio include augmented reality, virtual reality, mixed reality, listening to music, and watching videos and movies. In general, the term spatial audio is used to broadly mean sound that is not mono, wherein headphones one can hear the same thing on both ears. Admittedly, spatial audio is not a very unambiguous concept because there are a handful of different formats available for a variety of different purposes. All in all, there is a certain range of complexity to spatial audio. To reduce the potential burden of ignorance towards the subject, we must first go through the spatial audio formats and a little bit about their usage.

### 2.1.1 Formats

**Channel-Based Audio**

The most common spatial audio format that gives listeners at least some sense of space is stereophonic (stereo) sound. Unlike monaural (mono) sounds, stereophonic sounds can be localized anywhere on a straight path between two speakers. Stereo speakers consist of two channels, left and right. Then there is multichannel audio, or more commonly known as surround sound, which can be considered as an extension to the stereo speaker array. The most notable surround sound formats are 5.1, a six-channel format, and 7.1, an eight-channel format. The channels can be thought of as speakers. For example, 5.1 surround sound speakers usually include two front speakers, two side speakers, a center speaker, and a subwoofer. Together, they form a six-channel speaker layout.

In multichannel audio, sounds can now be heard from any direction. However, generally, surround sound speakers in most homes are more or less positioned in the horizontal plane, meaning that the sense of elevation is still missing. Another constraint of surround sound is precisely the fact that it is tied to specific speakers. A surround sound mix is only suitable for a speaker setup for which it was designed for. Although considerably less, the same constraint applies to stereo as well. In all channel-based audio formats, the audio is mixed into a fixed number of channels. Therefore, a stereo recording is mixed into two channels to be heard either with stereo speakers or headphones with the left channel being on the left and the right channel being on the right. In all speaker setups, the generic design pattern is to think that the listener is always somewhere in the middle. This way most speaker setups can provide the ideal listening experience.

An honorable mention called binaural audio delivers a fully 360-degree soundscape through a specially-encoded stereo file. The binaural recording method models the way sound reflects around the head and within the folds of ears. It is often recorded by using a dummy that resembles the shape of a human head. The recording happens by placing two microphones inside the ears of the dummy to mimic the real situation. To get a better understanding of how a binaural recording behaves, this video shortly demonstrates the traits of binaural audio. However, binaural audio has two major disadvantages. One, it is not responsive to user input, meaning that if a listener moves their head, the audio won’t change accordingly. And two, to get the most out of a binaural recording, it needs to be experienced through headphones.

While channel-based audio can be suitable for situations where the listener is expected to sit still, such as when watching a movie or listening to a record, its limitations manifest themselves when it comes to immersive media such as VR where it is often an essential part of the experience that the subject may freely move around in the virtual environment [4]. This means that in channel-based audio if a listener moves or rotates their head, the soundscape won’t change accordingly to the movement. This is the reason why VR requires an audio solution that can bring out the full-sphere of sound. A solution that can comply with any position or rotation of the listener. Such a solution exists, under the name ambisonic audio.

### Ambisonic Audio

Ambisonic audio, or ambisonics, is a technology that was developed in the 1970s. But, at that time, due to the lack of suitable applications, this technology's use was quite non-existent. Only recently, elevated by virtual reality, has this technology found its purpose. To put it shortly, ambisonics is a surround sound format that covers a full-sphere of sound, as was mentioned in the previous chapter. In addition to the horizontal plane, the technique can cover sound sources above and below the listener. With ambisonic audio, the full-sphere of sound can theoretically be decoded and routed to any number of speakers. But then again, unlike other multichannel audio formats, its transmission channels do not carry speaker signals. The strength of ambisonics is that when a listener with stereo headphones and trackers rotates their head, the soundscape does not turn with the listener. Therefore, in practice, the soundscape stays still while the listener is free to look around and the sounds will change accordingly to the movement. Ambisonics is an ideal technology for VR and other forms of media that might require similar methods.

Ambisonic audio is recorded by using a special microphone. These microphones have “microphone capsules” that point in different directions. The capsules and the directions where they are pointing at together enable the full-sphere of sound. The capsule arrangement varies depending on the manufacturers. All ambisonic microphones that have four capsules, which is the minimum amount of capsules, are called first-order ambisonic microphones. Having more capsules increases the resolution of the recordings. Such higher resolution ambisonic recordings are called second-order ambisonics (eight capsules), third-order ambisonics (sixteen capsules), and so on [1].

Picture 1. A picture of a first-order ambisonic microphone by Sennheiser.

## 2.2 Spatial Hearing

Spatial hearing is the capacity of the auditory system to interpret or exploit different spatial paths by which sounds may reach the head [2]. In other words, by using spatial hearing, our auditory system can determine the location of a sound source and identify certain sounds from each other. It is also capable of exposing specific sounds obscured by noise. Another one of its traits is that it can orient attention towards a sound source, and vice versa. Spatial hearing is almost entirely derived from binaural hearing, which, in essence, means to compare a sound signal at one ear with the other ear. These comparisons are reflected in terms of differences between time and level and are termed, interaural time difference and interaural level difference and are the basis of all binaural processing and are fundamental to nearly all spatial hearing [2].

The mechanisms of the auditory system involved in sound localization have been extensively studied. They use several cues to localize sound sources such as the previously mentioned time and level differences between both ears. The auditory system also analyses timings and correlations to support the localization of sounds. Apart from humans, other animals use these cues as well. However, differences in the effectiveness of these cues between humans and animals exist because humans, undeniably, are not able to move their ears to further enhance the ability to localize sounds. Therefore, animals have an evolutionary advantage over humans when talking about sound localization.

**Interaural Time Difference**

The interaural time difference (ITD) is the time difference of a sound between two ears. For example, if a sound is heard directly on the right, it takes longer for that sound to reach the left ear. This difference can be calculated with the angle of the sound source. The time difference itself works as a cue in sound localization. This means that it helps the listener to localize the direction or angle of a sound source. Another equally influential factor in sound localization is the difference in intensity.

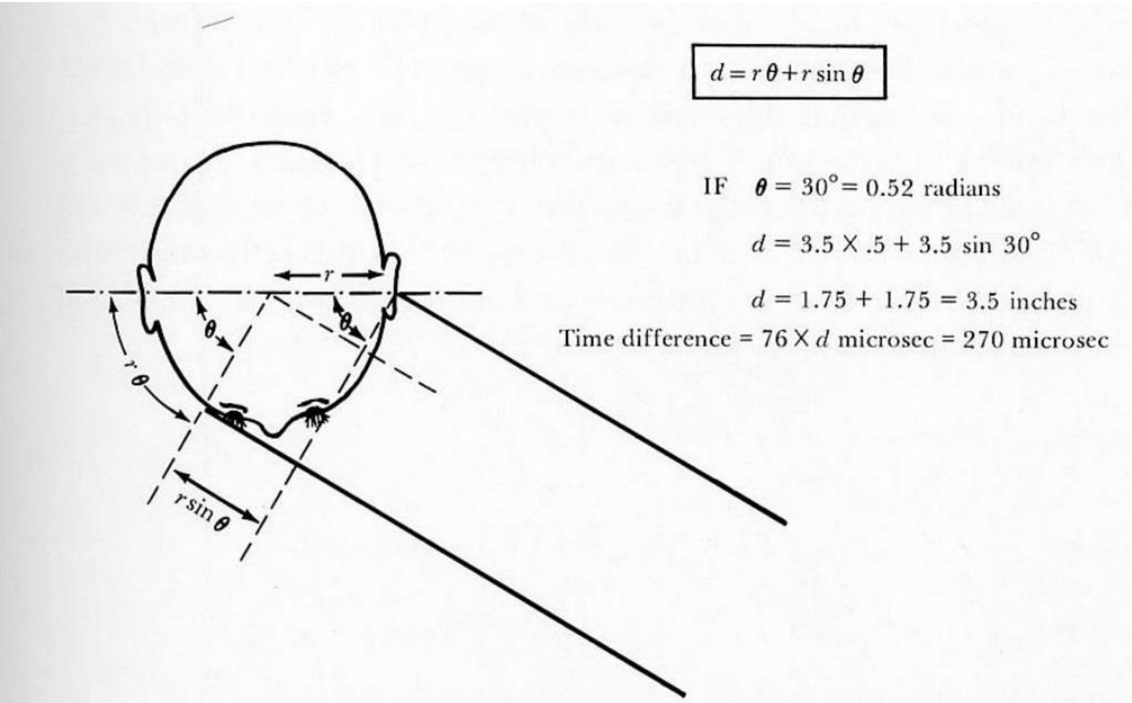



Figure 1. A figure showing how to calculate the time difference between both ears.

**Interaural Level Difference**

The interaural level (or intensity) difference (ILD) is the difference in loudness and frequency distribution between two ears. It is rather obvious that when a sound is heard close, it can occur as loud. But if that same sound is heard at some distance, it won't be as loud as before. The reason is simply that when sound travels, its strength dissipates. Amazingly, our ears can detect differences in loudness between the left and right ears. However, a far more important thing for sound localization is that the head casts an acoustic shadow, which changes the loudness and frequency distribution of sound going to each ear. An acoustic shadow or sound shadow is an area through which sound waves fail to propagate, due to topographical obstructions or disruption of the waves via phenomena such as wind currents, buildings, or sound barriers [12]. In this case, the acoustic shadow can be defined as the area behind the head opposite from a sound source. In the area, the loudness of a sound is less because the head blocks sound waves as visualized in Figure 2. It is good to know that low-frequency sound waves are more difficult to block than high-frequency sound waves.

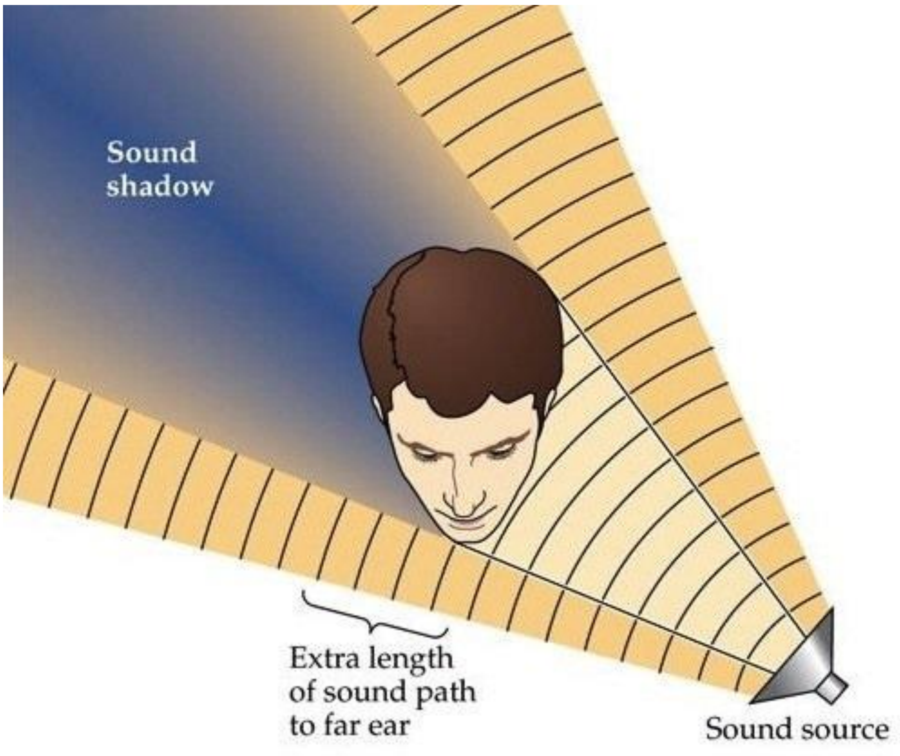



Figure 2. A representation of the acoustic shadow.

### 2.2.1 Head-Related Transfer Functions

Head-related transfer functions (HRTFs) are measurements that capture the directivity patterns of human ears. In other words, they are measurements that capture the way sound reaches the left and right ear. As sound strikes the listener, the size and shape of the head, ears, ear canal, density of the head, size and shape of nasal and oral cavities, all transform the sound and affect how it is perceived, boosting some frequencies and attenuating others [13]. HRTFs depend on several factors such as the direction, elevation, distance, and the frequency of a sound. The HRTF of a person can be measured by using a setup, similar to the one in Figure 3. In this setup, speakers are rotated around a listener who has microphones in both ears. These measurements will indicate the listener's spatial directivity patterns, or more precisely, their HRTFs. Through machine learning, we can synthesize personalized HRTFs by using anthropometrics: head width, height, depth; entrance coordinates of the ear canals; and more [3]. On the other hand, a crude head scan is enough to measure HRTFs. The more information and parameters gathered, the more accurate and personalized the HRTFs are. To make this technology usable from a practical standpoint is to find the right balance between anthropometrics and good-enough HRTFs [3]. As a clarification, anthropometry refers to the measurement of a human individual. It is a tool used for purposes to better understand human physical variation [5].

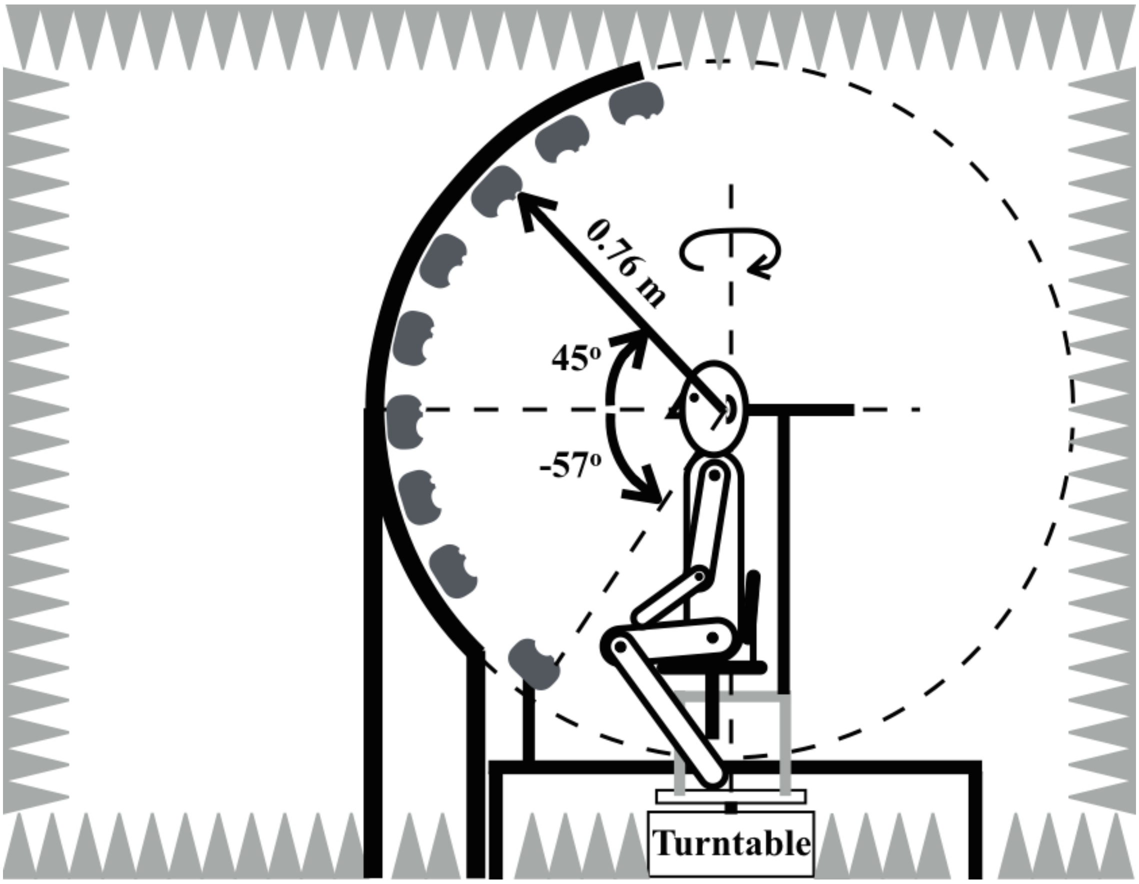



Figure 3. A representation of an example of an HRTF measurement setup.

## 2.3 Applications for Spatial Audio

Gaming and games are ideal applications for HRTFs because of the availability of the coordinates of the sound sources in a virtual space. This belief is supported by the ability to place each source to desired locations within the virtual space. In other words, game developers can place the sound sources to where the objects are visually positioned. Both augmented and virtual reality are also ideal applications where spatial audio is a must-have feature. Immersive spatial audio solutions are considered to be industry-standard for VR. Another ideal application for spatial audio is called 360-degree videos. They're also known as immersive videos or spherical videos because of the ability to view the video recordings from any direction. These recordings are shot using an omnidirectional camera or a collection of cameras that record all directions at the same time. During playback, the viewer can control the viewing direction similarly to panoramas or wide-view pictures.

## 2.4 Immersion in VR

Immersion in VR is the perception of being physically present in a virtual environment. This perception is created by surrounding the user of the VR system in images, sound, or other stimuli that provide an engrossing total environment [14]. To reach the perception in full, the VR application would need to affect all five senses: sight, sound, touch, smell, and taste. By fooling all five senses, the virtual environment would begin to feel physically real. However, with modern technology, immersing all five senses or directly affecting the nervous system is only a hypothetical future technology. For the most part, VR provides immersion only through visuals and audio. As it was mentioned in section 1, audio is an important factor in the entirety of VR, meaning that VR applications should aim to give the players auditory feedback. And if the immersion of the audio can be increased through different audio formats and techniques, theoretically, the hearing experience would only improve.

According to a university professor and a game design consultant Ernest W. Adams, immersion, in general, can be separated into three main categories: tactical, strategic, and narrative immersion. When you’re tactically immersed in a game, your higher brain functions are largely shut down and you become a pair of eyes directly communicating with your fingers [15]. Tactical immersion is produced when a player is flooded with simple enough challenges for them to be able to solve them in a fraction of a second. When you’re strategically immersed, you’re observing, calculating, deducing [15]. Strategic immersion is about building a path to victory by optimizing every situation that the player comes across. A player gets immersed in a narrative when he or she starts to care about the characters and wants to know how the story is going to end [15]. What creates narrative immersion is storytelling. The storytelling has to be good enough for the players to get immersed in it. Storytelling includes dialogue, characters, and plots. Staffan Björk and Jussi Holopainen, in *Patterns In Game Design*, divide immersion into similar categories, but call them sensory-motoric immersion, cognitive immersion and emotional immersion, respectively [14]. In addition to the three main categories, they add a new category that they call spatial immersion, which occurs when a simulated world is perceptually convincing to a user.

## 2.5 Unity's Audio System

A game would be incomplete without some kind of audio. Be it background music or sound effects, most games require audio in some form to fulfill the criteria for them to be complete. But whether a game project requires audio or not, each game engine in the industry offers its own features for using audio. Unity's audio system is simple but flexible. It can import most standard audio file formats and has sophisticated features for playing sounds in three-dimensional space. Optionally, the audio can be played with effects like echo and filtering applied. In the system, sounds are emitted by objects and heard by listeners, and the way a sound is perceived depends on several factors. For example, a listener can tell roughly which direction a sound is coming from and may also get some sense of distance from its loudness and quality. Also, a fast-moving sound source, like a passing car, will change in pitch as it moves. This is a result of the Doppler effect that the audio system is using. As a clarification, the Doppler effect, also known as the Doppler shift, is the change in frequency of a sound wave concerning a listener who is moving relative to the sound source (Figure 4). A complete explanation of the Doppler effect is beyond the scope of this thesis.

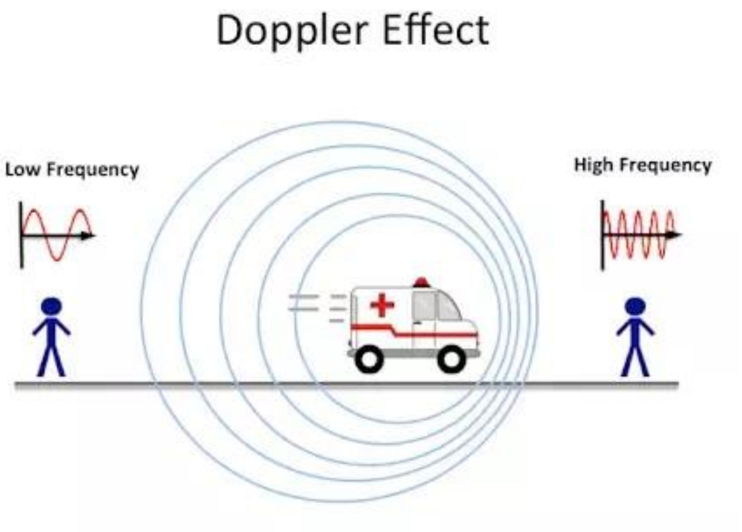



Figure 4. A representation of the Doppler effect.

The method that Unity's audio system is using to spatialize sounds is called stereo panning. The game engine calculates the distance between and the relative rotations of the sound source and the player and uses that information to set the appropriate volume level for the left and right channels [4]. However, this method cannot take the elevation of a sound source into account, thus, the spatialization mostly takes place on the horizontal plane only. Furthermore, the frequency spectrum of the sound isn't affected by the positions or the rotations of the sound source and the player in any way [4]. This means that the engine only tries to set the appropriate volumes for each channel without caring about the local rotations of the sound source and the listener. Another feature the audio system is unable to reproduce is the possibility of having reflections in sound caused by walls, floors, and other objects.

To simulate the effects of position, Unity requires sounds to originate from audio sources, which are components that can play sounds from the position of the object it is attached to [8]. The sounds emitted through these Audio Sources are then picked up by an Audio Listener, which is also a component that is attached to an object, most often the Main Camera. With these two components, the audio system can simulate the effects of distance and position of a sound source to the users.

Unity cannot calculate echoes purely from virtual geometry. However, it is possible to add audio filters to objects that have Audio Source components attached to them. For example, one could apply an echo to a sound that is supposed to play inside a cave. In situations where objects can move in and out of a place with a strong echo, a Reverb Zone component can be used to simulate an area in which sounds echo. While the object that has the Audio Listener component attached is inside the Reverb Zone, depending on the positions, all sounds heard by the Audio Listener will be distorted accordingly. For instance, a game could involve cars driving through a tunnel. If one places a Reverb Zone inside the tunnel then the sounds emitted by the cars will start to echo when they enter it. The echo will die down once the cars emerge from the other side.

In a recent update, it is now possible to import multichannel audio files into a Unity project. As it was mentioned before, all multichannel audio has to be used by a speaker setup that it was designed for so that the audio is played as intended. The same rule applies to the Unity projects using multichannel audio files. While multichannel audio offers more possibilities for creating different audio setups within a game, the sounds must still be used in a way that they match the speaker setup they were designed for.

## 2.6 Resonance Audio, the Spatializer Plugin

There are a few different spatializer plugins for game engines in existence. And the chosen plugin for this thesis, Resonance Audio, is one of them. Other plugins include Oculus Spatializer and Steam Audio. These two are also both great and valid options to enable more immersive spatial audio. The reason why this work is using Resonance Audio is solely because of a bachelor's thesis of Lasse Pouru from 2019 [4], which focuses on finding the optimal spatial audio solution and its parameters for Unity. This work uses the findings and results found in his thesis, and this includes using the Resonance Audio spatializer plugin.

Spatialization can be divided into four levels of complexity: basic 3D spatialization, modeled attenuation and occlusion, the addition of early reflections and room setups, and full reverberation and advanced geometry tagging. If Unity's default spatialization is stereo panning and adjusting the volumes of all sounds based on distance, then spatializer plugins like Resonance Audio can enable most of the levels mentioned above. The first level, basic 3D spatialization equals to have techniques like 360-degree localization, acoustic shadowing, and vertical sound applied. In this thesis, we attain all three techniques by simply setting up the spatializer plugin and going through the required components and parameters.

Resonance Audio supports a variety of different applications and development tools. It can be deployed across multiple platforms such as Unity and Unreal as well as middleware such as FMOD and Wwise. One of the plugin’s advantages is that its SDKs work to streamline workflows of developers and sound designers alike. Resonance Audio was built with technology optimized for mobile’s limited computational resources, providing advantages for developers who want high-quality, cost-effective output [6]. The purpose of this plugin is to simulate how sound waves interact with human ears. It uses interactions with sound waves to determine the horizontal and vertical locations of a sound. These interactions are replicated to create the illusion of sounds coming from specific locations within a virtual space. The list of features the plugin can imitate are interaural time and level differences, spectral effects (elevation), audio cues with HRTFs, simulating sound waves interacting with their environment, early reflections and reverb, occlusion and directivity, and last but not least, ambisonics. If game developers were to give a little more effort to fine-tune the spatial audio within their 3D game projects, there is a lot that can be achieved by using a spatializer plugin.

# 3 ENABLING AUDIO SPATIALIZATION

This section will go through all the components and features needed to achieve spatial audio with both Unity and Resonance Audio. However, this section won't go through all the logic behind each component. First, we will go through the required components needed to attain 3D spatialization using audio features built into Unity.

## 3.1 Audio Spatialization in Unity

The first step is to find the Audio Listener component, which is attached to the Main Camera object in fresh, out-of-the-box Unity projects. The Audio Listener is a component that listens and captures all heard sound waves within certain adjustable limits. In short, it is a tool to output audio to the player. In other words, the Audio Listener acts as the player's ears. The Audio Listener is attached to the Main Camera by default because most often games want to utilize a first-person player character. Of course, moving the Audio Listener component is perfectly allowed. Depending on the design of your game, you may not always want to keep it attached to a camera. There is nothing that needs to be done in preparation regarding the Audio Listener whose logic is all handled by the engine itself. Simply by having an Audio Listener in a scene is enough. However, it is good to keep in mind that no more than one Audio Listener can exist simultaneously as the engine will start to complain about the fact that two or more Audio Listeners exist at the same time.

The next step would be to create an object and attach an Audio Source component to it. The Audio Source, as explained in section 2, is a virtual sound source capable of playing sounds within the virtual space. Unlike the Audio Listener, it has a handful of parameters and settings that can be adjusted so that the source might match the needs of different projects and ideas. The following screenshots will show what settings are the bare minimum that needs to be adjusted to enable 3D spatialization.

In Figure 5, only the two settings marked in red are of importance in this context. They are the Audio Clip slot and the Spatial Blend slider. The Audio Clip slot is where the audio file is placed. A file can be placed there either by manually dragging a file to it in the editor or by setting a file to it via code. Once prompted, the Audio Source will play the Audio Clip that is placed onto that slot. The Spatial Blend slider determines how much the Audio Source is treated as a three-dimensional source. 3D sources are affected by spatial position and spread. The spatial position signifies the geographical position in the virtual space, whereas the spatial spread signifies the angle of a 3D stereo or multichannel sound in the speaker space. A 0-degree angle is when all sound channels are located at the same speaker location (mono). A 360-degrees angle is when all subchannels are located at the opposite speaker location compared to the speaker location where it should be according to its position. For the audio to be spatialized, we will set the Spatial Blend slider to 1 to let the engine know that we want this Audio Source to be treated as a 3D source.

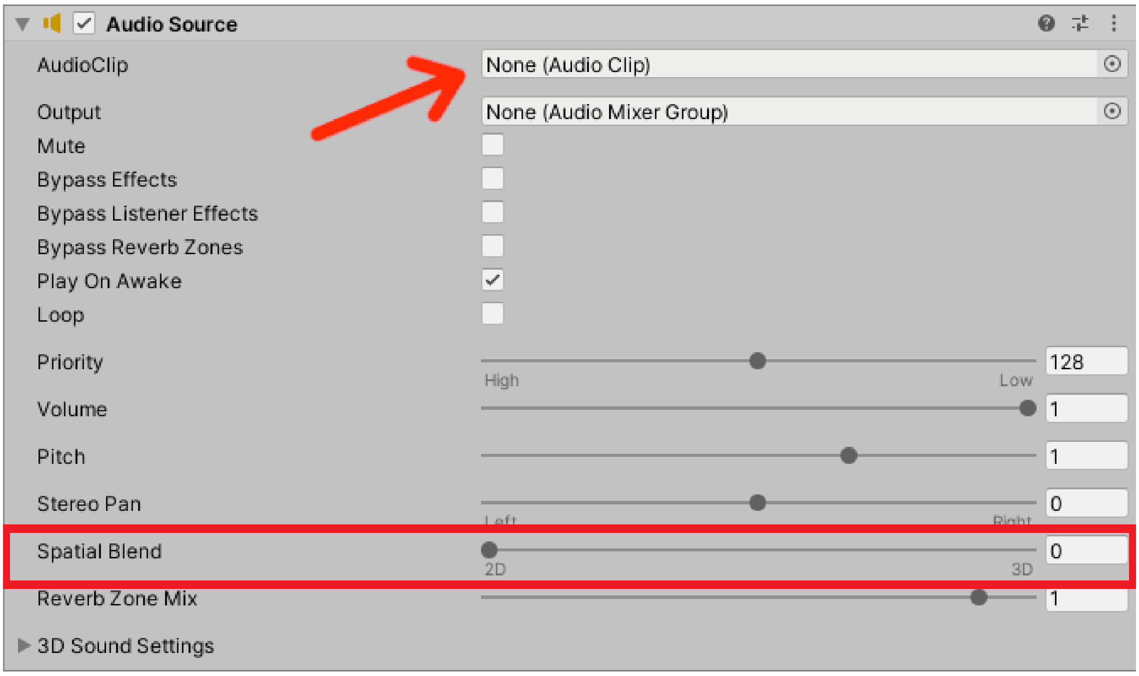



Figure 5. A screenshot of the settings in an Audio Source component in Unity.

Figure 6 reveals a set of additional settings in an Audio Source such as the possibility to adjust the Doppler effect and the ability to modify the volume's roll-off algorithm, or in this case, the roll-off curve. The Volume Roll-Off determines how the volume of a sound increases and decreases depending on the distance between the Audio Sources and the Audio Listener. By default, the roll-off is set as a Logarithmic Roll-Off since it corresponds to real life. Another option is to set it as a Linear Roll-Off, in which the volume of a sound changes linearly. In addition to these two roll-offs, the curve can also be adjusted manually by setting it as a Custom Roll-Off. But in this work, we will keep the roll-off as logarithmic because it adds to the realism.

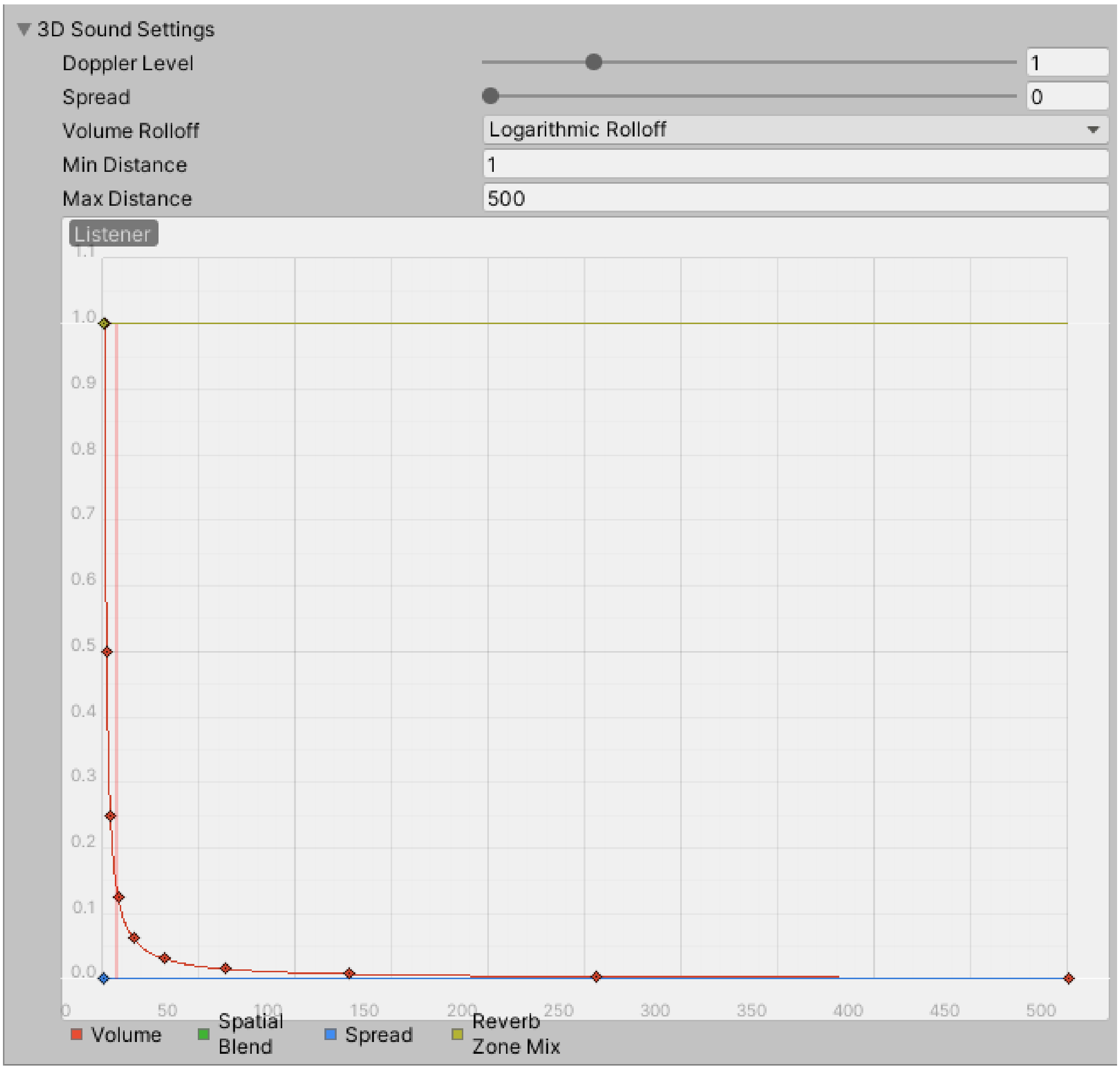



Figure 6. A screenshot of extra 3D sound settings in an Audio Source component in Unity.

This is the bare minimum that has to be done to enable spatial audio with features provided by Unity. We didn't go through all the settings because they are only optional in this context. Also, a handful of other audio-related components weren't used in this setup, such as the filters that were mentioned in section 2, because they are not required either. At this point, we can't help but notice how easy it was to enable spatial audio in a Unity project. Although, this should not be taken to heart as this doesn't necessarily mean that it is always as simple as this to enable spatial audio for every Unity project one might come across. In many cases, project structures and other needs that might complicate the usage of Audio Sources and affect where they are located must be taken into account.

## 3.2 Adding Resonance Audio

So far, we have seen how to achieve spatial audio in a Unity project by doing the bare minimum with Unity audio features. For comparability purposes, we must now import the Resonance Audio spatializer plugin to the same project, and, as we did with the Unity's features, do the bare minimum to enable spatial audio by using the plugin.

First, a package file containing the spatializer plugin needs to be downloaded from the Resonance Audio website. Once the package file has been downloaded, it needs to be imported to the project. Importing the package file will bring the necessary assets and tools to the project, but will not touch any project settings that would cause systematic changes. Next, to enable a custom spatializer plugin in a Unity project, one has to open the project settings and assign the Resonance Audio as the spatializer plugin. To do this, one needs to go to [Edit > Project Settings > Audio] and change the spatializer plugin and ambisonic decoder plugin to Resonance Audio from the drop-down menus. With this, we have gained access to the audio features provided by the Resonance Audio spatializer plugin by informing the engine that we want it to treat the plugin as the project's spatializer.

Resonance Audio provides an Audio Listener component and an Audio Source component that are separate from the ones built into Unity. However, Resonance Audio's components are add-ins that won't replace the components provided by Unity. So, in the Main Camera's case, both Unity's and Resonance Audio's Audio Listener components need to be attached to it for the plugin to work as intended. The same goes for the Audio Sources. If the Audio Source needs to play sounds that the engine should deem as sounds using the spatializer plugin, it is important to have all the necessary components attached and be certain that all settings are correct too.

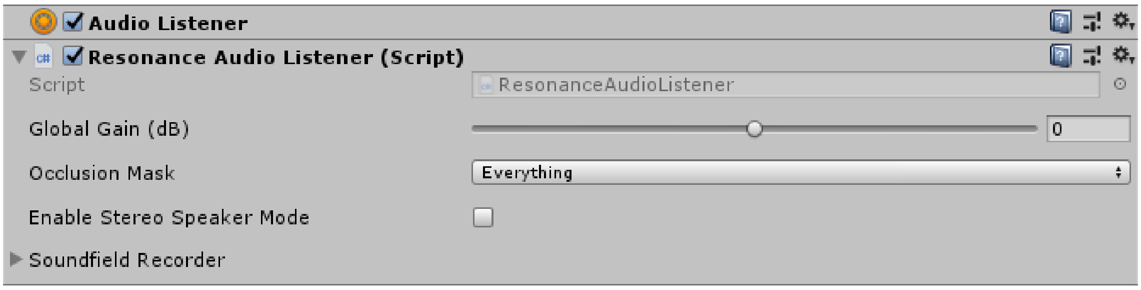



Figure 7. A screenshot of both Unity's and Resonance Audio's Audio Listener components attached to an object.

Like previously, every setting marked in red in Figure 8 is of importance for the current setup. Regarding the Resonance Audio spatializer plugin, toggling the Spatialize setting on in the Audio Source component is quite critical. Notice that it is missing from the Audio Source in Figure 5. This is because the Spatialize and Spatialize Post Effects are settings that the plugin brought with it. The reason why the Spatialize setting is important is that when it's toggled on, the Audio Source is treated as a source that uses the plugin and the same object probably has the Resonance Audio Source component attached as well. Therefore, when it's toggled on, the engine knows that the Audio Source wants to be spatialized with the plugin. The Spatialize Post Effects setting is more or less optional. It only determines whether the spatializer plugin is applied after or before the possibly attached audio filters are applied to the Audio Source. It won't have any effect on the Audio Source if the Spatialize setting is not toggled on or if there are no filters applied to the Audio Source. Just as it was set to 1 in the setup only using audio features provided by Unity, the Spatial Blend is also set to 1 here.

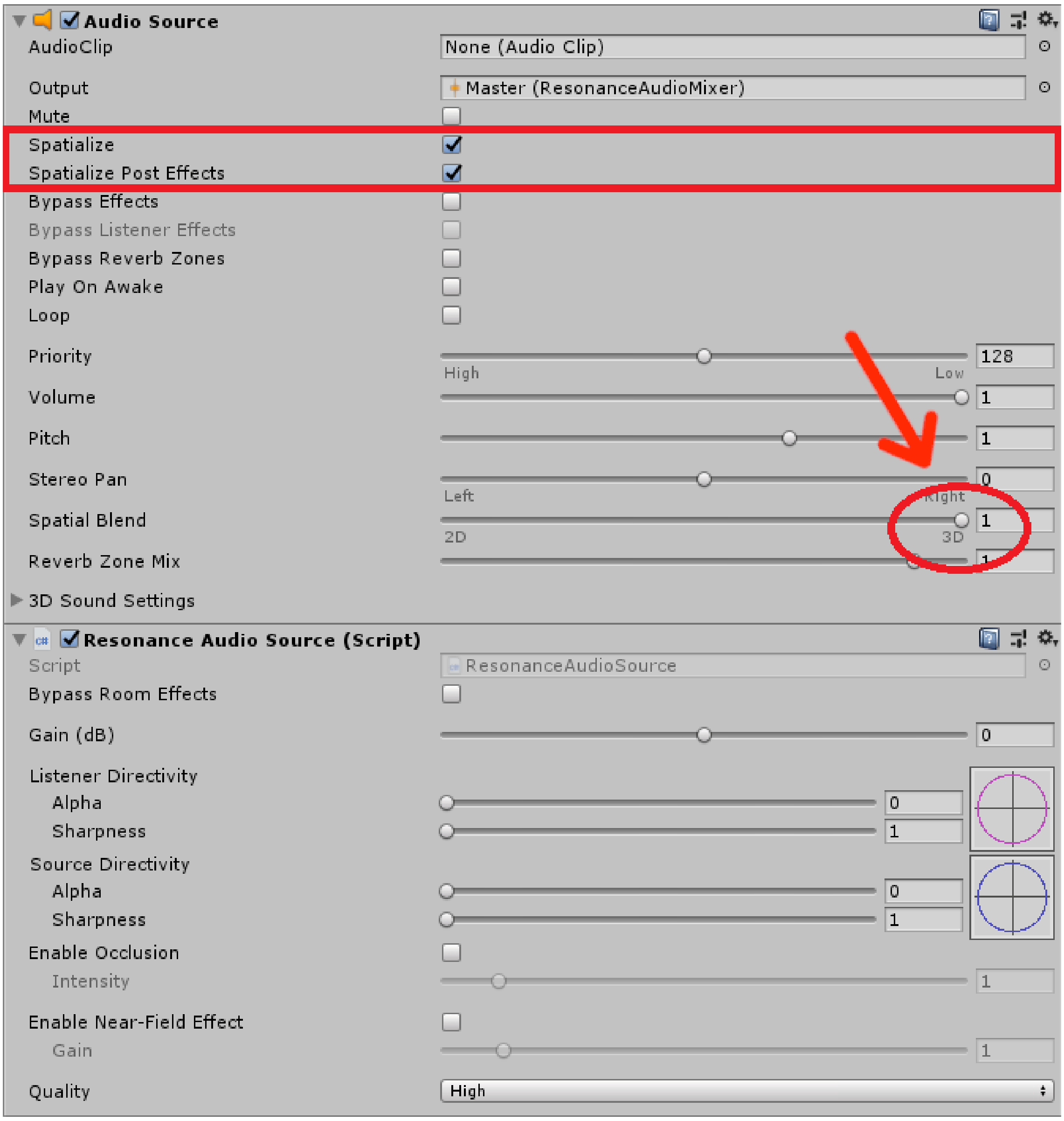



Figure 8. A screenshot of both Unity's and Resonance Audio's Audio Source components attached to an object.

It is worth to mention that Unity has an audio mixing system built into it. A handful of different feats are achievable by using an Audio Mixer. As can be seen in Figures 5 and 8, there is an Audio Mixer Group slot right under the Audio Clip slot. To use Audio Mixers is completely voluntary. The most notable feature of an Audio Mixer is that it bundles Audio Sources together. With an audio mixer group, one can easily adjust the volumes of all sounds that belong to the group with a single slider. For example, it is common to separate sound effects and music from each other by making groups for them in an Audio Mixer. Additionally, certain effects such as reverb can also be applied straight to the

Audio Mixer Group without having to apply them to all source objects separately. In other words, every Audio Source using an Audio Mixer Group that has effects applied to it will have the same effects applied to them as well. However, another noteworthy point is that the Resonance Audio spatializer plugin comes with an Audio Mixer of its own. The only difference between a default Audio Mixer and the one provided by Resonance Audio is that the plugin's Audio Mixer has a different audio renderer in it. As of yet, Resonance Audio does not support grouping and mixing of audio. So, unfortunately, many features provided by the Audio Mixers such as mixer groups and snapshots are not supported by the plugin.

With this, we have successfully enabled spatial audio using the Resonance Audio spatializer plugin. It wasn't as overwhelming as one could expect. Although, we have only scratched the surface. Spatializer plugins can do a whole lot more to further enhance the hearing experience and to increase the immersion of the spatial audio. But everything we went through is enough for this thesis.

## 3.3 Audio Management

During an advanced game technology course, at TUAS with Turku Game Lab (TGL), a certain VR project called Simulandia had to be further developed. The customer of the project is a research and training organization called TTS who hired a graphic arts professional and software development expert called ADE Oy to do the project for them. ADE is a regular RDI and training partner of TGL and often enlists the help (in the form of thesis workers, trainees, and/or group projects) of students on their projects. That is where this thesis comes in. So, because this thesis had the opportunity to use scenarios from a high-fidelity VR project such as Simulandia, the usage of sounds would no doubt be dependent on the structure of the project. For this reason, certain custom-made scripts had to be programmed to play sounds in the project. This part will go through shortly how the scripts work without going too much into any technical details.

To make it easier to perceive and go through the features of the Scriptable Object, the settings have been divided into three parts: red, blue, and green, as is revealed in Figure 9. One might notice that there are similar settings visible in Figure 9 that are in the Audio

Source components provided by both Unity and Resonance Audio. The truth is, the Scriptable Object tries to control the Audio Sources that are in use to make it easier to use spatial audio in demanding 3D projects such as Simulandia. The functions of the Scriptable Objects have been tailored to collect all necessary settings to a single place while allowing them to be used in a simple but clever manner. If a developer using the Scriptable Objects and the Audio Manager wants a certain Audio Source to be spatialized with the Resonance Audio spatializer plugin, it is achievable. Also, if a developer using the custom-made scripts wants another Audio Source to not be spatialized with the plugin and only use the features provided by Unity, it is achievable as well. In addition to those, if a developer using the scripts wants an Audio Source to play sounds two-dimensionally, it would be weird for the scripts to not be able to do that. Therefore, the custom-made scripts are capable of controlling and playing both types of audio including spatial audio that uses a spatializer plugin.

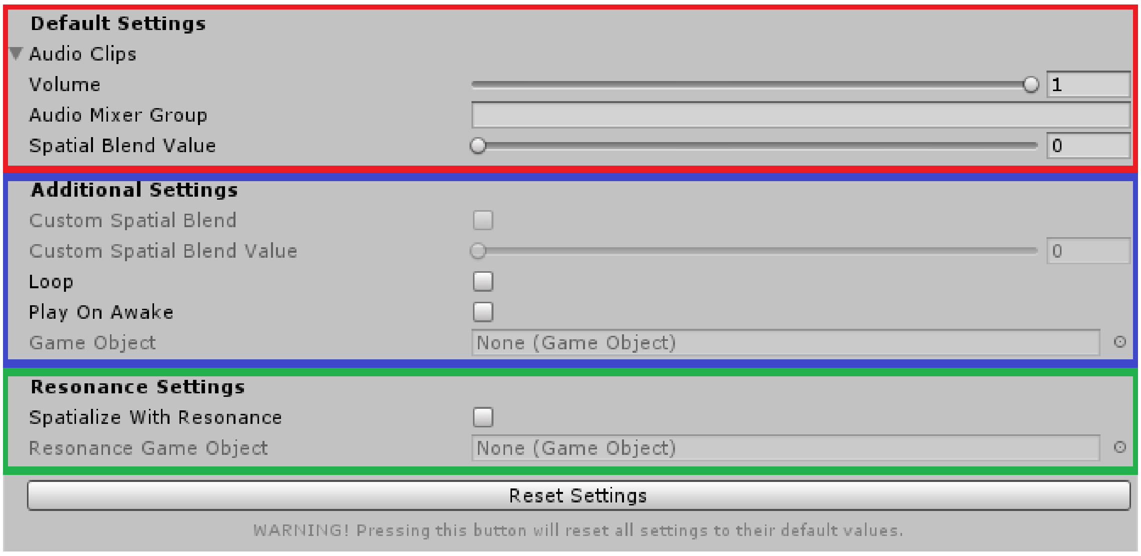



Figure 9. A screenshot of the available settings in a custom-made Scriptable Object.

As good as the scripts sound, there are still certain structural limitations to them. For instance, most of the optional effects and components mentioned in section 2 are not taken into account by the Scriptable Objects nor the Audio Manager. However, the code can be extended to support other audio-related components and features in the future. Unlike most scripts, the Scriptable Objects are scripts that are not attached to objects or prefabs. They are assets that only exist in the project's data. They can, however, control assets and information provided by other objects, scripts, and components that exist in a scene. The Scriptable Objects are not limited to one and each Scriptable Object created is an object of its own. Whilst, the Audio Manager who is watching and counting all the Scriptable Objects has to be attached to an object within a scene. As its name implies, it manages the Scriptable Objects, their settings, and other events related to them.

The settings displayed in Figure 9 are the options one can alter in a single Scriptable Object. As the heading states, the Default Settings marked in red are something that must at least be adjusted for the Scriptable Object to work. Clicking the Audio Clips setting opens up a drop-down menu where it is possible to insert as many audio files as one wants. For example, one could create a Scriptable Object to the project's data, insert a bundle of similar sounds to it, and then name it to match the sounds it holds. If there is more than one Audio Clip inserted to a Scriptable Object, the Audio Manager will always randomly pick one of them to be played whenever an event involving the Scriptable Object is prompted. The Volume slider is exposed in case a Scriptable Object is supposed to use the Resonance Audio spatializer plugin. As it was mentioned before, Resonance Audio does not support grouping and mixing of audio. That is the sole reason why the Volume slider is exposed. With the slider, all Audio Clips in a Scriptable Object are played with the volume specified in the setting. If the inserted Audio Clips should be spatialized, one setting from either Additional or Resonance Settings is a requirement depending on which audio solution the Scriptable Object should use. The settings in question are the slots that ask for a game object. The object that needs to be inserted to either one of these slots should have an Audio Source component attached to it. The Audio Manager tries to look for game objects that are in the Scriptable Objects. In case it can't find a game object that has the required components inserted in a Scriptable Object, it will generate an object as a child object under the object that has the Audio Manager script attached. The child objects generated under the Audio Manager object will have the Spatial Blend setting set to 0 by default, therefore, the sounds played through them are two-dimensional.

The Additional Settings marked in blue include a slider to set a custom value between 1 and 0 to Spatial Blend. It might seem silly to have multiple settings that can change the same value. However, the Spatial Blend slider in the Default Settings is meant for quick setups and thus can only be set to either 1 or 0. The Loop setting allows the Audio Clip selected by the Audio Manager to be looped. The Audio Clip will repeat itself until the Audio Manager is asked to stop playing it, or the active scene is reloaded or the scenes change. The Play On Awake setting makes it so that the Audio Clip selected by the Audio Manager starts playing as soon as the scene containing the manager is awakened. This setting is mainly for initial background music tracks.

The Resonance Settings marked in green are the settings that determine whether the Audio Manager should direct the Scriptable Object's sounds to the Resonance Audio spatializer plugin. Toggling on the Spatialize With Resonance setting nullifies the game object slot in the Additional Settings. If it's toggled on, the Audio Manager will try to look for an object inserted in the game object slot under the Resonance Settings instead. In summary, all objects inserted into the game object slot in Additional Settings should only have the Unity's Audio Source component attached to them because the Audio Sources attached to those objects use the Unity's spatialization solution. And all objects inserted into the game object slot in the Resonance Settings should have both Unity's and Resonance Audio's Audio Source components attached to them.


Figure 10. A screenshot of the available settings in a custom-made Audio Manager.

Once the Audio Manager script is attached to an object in a scene, it needs information from every Scriptable Object that could be called and used during gameplay. This means that to play Audio Clips that are inserted into Scriptable Objects, the Audio Manager will need to have references to the Scriptable Objects containing them first. The Scriptable Objects are inserted into the Audio Manager script in the same way audio files are inserted into the Scriptable Object. Clicking either the Sound Effects, Ambiance, or Music setting will open a drop-down menu to which the Scriptable Objects need to be dragged into. Once the Scriptable Objects have the correct audio files and Audio Source objects inserted into them and they are referenced in the Audio Manager, the system will work efficiently.

These scripts together enabled a couple of very vital things. Firstly, they made it possible to easily add sounds to the Simulandia's scenarios. I was able to directly create empty Audio Source objects and take them to the positions in the virtual space where I wanted certain sounds to play. And secondly, because the scripts support both Unity's and Resonance Audio's spatialization solutions, it was easy for me to create the A/B test to compare the two solutions. Also, together the scripts form a system that can be used in any other scenarios ADE and TTS might work on in the future.

# 4 TEST METHOD

So far, we’ve gone through the theory behind spatial audio and its formats while introducing both spatialization solutions provided by Unity and Resonance Audio. This section introduces the A/B test, shows the structure of the test that is used to expose VR users to both spatialization solutions and presents screenshots and descriptions of the two test environments. It also provides a few mentions about the participants and how the results were collected.

## 4.1 Environments

The test environments consisted of two scenarios from the virtual learning environment, Simulandia. The development of Simulandia’s scenarios is performed in sections, which means that every scenario is in a repository of its own. Thus, the whole project is divided into several parts. For this reason, it is easier to work on those parts when each scenario has a separate team working on it. The project parts utilized in this work are called Forest and Lifting.

### 4.1.1 Forest Scenario

The Forest scenario is a virtual environment where players are allowed to visit different types of forests while also providing them the ability to inspect, chop down, and grow trees. It aims to provide the players with information about different tree species and their growing environments while providing ways to take care of the available forest scenes. The Forest scenario is perfect to sample sounds designed for outdoors from bird’s singing to miscellaneous foley sounds.


Figure 11. A screenshot of the Forest scenario from Simulandia showing the information on a tree.

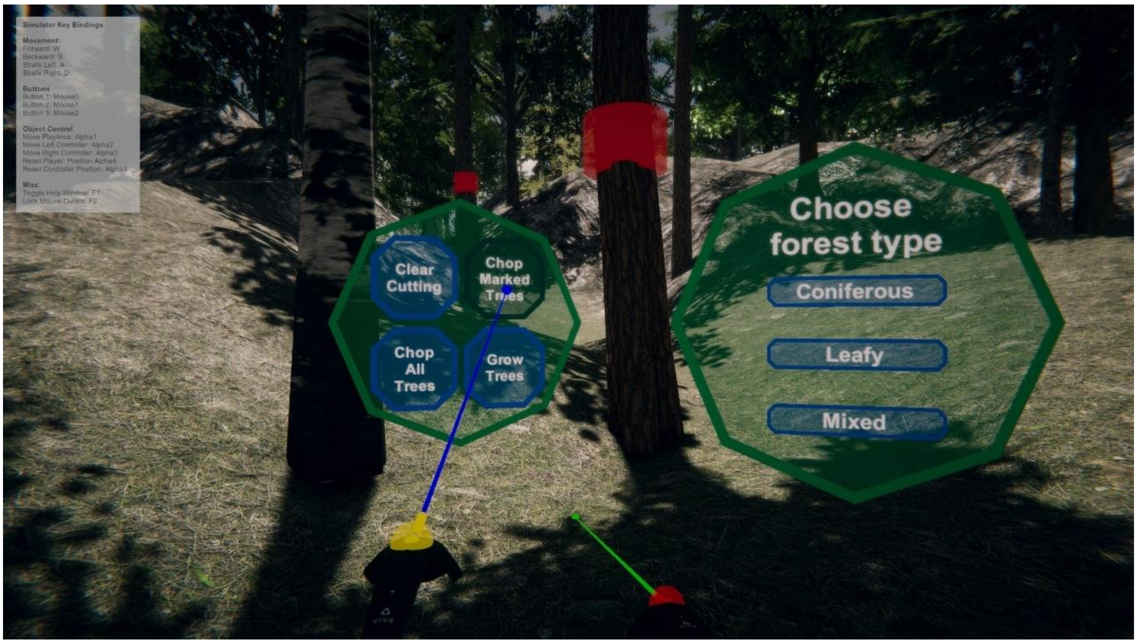



Figure 12. A screenshot of the Forest scenario from Simulandia showing the options granted for the player.

### 4.1.2 Lifting Scenario

The Lifting scenario is a virtual environment where players are allowed to operate a crane from its cockpit. The players can turn the crane, lift and lower objects with its hook, and move the hook system closer or further from the cockpit. The crane itself is operated from within the cockpit by using joysticks. The cockpit includes a monitor that provides the player with valuable information such as the hook's distance to the ground. The Lifting scenario is perfect to simulate the effect of being inside a room and the sounds heard from outside are being muffled and blocked by the walls, floor, and roof of the cockpit.


Figure 13. A screenshot of the Lifting scenario from Simulandia showing the crane's cockpit from the inside.

## 4.2 Test Subjects

For this work, the test subjects can be anyone with access to a VR headset and controllers. However, since the work aims to prove that using immersive spatial audio is better in VR applications than to go with solutions that just pass the mark of 3D spatialization, it is not enough that the participants only consist of ordinary consumers and users with little experience in VR. VR is known to be a very overwhelming and perhaps even motion sickness inducing experience for beginners and users that don't use the technology frequently. If the test would only gather results from users like that, for many, the results might get slightly skewed because of the general immersion produced by the virtual reality experience. The interest and focus towards audio and the soundscape might get overshadowed by everything else. Therefore, it is quite important to get results from more experienced VR users. This thesis certainly recognizes the importance of testing with TTS trainers and customers because these are ultimately personas of the Simulandia end-user group. However, due to the outbreak of COVID-19, only a limited number (7) of this user category were able to participate in the research. The test had an overall eighteen (18) participants from which seven were experienced VR users and industry experts and eleven regular consumers with less experience with VR.

### 4.3 Research Process

The sounds portrayed in the A/B test are played in a predetermined order. That way the test itself will always be the same for every participant. In A/B tests, it is quite crucial to maintain control over all the variables present because then the results collected are also more accurate. A/B testing is a way to compare two versions of a single variable, typically by testing a subject's response to a variant A against variant B, and determining which of the two variants is more effective [16]. In this case, the A/B test had to be implemented directly into both scenarios. Apart from the user interface related to the tests, all other interactions with the environments and the player movement had to be disabled to reduce the variables that could affect the results. Although the participants were allowed to move in place because for example, the head movement is an exclusive part of VR and also critically involved in the testing of the directivity of the audio. The next chapter is going to go through the implementation of the A/B test and its structure in better detail.

#### 4.3.1 Structure

The A/B test was implemented directly into both scenarios. For that purpose, the main scenes holding 3D models and assets in both scenarios were copied so that implementing the test to them would not cause any issues in the future. In other words, the original, main scenes were left untouched. In the scenes copied from the main scenes, all interactions with the environments were disabled. All movement and teleportation using the controllers were also disabled because the participants are required to stand or sit in one place. The test itself is a series of interfaces that exist in the virtual space. The participants had to use the controllers to press buttons for the test to progress. The next screenshots show how the test looks like in the Forest scenario.


Figure 14. A screenshot showing the A/B test's initial interface in the Forest scenario of Simulandia.


Figure 15. A screenshot showing the A/B test's instructions within the Forest scenario of Simulandia.

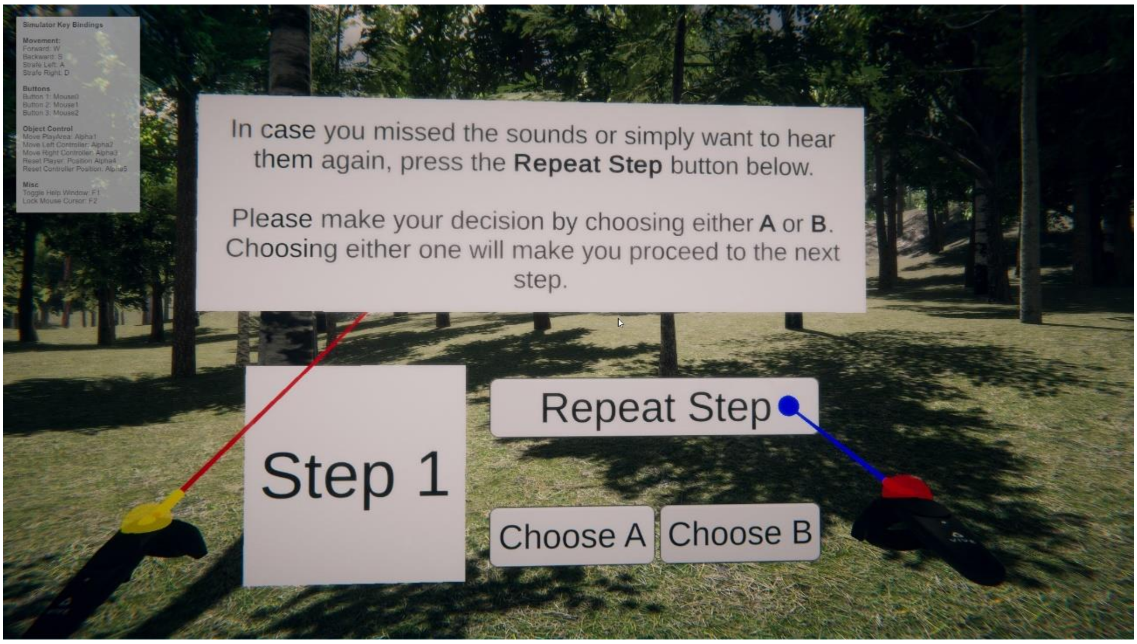



Figure 16. A screenshot showing the A/B test's available options in the Forest scenario of Simulandia.

The test progressed in steps. Each numbered step played a sound two times, once using the Unity's built-in spatialization solution and once using the spatializer plugin. The test did not, however, indicate which option used which solution. As it has been mentioned before, the sounds were played in a predetermined order defined beforehand in the script controlling the test. The participants were given the possibility to repeat the sounds in each step in case they wanted to hear them again before making their decisions. Both scenarios had four different sounds designed for them, therefore both tests played each of those four sounds twice. After the sounds have been played to the participant and they have made their decisions, the choices they made are shown for them at the end of the test. The questionnaire that the UI panel in Figure 17 is talking about refers to the way the results were collected.

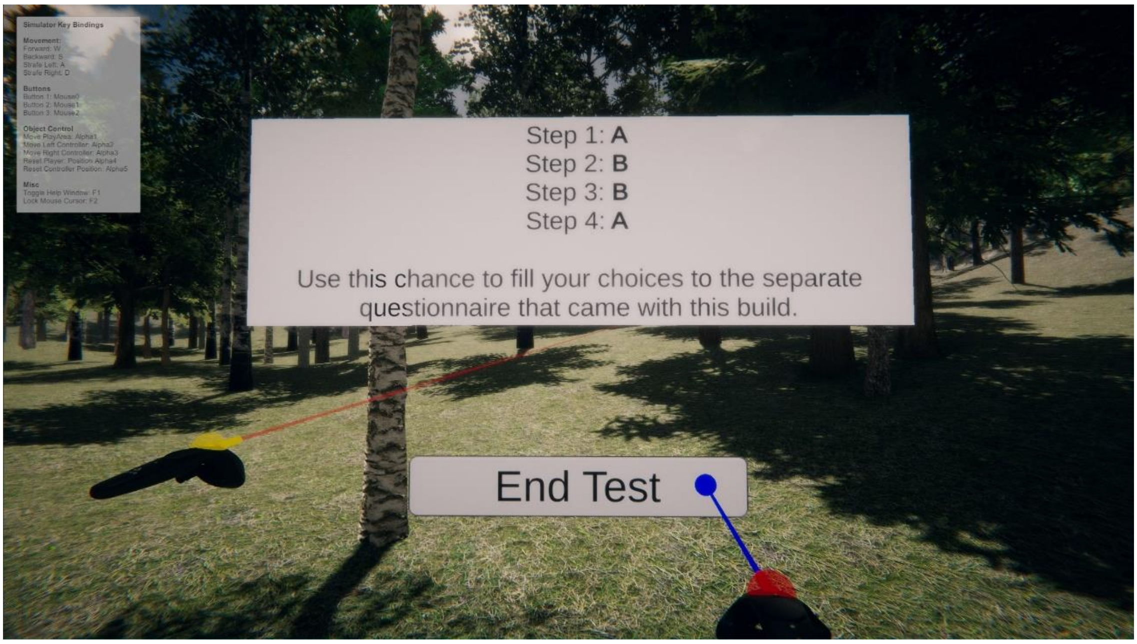



Figure 17. A screenshot showing the options chosen by a participant in the A/B test in the Forest scenario.

### 4.3.2 Communication

Communication with the test subjects happened mostly via text files and email. Due to the global pandemic named coronavirus, or more specifically, the COVID-19 virus [9], it was not recommended to perform testing in close contact. Therefore, considering the outbreak, the A/B tests were designed to be performed remotely. For both scenarios, the scenes containing the tests were built into executables that were packed into a compressed folder. The compressed folder or the zip file also contained a text file containing information about hardware and the builds themselves. After performing the tests, the participants were asked to submit the results to a separate questionnaire. A link to the form was also included in the text file that was sent alongside with the builds.

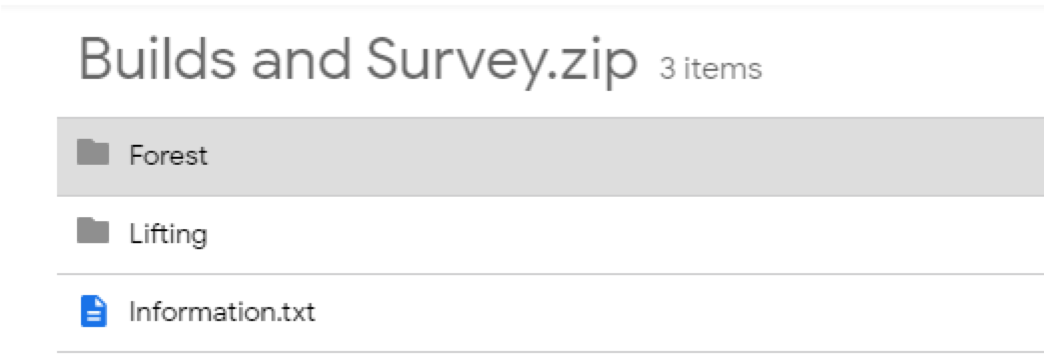
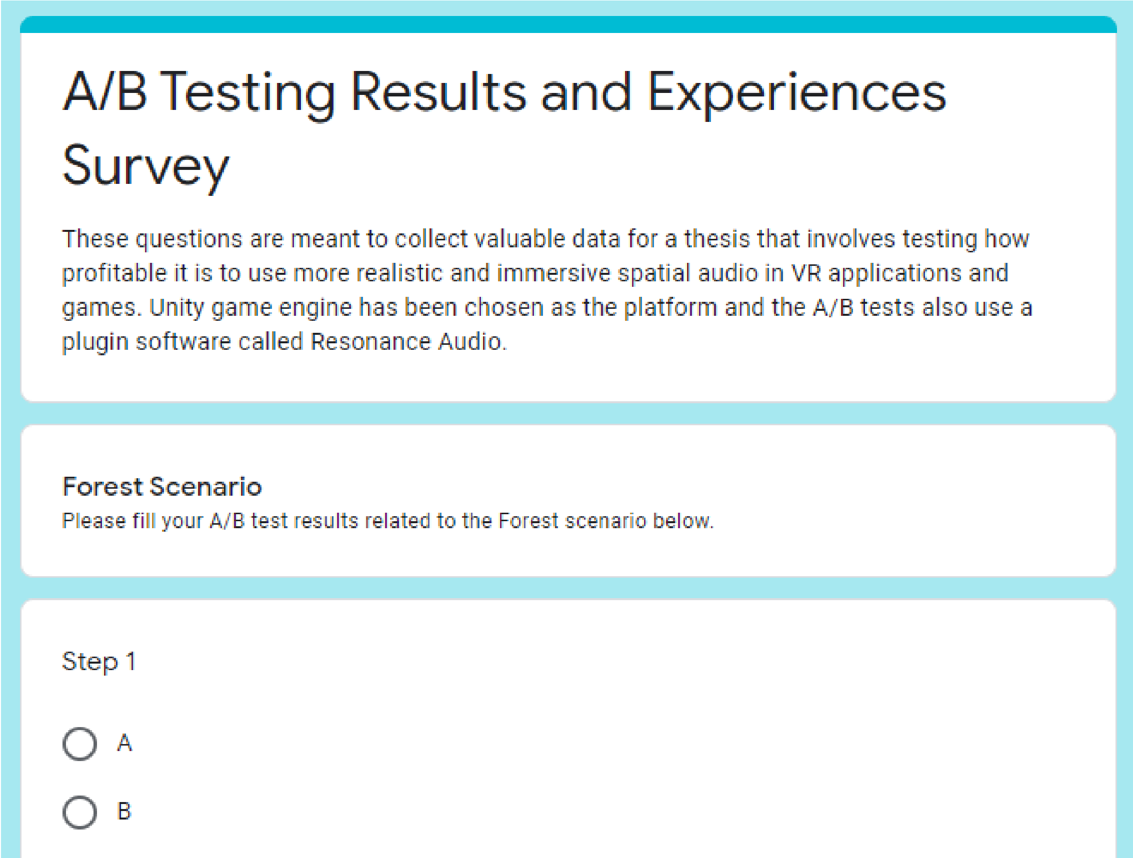



Figure 18. A screenshot showing the contents of the compressed folder.



A/B Testing Results and Experiences Survey

These questions are meant to collect valuable data for a thesis that involves testing how profitable it is to use more realistic and immersive spatial audio in VR applications and games. Unity game engine has been chosen as the platform and the A/B tests also use a plugin software called Resonance Audio.

Forest Scenario

Please fill your A/B test results related to the Forest scenario below.

Step 1

A

B

Figure 19. A screenshot showing the beginning of the questionnaire.

The full questionnaire can be found at https://forms.gle/UzduBYbWD94CkGqZ8.

The builds and the questionnaire were sent in an email to anyone interested in participating in the tests and to people with a VR headset. The results received came mostly from experienced VR users who owned a VR headset of their own. Many of the participants were students studying game development and other people working in the industry. Despite the pandemic, for a handful of participants, the tests were performed in close contact because they did not own the necessary devices. All close contact testing was performed carefully while disinfecting the VR headset and controllers.

# 5 RESULTS AND DISCUSSION

In this section, we will go through all the relevant results gathered by the questionnaire. Figure 20 reveals the results gathered from the Forest scenario's A/B test. For both scenarios, the collection of the results was divided into four steps. As already mentioned in the previous section, one sound was played twice in each step. The sound was played once using Unity's built-in spatialization solution and once using the Resonance Audio spatializer plugin. In the Forest scenario, the choices that used the plugin came in the order of B, A, A, B. As can be seen in Figure 20, the choices that used the spatializer plugin were somewhat favored. Although, upon on closer examination of the results, one might notice that the choices that used the plugin were not overwhelmingly more popular compared to their alternatives. A couple of factors might affect why the results are how they are. One factor may be that the position of the sound source was too close to the horizontal plane for the sound to stand out well enough from the solution provided by Unity. In other words, the improved directivity of the sound and the acoustic shadowing did not create a large enough difference for the differences in immersion to be noticed by some of the participants. Another factor may hide in the design of the sound effects. For both scenarios, the sounds were designed with the same values in mind and they were all mono.

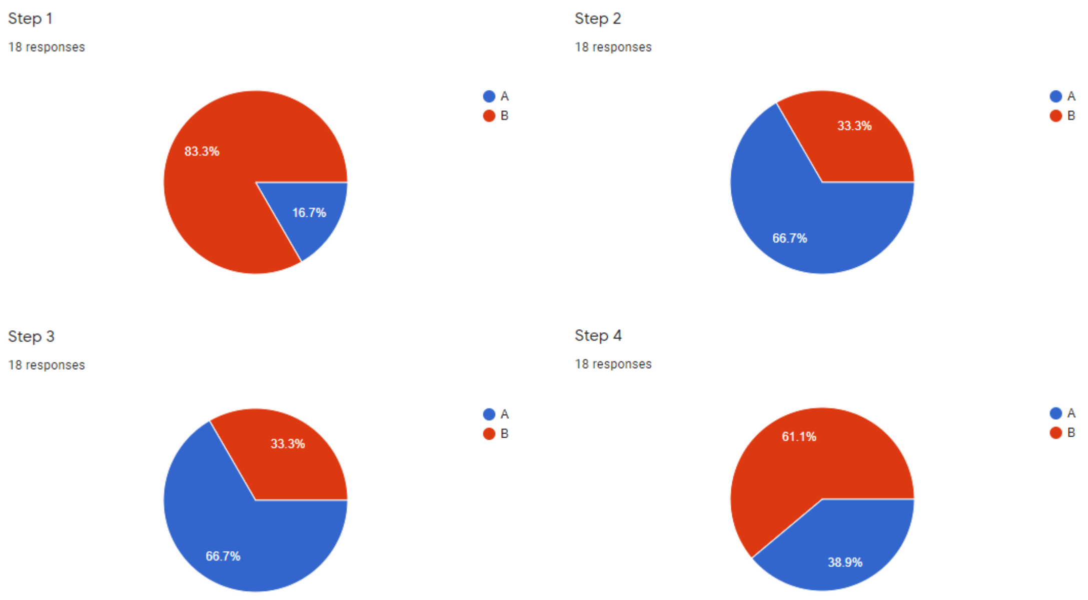



Figure 20. A bundled screenshot of the results of the Forest scenario's A/B test.

With the results collected from the Forest scenario's A/B test (Figure 20), we can calculate the significance rate for the Forest scenario. This is done by first calculating the conversion rates for both spatialization solutions. The conversion rate is calculated by dividing the traffic by the conversions and then multiplying it by 100. Fortunately, some online calculators can calculate the significance rates for us, as long as we know what values to use. This is demonstrated in Figure 21 by inputting the results collected from the Forest scenario's A/B test to an online calculator. Variation A represents the choices that used Unity's built-in spatialization solution. Variation B in turn represents the choices that used the Resonance Audio spatializer plugin. At this point, a question may arise as to why there is a total of seventy-two (72) traffic, even though there were only eighteen (18) participants that answered the questionnaire. The number 72 is obtained by multiplying 18 (participants) by 4 (sound effects). In other words, every participant made 4 choices per scenario.

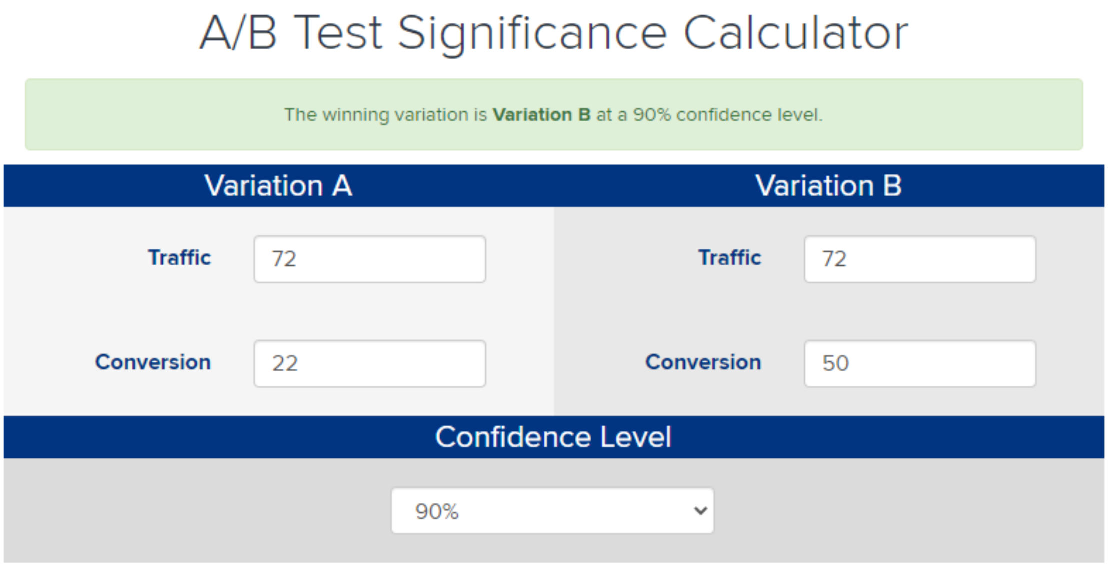



Figure 21. A screenshot of the Forest scenario's results inputted to an online A/B test significance calculator.

Figure 22 reveals the significance rate calculated by an online A/B test significance calculator using the results collected from the Forest scenario's A/B test. The uplift is calculated by dividing the conversion rate A by the conversion rate B and then multiplying it by 100. It is the relative increase in conversion rate between variation A and variation B. It would have been possible to have a negative uplift if the choices that used Unity's spatialization solution were more popular than the other choices. The p-value is the probability that the collected results have occurred as a result of random chance. If it is lower than the alpha value (100 minus the confidence level), then the results are significant. A high p-value simply means that the results are not significant.

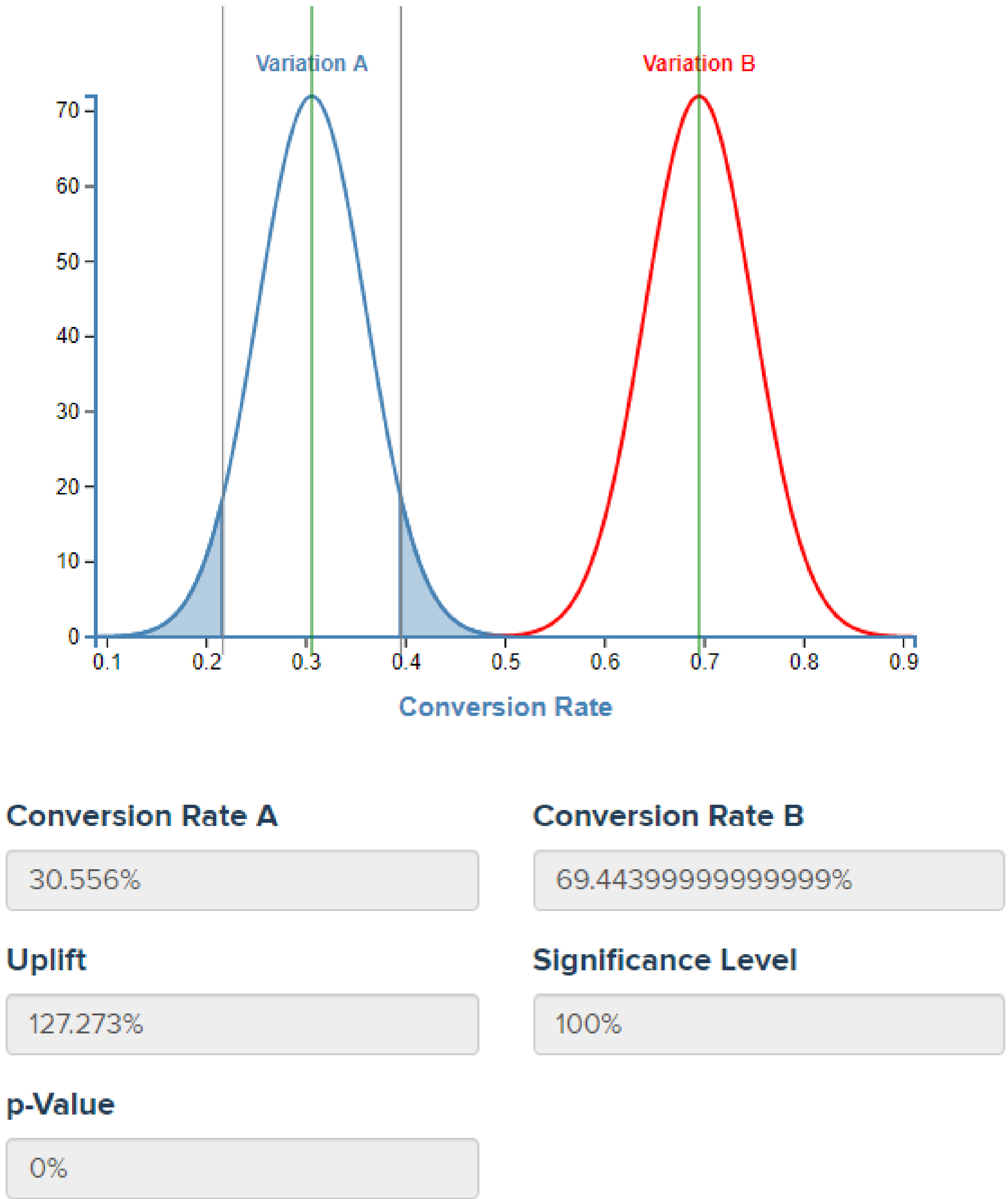



Figure 22. A screenshot of the Forest scenario's significance rate calculated by the online A/B test significance calculator.

Figure 21 reveals the results gathered from the Lifting scenario's A/B test. As with the results of the Forest scenario, the choices that used the spatializer plugin came in the order of B, A, A, B. Similarly, the choices that used the plugin were also more popular compared to their alternatives. The same factors apply here as well even though the sounds were designed for a completely different environment.

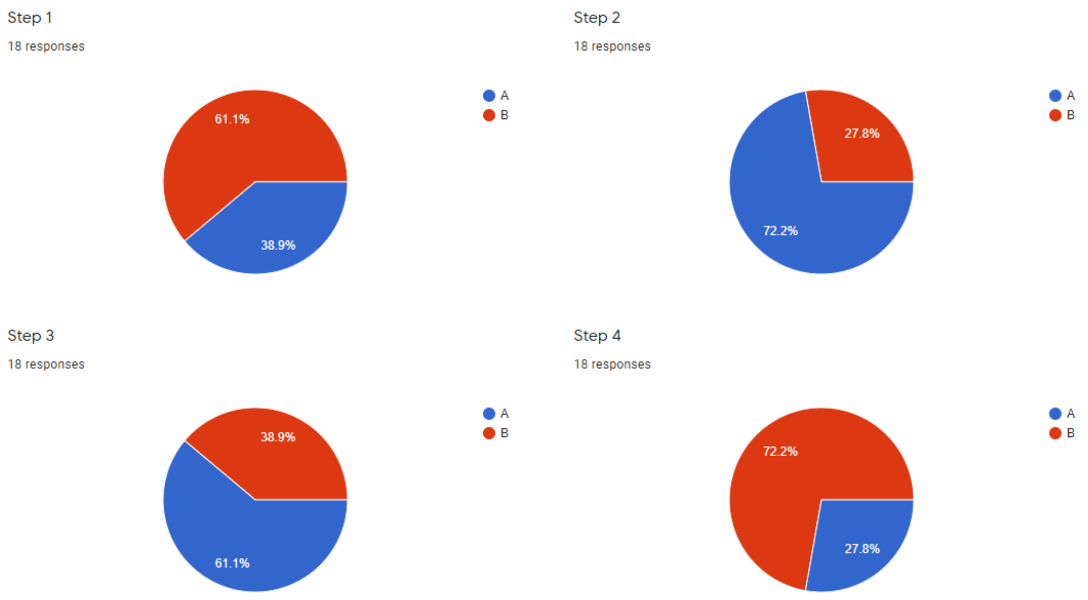



Figure 23. A bundled screenshot of the results of the Lifting scenario's A/B test.

From the results of both scenarios, it can be seen that the choices that used the spatializer plugin were not significantly more popular. Some of the choices were less favored than others. On the other hand, there were far too few participants to draw any concrete conclusions out of these results. Even the plugin was only used to the extent that it's enabled. Therefore, the only features that were used with the plugin are the features that are initially enabled without touching any optional parameters and settings. As it was already mentioned, spatializer plugins can achieve a much more personalized and immersive soundscape. Of course, achieving setups that provide as immersive of a listening experience as they can require a lot more time and effort.

With the results collected from the Lifting scenario's A/B test (Figure 23), we can calculate the significance rate for the Lifting scenario. The same steps were taken here as were taken with the results collected from the Forest scenario's A/B test. Variation A represents the choices that used Unity's built-in spatialization solution and variation B represents the choices that used the Resonance Audio spatializer plugin.

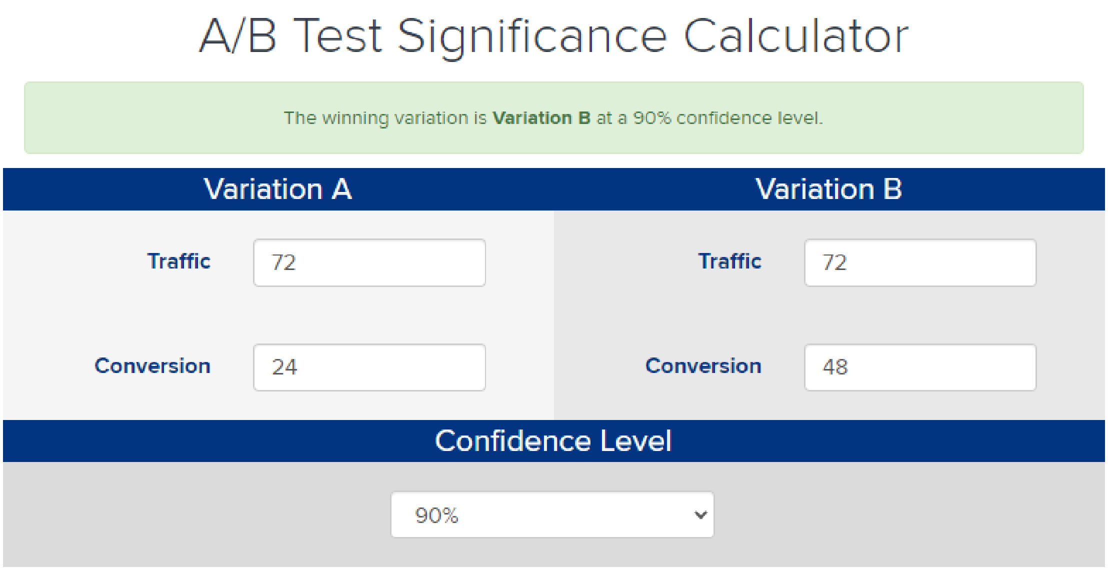



Figure 24. A screenshot of the Lifting scenario’s results inputted to an online A/B test significance calculator.

Figure 25 reveals the significance rate calculated by an online A/B test significance calculator using the results collected from the Lifting scenario’s A/B test. The p-values should not be exactly zero as demonstrated in Figures 22 and 25. However, the reason why it is zero could be due to the online calculator. The number may be rounded to zero simply because it is incredibly small. Also, the calculations are all a bit off because of the small amount of traffic and conversions for both scenarios. For these results to look like more how they should, one would need to increase the sample size, therefore create a larger uplift.

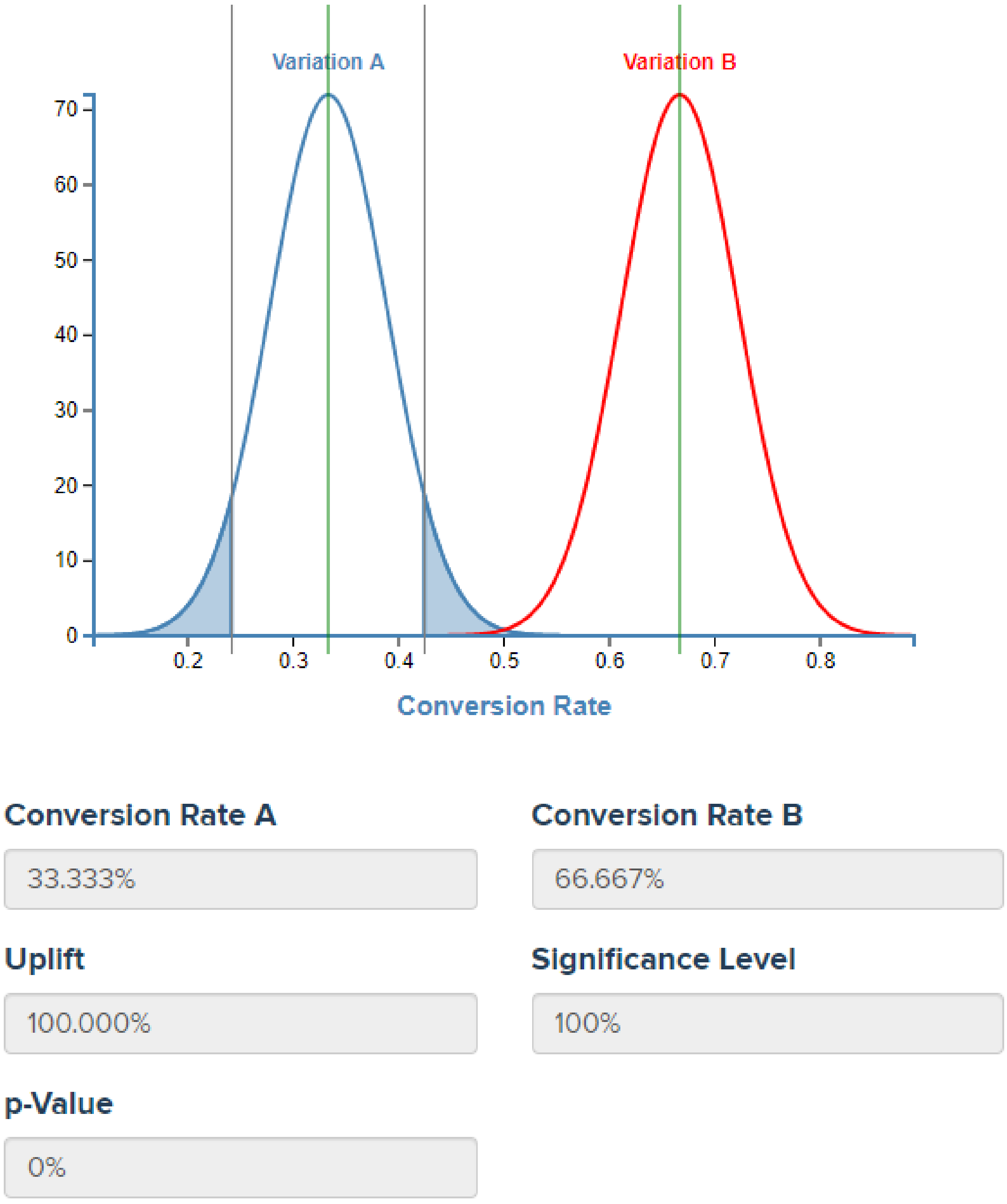



Figure 25. A screenshot of the Lifting scenario’s significance rate calculated by the online A/B test significance calculator.

In addition to collecting test results from the A/B test, the questionnaire also included questions related to the topic in general, such as whether the respondent had initial knowledge and understanding of a particular matter related to the topic or not.

When looking at the results, with just acoustic shadowing and better sound directivity, the difference is not so noticeable compared to the stereo panning. If the audio is only compared within the horizontal plane, Unity's spatialization solution seems to perform well at the required level. At this level, the Unity's solution seems to be lacking only when talking about the elevation of sound. The A/B tests had certain choices that were more favored than others and that is because the sound sources had differences in elevation. Judging by the circle graph in Figure 22, only one (1) participant out of eighteen (18) thought that there was no difference between most of the A/B choices presented in both scenarios. However, this doesn't necessarily mean that all the sounds in the participant's mind had no differences at all. For some sounds, the difference might have been very noticeable while for some the difference might have been barely noticeable if at all. To obtain more accurate results about this, it would be a good idea to ask how noticeable the difference was for each A/B option. Also, increasing the diversity in elevation for the sound sources might affect the results positively meaning that choices using spatializer plugins might be even more favored than they are now.

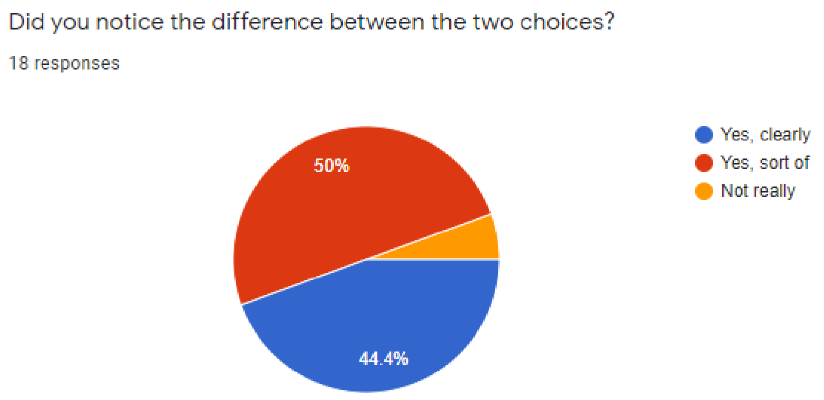



Figure 26. A screenshot showing the results of a question asking did the participant notice the difference between the two choices.

Another question in addition to the results of the A/B test was about previous experience in VR. As the circle graph in Figure 23 shows, the majority of the eighteen (18) participants had never tried VR before. This is a valuable part of the results regarding the whole testing since it will affect any conclusions that can be made from the work. As it was mentioned before, VR is known to be a compelling experience for first-timers. Although, it's good to keep in mind that during the tests all interactions with the environment were disabled. They were deduced to be distractions that could negatively affect the results. In any sort of testing, it is wise to stay in control of all variables that might affect the results. The fewer variables, the more accurate the results.

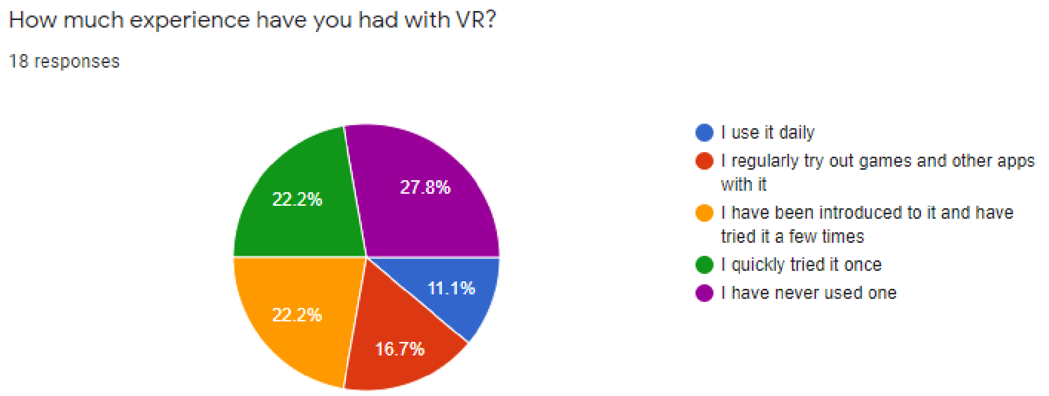



Figure 27. A screenshot showing the results of a question asking how much experience did the participants have with VR.

The final question that is of importance for this work is a question that gathered information about the participants' previous knowledge about the topic. Looking at Figure 24, it can be seen that half of the participants did not have any previous knowledge on the subject. In a sense, this is quite interesting because for the eighteen (18) participants the choices that used the spatializer plugin were nonetheless more prominent.

"Do you know what is spatial audio? Do you know what it means to have interaural time differences? Do you know what is a technology called ambisonics?" If you answered no to all these questions, then you can skip the last four questions.

18 responses

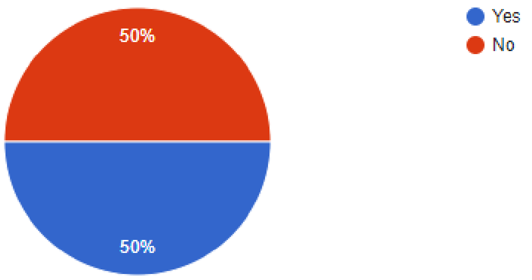



Figure 28. A screenshot showing the results of a question asking did the participant have any knowledge about the topic beforehand.

# 6 CONCLUSION

Despite the small number of responses collected, the outcome is quite positive. The purpose of this work was to prove that achieving more immersive spatial audio with the Resonance Audio spatializer plugin does not require much extra work compared to the spatialization solution provided by Unity. Recalling the test method section, enabling the use of the spatializer plugin does not, in my view, require any more effort than using the features provided by Unity alone. Objects and their components that are intended to play audio three-dimensionally in a virtual space must in any case be prepared whether they use a spatializer plugin or not. So adding the plugin to the equation is, in my opinion, worth the effort. Users' preferences also seem to be leaning towards the solution offered by Resonance Audio even though the work only used it to the extent of simply enabling it with a minimal amount of work. When looking at the results, the majority of the preferences support the conclusion that the plugin should be preferred as the spatialization solution over the solution offered by Unity.

Without belittling other 3D projects, spatializer plugins are like they were created for VR. The features they provide and what they are capable of are perfect for VR applications. The effect of plugins on the immersion brought by spatial audio, and the potential they bring in general, can only be good for 3D projects. Therefore, investing in more advanced spatialization solutions and favoring immersive spatial audio in VR projects is anything but pointless.

# REFERENCES

[1] Stephen Schütze & Anna Irwin-Schütze. *New Realities in Audio: A Practical Guide for VR, AR, MR and 360 Video.* CRC Press, 2018.

[2] Christopher J. Plack. *Oxford Handbook of Auditory Science: Hearing.* Oxford University Press Inc., 2010.

[3] Microsoft's article about head-related transfer functions. Accessed: 30.05.2020. https://www.microsoft.com/en-us/research/project/spatial-audio/.

[4] Lasse Pouru. *The Parameters of Realistic Spatial Audio: An Experiment with Directivity and Immersion*. Bachelor's Thesis, Turku University of Applied Sciences, 2019.

[5] Wikipedia's article about anthropometry. Accessed: 19.05.2020. https://en.wikipedia.org/wiki/Anthropometry.

[6] Resonance Audio's website. Accessed: 19.05.2020. https://resonance-audio.github.io/resonance-audio/.

[7] Valve's website about Half-Life: Alyx. Accessed: 24.05.2020. https://www.half-life.com/en/alyx/.

[8] Unity's reference guide explaining its audio system. Accessed: 31.05.2020. https://docs.unity3d.com/Manual/AudioOverview.html.

[9] World Health Organization's topic about the coronavirus outbreak. Accessed: 02.06.2020. https://www.who.int/health-topics/coronavirus#tab=tab_1.

[10] Jennifer Walinga, Royal Roads University. *Introduction to Psychology – 1st Canadian Edition*. Bccampus, 2010.

[11] A demo video showing the traits of binaural audio. Uploader: BBC Click. Accessed: 30.05.2020. https://www.youtube.com/watch?v=51za5u3LtEc&feature=youtu.be&t=292.

[12] Wikipedia's article about the acoustic shadow. Accessed: 30.05.2020. https://en.wikipedia.org/wiki/Acoustic_shadow.

[13] Wikipedia's article about HRTFs. Accessed: 14.06.2020. https://en.wikipedia.org/wiki/Head-related_transfer_function.

[14] Wikipedia's article about immersion (virtual reality). Accessed: 14.06.2020. https://en.wikipedia.org/wiki/Immersion_(virtual_reality).

[15] Ernest Adams. *Postmodernism and the Three Types of Immersion*. Gamasutra, 2004.

[16] Wikipedia's article about A/B testing. Accessed: 14.06.2020. https://en.wikipedia.org/wiki/A/B_testing.

[17] The questionnaire used in this thesis in Google Forms. *A/B Testing Results and Experiences Survey*. Created: 2020. https://forms.gle/UzduBYbWD94CkGqZ8.