TAMPERE UNIVERSITY
OF APPLIED SCIENCES

Introduction to Sound Design for Virtual Reality Games

A look into 3D audio, spatializer plugins and their implementation in Unity game engine

Julius Nuora

Bachelor's thesis
March 2018
Degree Programme in Media and Arts
Music Production

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Media and Arts
Music Production

JULIUS NUORA:
Introduction to Sound Design for Virtual Reality Games
A look into 3D sound, spatializer plugins and their implementation in Unity game engine

Bachelor's thesis, 47 pages, appendices 4 pages
March 2018

Virtual reality applications have become a considerable field in recent years. With evolving technology, sound design has had to develop and adapt to new requirements as well. Compared to non-VR games, what is new and different when creating sound design for VR games? This thesis is targeted especially for a sound designer working on their first VR game project.

VR games are based on three-dimensionality and immersion and sound design should reflect that. In order to create believable 3D sound design we need to be able to spatialize sound sources three-dimensionally. We as humans are able to localize sounds with our ears receiving the sound and our brains interpreting the information. One of the objectives of this thesis was to understand how spatialization techniques are based on our ability to localize sound sources.

3D audio's characteristics also affect how audio material should be created and implemented in a game. Although there are no strict rules on how to create sound design for VR games, there are some guidelines and experiments about the subject. The objective of this thesis was to look into some general concepts, what they are based on and how the concepts have worked in some previous experiments.

As the practical part of this thesis three spatializer plugins were overviewed and implemented in Unity game engine. This included creating a demo scene in Unity, with which one was able to examine the plugins and their properties. In addition to testing and comparing the plugins, the objective was to create a scenario that anyone could try at home, including walk-through videos describing how to create the demo scene and how to implement the plugins in Unity.

Although the practical study was useful in learning and covering the implementation and comparing the differences between the plugins, it should be concluded that any demo scene is hardly able to reflect an actual, real-life game's sound design process.

Key words: VR, games, sound design, 3D audio, Unity.

CONTENTS

1 INTRODUCTION

In recent years virtual reality applications have become exceedingly common throughout the media field, mostly due to technical advancements and greater accessibility for customers. As the applications, gear and visualization of VR have evolved, audio and sound design have had to meet the new requirements as well. This has meant greater exploration and advancements in 3D and binaural sound design, sound localization and spatialization techniques in order to provide tools for creating a truly immersive experience especially with VR games.

As an introduction to sound design for VR games this thesis covers some basic guidelines, issues, problems and possibilities involved in VR sound design. The objective is to understand the differences between sound design for a non-VR and a VR game, examining the concepts behind 3D audio and spatialization. In a nutshell, what should one know when doing sound design for their first VR game?

Currently there is a lot of information about the subject (mostly online), however quite scattered and sometimes hard to find. This thesis has gathered and filtered the useful information into one package. Nevertheless, it should be noted that some of the information may not stand the test of time due to the constant developments in the industry. On the other hand the theories that 3D audio and spatialization are based on, such as human hearing and localization, are more long-lasting. With some subjects (especially with the implementation of audio) there are no right answers, but rather guidelines and preferences.

Additionally, this thesis includes creating a simple demo scene in Unity game engine, covering the implementation of a few spatializer plugins. This is done in a walk-through style, explaining the process of creating the demo scene and implementation of the plugins, enabling anyone to test them at home (all the plugins and Unity are freely downloadable online). The purpose is in understanding and comparing the properties and effects of the plugins when applied to a sound source.

2 REFLECTING REALITY

Game audio has many functions, not only in creating the atmosphere and mood of the game, but also as a source of information and narration. The role of audio depends on the game from being a minor supportive element to functioning as the driving force of the game. It may be relatively difficult (or at least boring) to play Guitar Hero without audio, whereas many games work well with no audio at all. The development of game industry has widened the scope of games, nowadays one is able to play mobile games anywhere and on the other hand one can have a very immersive gaming experience at home.

Virtual reality games are at the deep end of an immersive experience, the player being isolated both visually (using head-mounted displays, HMDs) and aurally (using headphones). Game designers Salen and Zimmerman (2003, 450) are describing immersion as “reality [is] so complete that ideally the frame falls away so that the player truly believes that he or she is part of an imaginary world.” In many cases, VR games are aiming to create these imaginary worlds by reflecting reality, including the way we hear and are able to localize sound sources three-dimensionally.

There has been studies on the effect of sound in creating immersion, including both the functional aspects of gameplay and the emotional connection to the game world. Still, these studies have been unable to examine why and how sound influences immersion. (Collins 2013, 56-57.) It could be argued that the effects and importance of sound design are best (or worst) exposed when sound design is not on a par with other components, especially visuals. An immersive experience is easily disturbed by sounds that are not in sync with movement, sounds that are not spatialized correctly and so on. With VR games these issues are even more present than before.

2.1. Old ideas, New Implementations

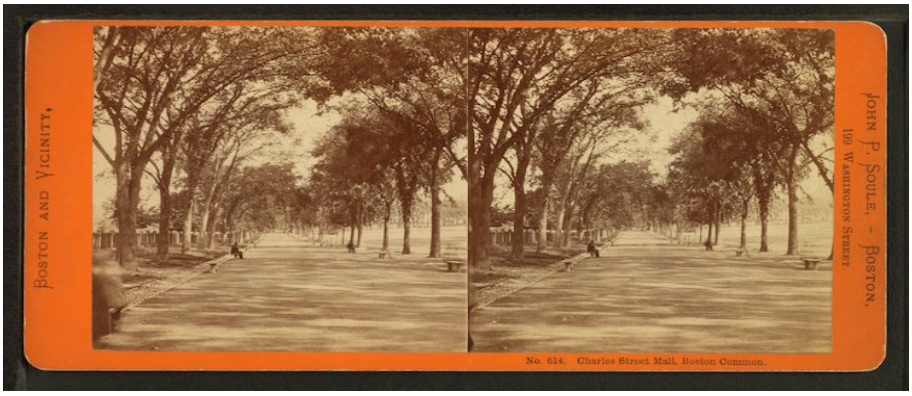
Similarly to many technological advancements, it feels like VR has become a field to be taken seriously in a relatively short time, at least for customers. The big push for current, “modern” VR was Oculus' Kickstarter campaign in 2012, based on a

demonstration of the Oculus Rift prototype HMD (picture 1). Subsequently Facebook bought the Oculus VR system for two billion dollars, creating further attention to VR. Oculus' ability to create functional VR systems for consumers, with support of a range of applications, led to other manufacturers realizing the market possibilities and creating their own VR headsets. (Filmora, 2017.)

PICTURE 1. Oculus Rift prototype 2013. According to a review by Alex Wawro "Oculus Rift VR headset prototype works so well it's a little scary". (PCWorld, 2013)

2.1.1 3D Visuals

However, in a larger scope of virtual reality one could go very far back. Visually, as in creating an illusion of us being present somewhere we are not, there has been efforts since the nineteenth century. 360-degree panoramic paintings and murals were in the forefront of creating virtual reality, filling the viewer's entire field of vision. Eventually, there were more technical ideas introduced. Covering ground for 3D vision (and modern HMDs), in 1838 Charles Wheatstone put forth his research about the brain processing two different two-dimensional images into a single object of three dimensions, presenting the basis for seeing images in three dimensions (picture 2). (Virtual Reality Society, 2017.)



PICTURE 2. Stereoscopy (Virtual Reality Society, 2017)

This same principle, stereoscopy, was later further developed and patented in 1939 by the popular View-Master stereoscope (that some may remember from childhood) (picture 3). The same design principles are used today in low-budget HMDs for mobile devices, such as the Google Cardboard. In the current sense of how we perceive VR, there has been practical efforts since the 1960s, enforced by the developing technology involving computers. The Telesphere Mask (patented in 1960) is regarded as the first HMD. Gradually, more and more advanced HMDs were created, eventually also with head tracking and binaural sound. (Packer & Jordan 2001, 220; Virtual Reality Society, 2017.)

PICTURE 3. View-Master stereoscopes (Wikipedia, 2017; Pinterest 2017). Edited.

2.1.2 3D Audio

Three dimensional audio has many names with varying meanings in different contexts and for different people (Collins 2008, 64; Schmidt, 2015). The terms often also overlap each other. The terminology includes for example binaural, spatialized, stereo expansion, surround, virtual surround, HRTF-processed, positional 3D -audio/sound and so on. This unclarity may in part be because of the surprisingly long history of 3D audio, the first binaural recording dating as early as year 1881 (Kall, 2011).

What has been thought (or at least marketed) as 3D audio, has often only been 360° (or less) surround audio, lacking the ability for the player to localize sound sources from above or below the listener. Additionally, most of the 3D audio applications would lack head-tracking, which has become a standard requirement for positional 3D audio with VR applications. (Schmidt, 2015.)

In gaming context, three dimensional audio was introduced in the 1990s, following the first-person 3D shooter games such as Wolfenstein 3D (1992) and Doom (1993). Three-dimensionality would increase the importance of sound design, for example in stealth games such as Thief: The Dark Project (1998), where localizing sound cues would inform the player of nearby enemies. (Collins 2008, 65.)

Additionally, surround loudspeaker systems would become more common, enabling further use of 360° sound. Some games would eventually also use binaural audio, or update their sound design with binaural audio, such as Counter Strike: Go recently. However, it has not been until VR games that 3D audio is seen as a requirement, a standard of quality, not just as an additional perk.

2.2 What is Different?

As noted, wearing an HMD is visually so immersive that “traditional” sound design would break the immersion. According to sound designer Can Uzer: “foley work has to be more fluid than, say an FPS. Also, it makes everyone touch everything, thus forcing you to create sounds for all the small props.” Wearing an HMD, the player is visually so immersed that audio has to reflect that feeling and perception.

In order to create functional spatialization for VR games, head tracking is essential, thus requiring the use of headphones. Speaker setups and even complex 360 degree speaker arrays are inevitably inadequate for an immersive 3D sound in VR games, due to the movement of the player. There are further drawbacks for speakers because of room acoustics (reverberation and reflections), imprecise imaging (localization) and the lack of elevation. In addition, headphones offer better isolation, privacy and portability, not to mention affordability for the consumer. (Hook, 2014; Oculus, 2017.)

In addition to technical challenges, VR environment raises questions about the role and functionality of non-diegetic sound (sound that has no source in the visible world, often for example narration and music). Due to head tracking, sounds that are locked “inside the head” (such as music in stereo), may break the immersion (Angus & Howard 1996, 114; Hook, 2014). Should the music be mixed and implemented differently, or only use diegetic sound sources for music as well? Many of these issues can be handled in various ways, depending on the project and personal preferences of the sound designer.

The rise of VR applications has been realized throughout the audio community; there are a variety of plugins developed by several companies to meet the new standards. Additionally, there is further understanding on how spatialization affects the entire workflow of sound design. At the time of this thesis being written, there has been new developments and updates on a steady basis, monthly updates and “breakthroughs” in 3D audio. The importance of truly convincing 3D audio is promptly summarized by Oculus' Audio Content Lead, Tom Smurdon: “You can't fake this stuff anymore (2015)”.

3 HUMAN AUDITORY SYSTEM AS THE BASIS OF 3D SOUND

One could argue that a sound designer or mixing engineer is nowadays able to make decent soundscapes even with a lacking knowledge on how certain plugins function, or the terminology involved. That said, plugin presets can be very useful for even the most experienced sound designer (especially in terms of workflow). Nevertheless, further knowledge helps the sound designer to solve problems and more complex situations, enabling more creative solutions. (Modern Mixing, 2013; Recording Revolution, 2011.)

This is no different with 3D spatializer plugins, many of those having settings and options that, in order to make reasonable adjustments, require some understanding about the subject and terminology involved. Although the spatializer plugins used within a game engine or middleware software modify audio somewhat automatically from 2D to 3D, it is useful to understand the fundamental concepts of what the plugins are modelled on.

3.1. Localization

Spatializer plugins are created based on the human auditory system, on our ability to locate sound sources within three dimensions. Localization stands for “the ability to predict the position of a virtual sound source based on nothing but audio”. It depends on the brain's ability to interpret multiple sound cues, further founded on human body and ear geometry and the properties and characteristics of sound waves. (Hook, 2014.)

This thesis does not delve deeper into the fundamentals of sound waves, nor look into the mathematical equations that the sound localization theories are built on. However, the basic relation between a sound wave's frequency and wavelength should be addressed. The relation is based on:

$$\lambda = v/f,$$

where
λ = wavelength (meters)
v = velocity, (sound travelling through air, 343 m/s, in dry air at 20 °C)
f = frequency (hertz)

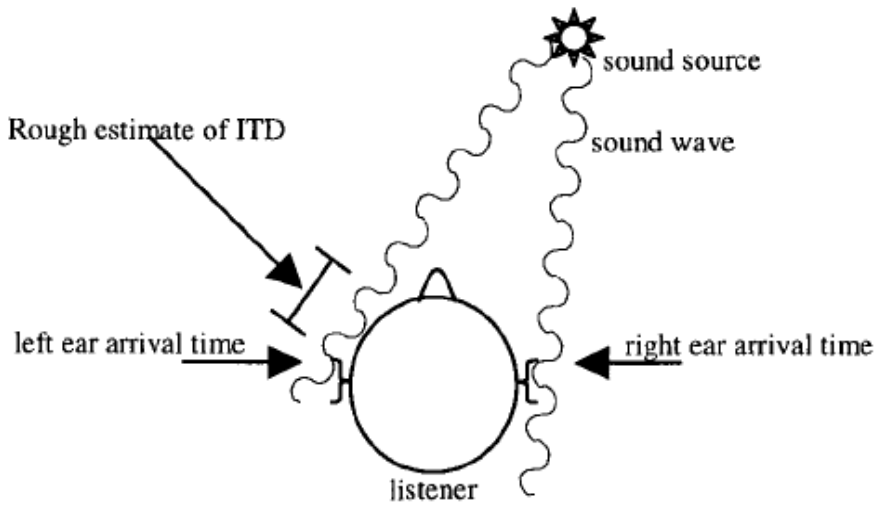
For example a sound with the wavelength of an average human head (0,18m) has a frequency of 1906 Hz. Sound waves with higher frequencies have shorter wavelengths and lower frequencies have longer wavelengths. This relation causes differences in how we localize sounds with different frequencies.

3.1.1 Direction

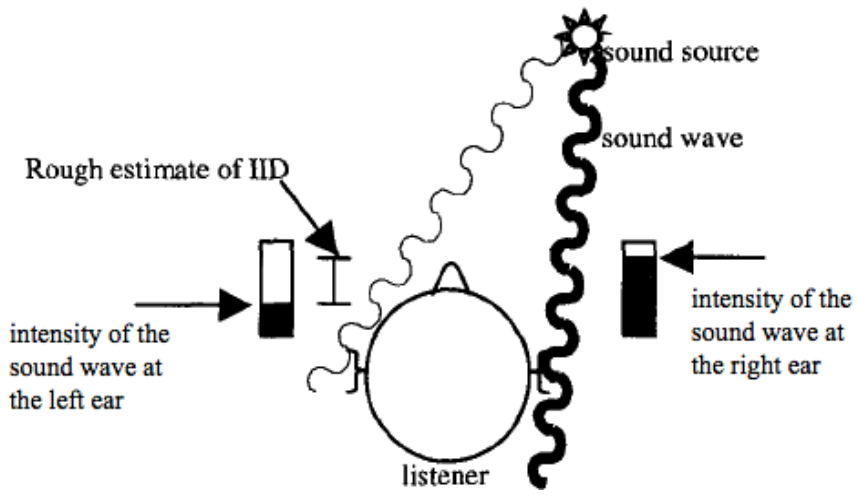
Considering sound localization, direction and distance of the sound source are the two key components. The separation of our ears causes differences in time and level at which the sound signal arrives at each ear. These cues are called interaural time difference (ITD) and interaural level difference (ILD). (Angus & Howard 2006, 111-113; Hook 2014; Oculus, 2017.)

For example, due to ITD (the time difference) a sound arriving from the left will be received first by our left ear (picture 4). If the sound is directly in front of us, it will arrive at both ears simultaneously. On the other hand, there can be differences in the level (intensity) that the sound reaches the two ears (ILD) (picture 5). Although sound wave's strength (loudness) dissipates as it travels, the level difference is mostly because of head shadowing (picture 6). (Angus & Howard 2006, 111-113; Hook 2014; Oculus, 2017.)

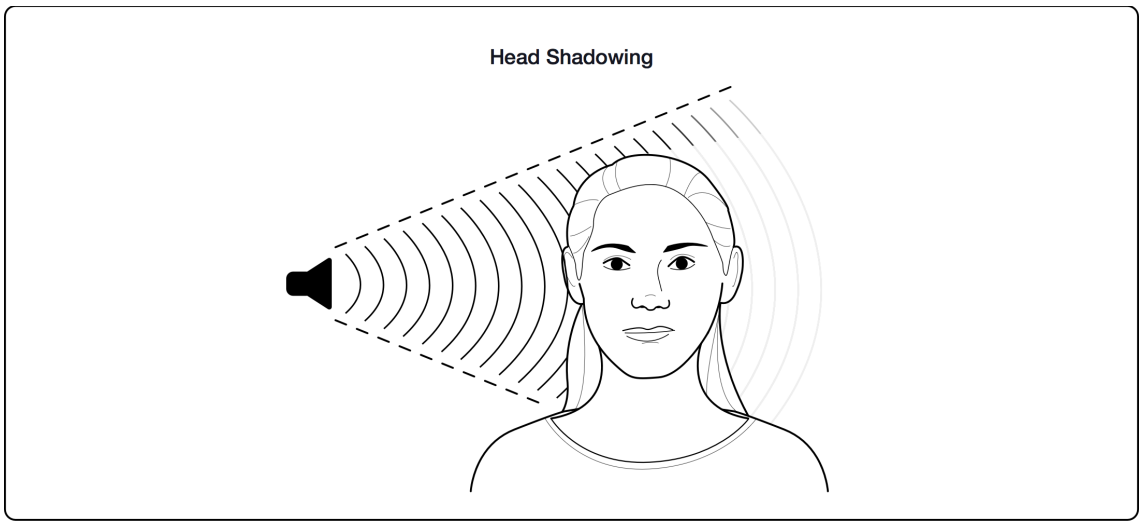
Depending on the frequency, sound may be obstructed by the head, having to travel through and around the head to reach an ear, so that there can be a significant attenuation of overall intensity of the sound signal in one ear. The sound arriving from the left may be louder for our left ear, from in front of us equally loud to both ears, and so on. Which one of the localization techniques (interaural time or level difference) is used is determined by the sound signal's frequency content, based on the differences in wavelengths. (Hook 2014; Oculus, 2017.)



PICTURE 4. Interaural time difference (ITD) (archive.cnx.org)



PICTURE 5. Interaural level difference (ILD) (archive.cnx.org)



PICTURE 6. Head shadowing (Oculus, 2017)

Due to the size of an average head, and the fact that an object (in this case the head) needs to be about two thirds of a wavelength in order to obstruct the sound signal, there is a minimum frequency of about 650 Hz below which ILD is no longer useful for localization. Therefore time difference (ITD) is used to localize the low frequency sounds (generally under around 800 Hz). (Hook 2014; Oculus, 2017.)

On the other hand, using ITD, the ear resolves the direction by using phase shift which is caused by the time difference of sound arriving to the ears. As the frequency gets higher, the wavelength shortens and at above approximately 1500 Hz the phase information is no longer reliable for localizing the sound (due to the wavelength being shorter than the head). Therefore level difference (ILD) is used to localize the high frequency sounds. Between around 800 Hz and 1500 Hz, there is a transitional zone in which both time (ITD) and level differences (ILD) are used for localization (table 1). (Hook 2014; Oculus, 2017; Schmidt 2015.)

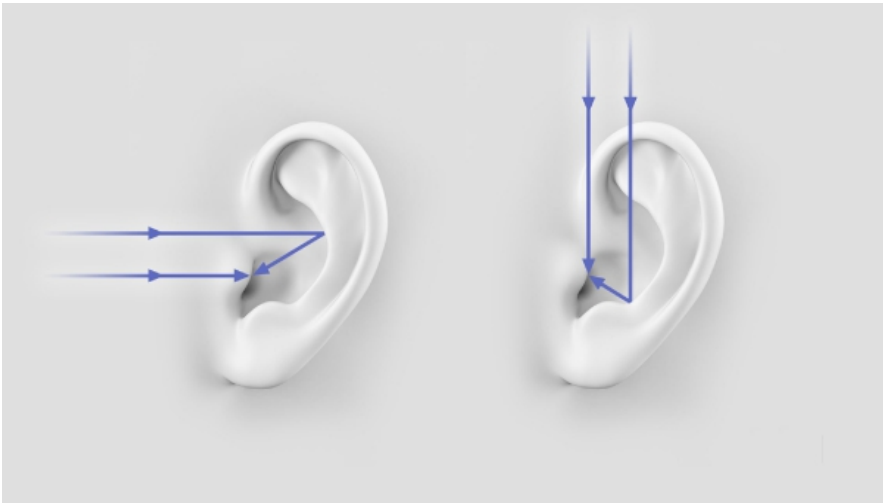
TABLE 1. Human sound localization (lateral), which one of the cues is used.

FREQUENCY	**HEAD SIZE**	**CUE USED**
80-800 Hz	< wavelength	ITD
800-1500 Hz	= wavelength	ITD +ILD
above 1500 Hz	> wavelength	ILD

However, since interaural time and/or level differences may be zero in front or back, these techniques are only covering the lateral localization. In order to achieve three dimensional localization, head movement and spectral filtering by our bodies and ears provide further information. (Angus & Howard 2006, 114; Hook 2014; Oculus, 2017.)

The ability to localize and distinguish front, back and elevation (up/down) is based on the geometry of the human head and body and especially on the shape of the outer ears (pinnae). They create spectral modifications to arriving sounds, creating cues with which the brain can localize the sound. For example, sounds coming from behind are more muffled (less high frequencies). (Hook 2014; Oculus, 2017.)

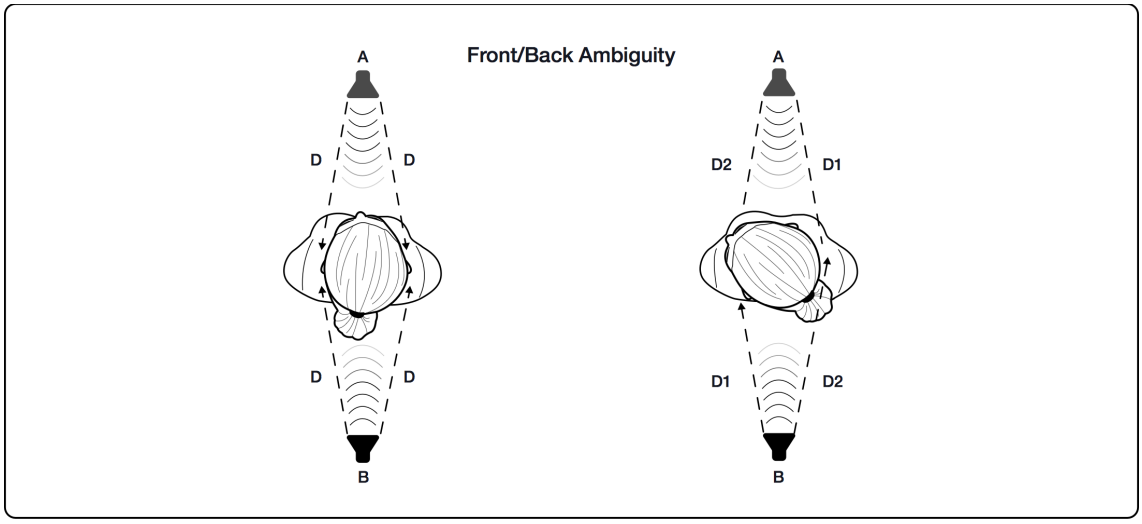
As sound hits the pinnae, it is being reflected to the ear canal by the complex set of ridges that exist on the ear. These pinnae reflections create filtering that helps resolve the ambiguities localizing the sound (picture 6). Additionally, the human head, shoulders and torso create cues for localization by reflections, shadowing and filtering. (Angus & Howard 2006, 113; Hook 2014; Oculus, 2017.)



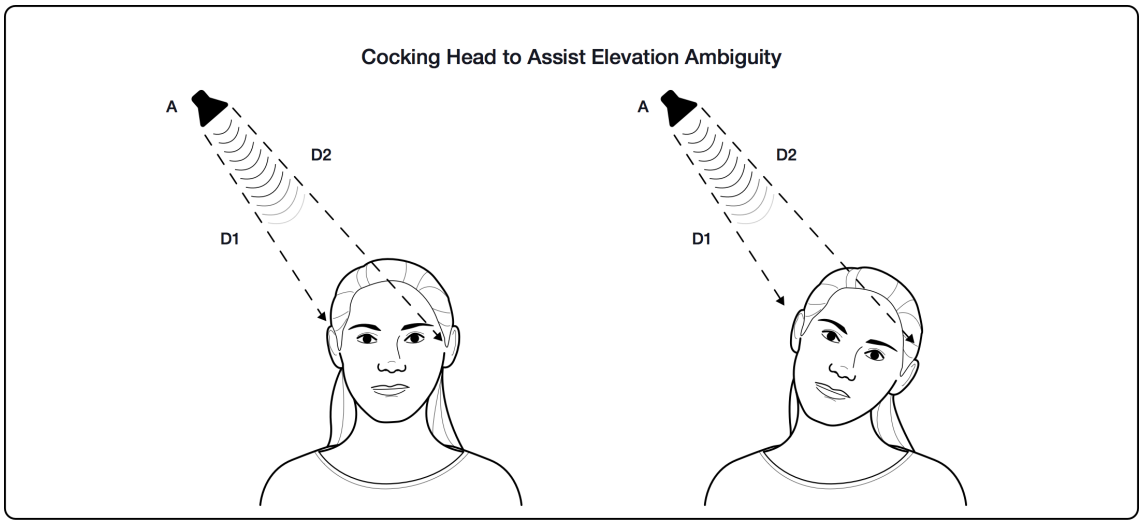
PICTURE 6. A simplified image of pinnae reflections and spectral filtering (Google, 2017)

Naturally, the effects of human geometry in localizing sound sources are person specific, as we have varying shapes of ears and bodies. We learn to interpret these cues as we grow up, but sound recorded through other people's ears may cause confusion. The more effective (and obvious) solution to resolving directional unclarity is to move our heads. By turning our heads, the front/back ambiguity becomes a lateral localization problem which we are better equipped to solve. A sound from behind will move in a

different direction compared to a sound in front of us (pictures 7.1 and 7.2). (Angus & Howard 2006, 114; Hook 2014.)



PICTURE 7.1. Front/back ambiguity (Oculus, 2017)



PICTURE 7.2. Movement cue (Oculus, 2017)

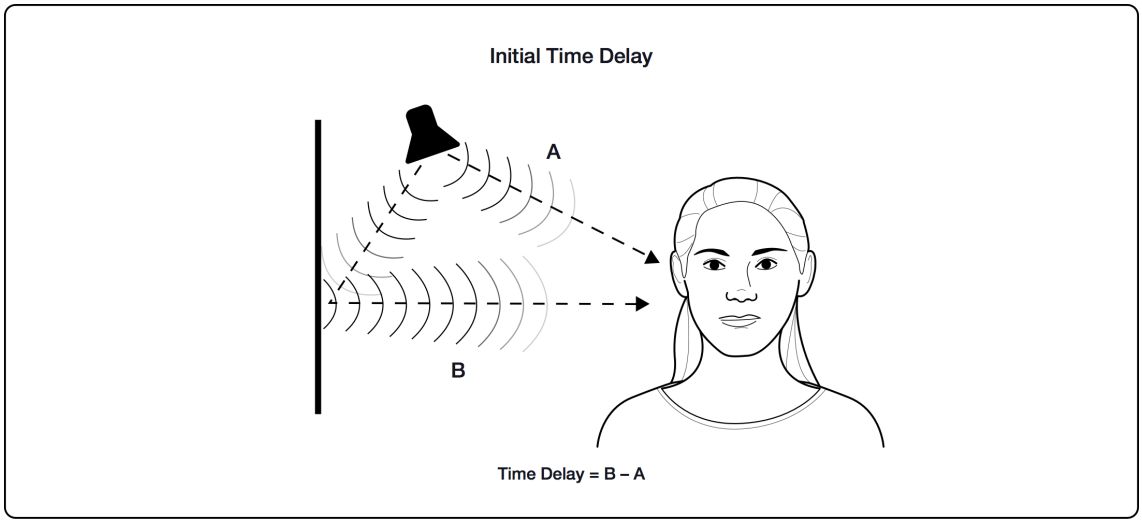
This subject of movement cues becomes central in VR sound design. Using headphones and listening to a sound source that is not connected to head movement creates a feeling of the sound being "in our heads" (Angus & Howard 1996, 114; Hook, 2014). For example music or ambience tracks that would work well as basic stereo files in a non-

VR game face trouble in VR environment as they are breaking the immersion of three dimensionality. Solutions to this issue will be examined later in chapter 4.

3.1.2 Distance

ITD and ILD -techniques do not fully explain how we are able to determine the distance of a sound source. The loudness of the sound source is clearly a big factor, but that is only true for familiar sound sources. For sound sources that we do not have a frame of reference with (synthetic or unfamiliar sound sources) we have to rely on other cues such as initial time delay, reverberation, motion parallax and high frequency attenuation to predict the distance and whether a sound is approaching or receding. (Hook 2014; Oculus 2017.)

Initial time delay stands for the time difference of the direct sound and its first reflection. If a person is talking next to you, the direct sound will arrive sooner than the reflections, compared to a person talking farther away when the direct sound and reflections could arrive around the same time. We can assume that longer the delay, the closer we are to the sound source (picture 8). (Hook 2014; Oculus 2017.)



PICTURE 8. Initial Time Delay (Oculus, 2017)

Related to this is the room ambience, the reverberation mixing with the direct sound. Based on the mix of direct and indirect sound (wet/dry mix), we are able to draw

conclusions about the distance. Additionally, distance is indicated by motion parallax, the movement of a sound source through space. If a sound travels quickly from our perspective, we tend to perceive it coming from nearby. (Hook 2014; Oculus 2017.) This depends on the sound source however, for example a jet fighter may be far away even though the sound travels very quickly from our perspective, due to the jet's speed.

We are also able to determine distance based on the attenuation of high frequencies because of the damping of sound moving through air (dependant on the humidity and temperature). This is however to a lesser significance especially in interior scenarios and most gaming situations since the high frequency attenuation takes effect only on relatively long distances. (Hook 2014; Oculus 2017.)

3.2. Spatialization

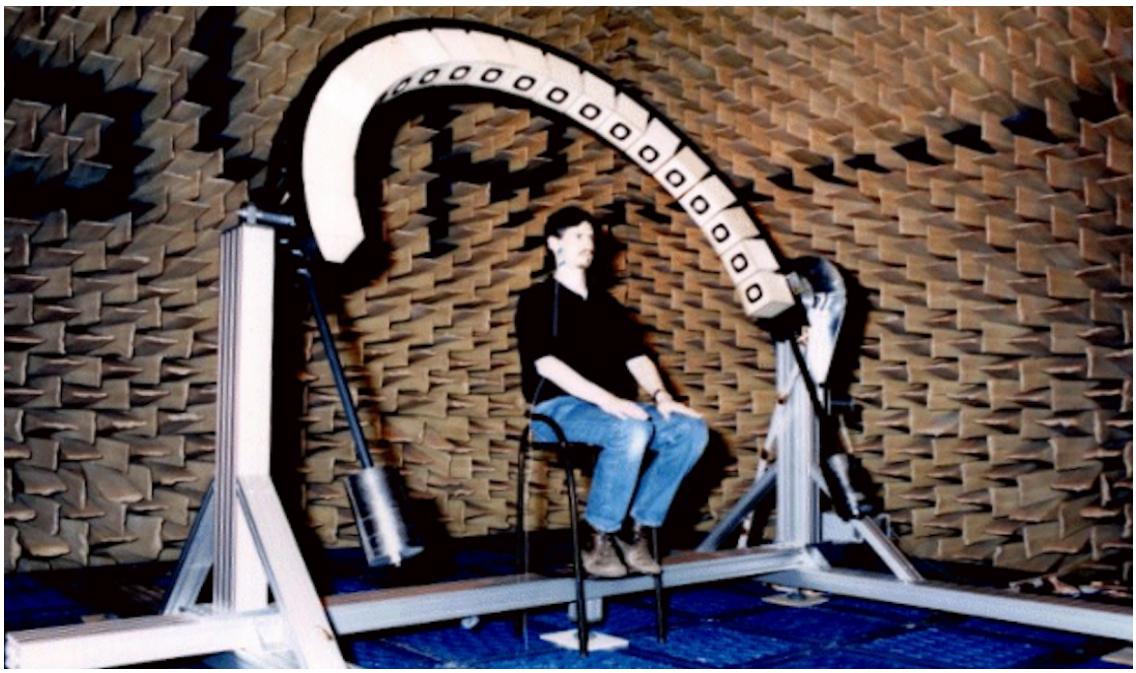
Spatialization stands for inverting the information we have about localization, and applying it to create an illusion of a sound coming from a specific point in space. Applying these techniques also has an effect on how sounds need to be designed, considering audio material and implementation. (Hook, 2014.)

There are simple methods for spatialization that have been used in earlier times and especially before modern HMDs, such as panning for left and right, low pass filtering (LPF) for front and back and volume attenuation for distance. Elevation and head tracking have been basically non-existent. More accurate spatialization has not been regarded that important due to the range of variety that the gamers could have as their audio or speaker setup and listening spot. Nowadays there are, however, more advanced methods available for creating spatialization. The foundations and scientific knowledge of these methods have been around for a longer time, but with VR implementations, 3D spatialized sound and the advanced methods have become more relevant than before. (Hook, 2014.)

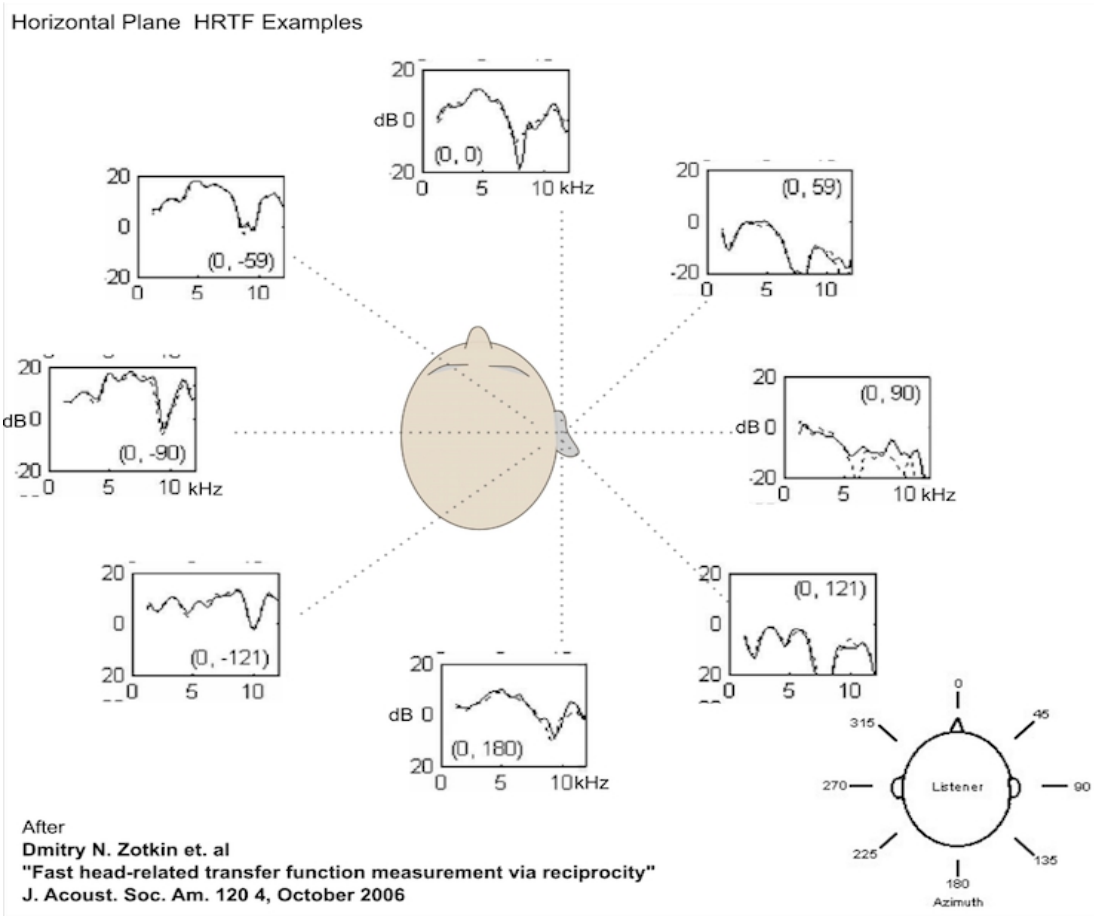
3.2.1 Modelling Directionality

As mentioned, our ability to locate the direction of an arriving sound is based on our body and ear geometry. The response that characterizes how an ear receives a sound from a point in space is called a head-related transfer function (HRTF). 3D sound spatializer plugins are based on the encoding of HRTF and it is basically the cornerstone for most of the modern sound spatialization techniques. (Oculus, 2017.)

HRTF is a complex function defined for each ear, all of us having individual HRTF responses. The most accurate way of capturing an individual HRTF is having the person in an anechoic environment, putting a pair of microphones in the person's ears, playing sounds from every direction required, and record the sounds with the microphones (picture 9). These recordings are then compared to the original sound and computed to HRTF, for both ears separately (picture 10). (Oculus 2017; Potisk 2015.)



PICTURE 9. Measuring HRTFs in an anechoic chamber (Code & Sound)



PICTURE 10. An example of HRTF frequency responses in horizontal (lateral) plane for right ear (recorded using a KEMAR dummy-head mannequin). There are drastic variations especially in the high frequency area. (Beware that the frequency axis in these graphs is definitely not linear, 5 kHz is around in the middle.) (recordingsofnature.wordpress.com, 2015)

The optimal idea would be to have a personalised HRTF for every specific person. However, this is not really practical in reality (one would have to do HRTF measurements for every individual), so the spatialization plugins use a generic reference set that is adequate for most situations. There are publicly available databases of HRTF measurements that are captured either with human subjects or by using synthetic head models (dummy heads). (Oculus 2017; Potisk 2015.)

3.2.2 Modelling Distance

HRTFs provide us with tools to spatialize the sound's direction, but they do not model distance. However, the same principle of using localization techniques inversely applies to distance modelling as well. These include loudness, initial time delay, the mix of direct and indirect (reverberation, wet/dry mix) sound, motion parallax and high frequency (HF) attenuation (absorption by air). (Oculus 2017.)

Loudness is relatively simple to model, with sound level attenuation based on the distance. Motion parallax does not require specific attention, since it is created "automatically" as a byproduct of the velocity of a sound source. Additionally, HF attenuation is quite easy to model by applying a low-pass filter and by adjusting the cutoff frequency and slope. (Hook, 2014; Smurdon 2015; Oculus 2017.)

Initial time delay and the wet/dry mix offer more of a challenge, with more variables depending on the given set of geometry. The early reflections require complex computing based on the environment, its geometry and characteristics (acoustics). Nevertheless, several spatializer plugins offer means to create detailed, full scene geometric modelling, or at least simplified "shoebox models". These systems also offer the wet/dry mix of direct and reverberant sound as a natural byproduct of the modelled geometry. (Hook, 2014; Smurdon 2015; Oculus 2017.)

4 DESIGNING AND MIXING SOUND FOR VIRTUAL REALITY

There are guidelines and principles about VR sound design, though they provide only a general starting point, not “rules” to be taken for granted. A good example of the lack of general rules (and constant development of the industry) is that there are some varying guidelines even from different Oculus' sources, depending on the date or the individual providing the information. The “best” way of designing and mixing sound for VR varies between scenarios and individual preferences. (Silventoinen, 2017.) It should be noted that in this context, the term “mixing” rather refers to the implementation of sound in the game or middleware software than mixing audio in a DAW.

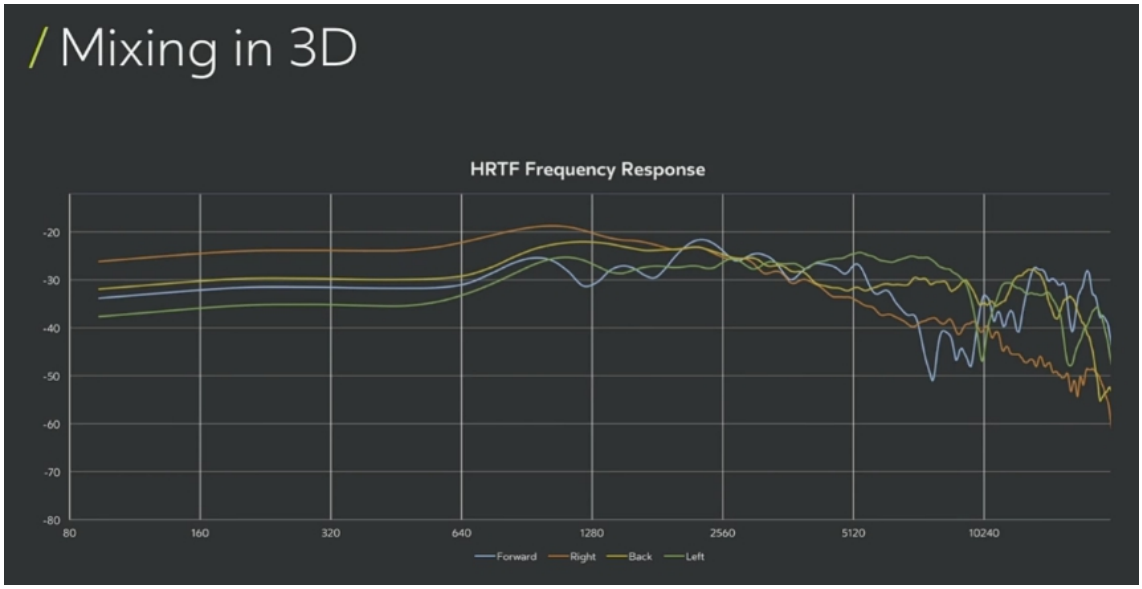
4.1. Audio Material

The fact that a sound will be implemented in VR environment should be taken into account early on when creating and modifying the audio material. One should also consider which of the audio sources actually require spatialization. Most of the material should be mono because of head tracking, HRTF and the issue of stereo sound breaking the immersion (the feeling of sound coming from “inside the head”). (Oculus, 2017.)

Applying spatialization, the HRTF frequency response creates a lot of coloration, which may be quite radical, even 10-20 dB notches at some frequencies (picture 11). In the traditional sense of audio mixing, one would equalize the sounds so that different instruments or audio components would not fight over the same frequencies. With 3D mixing however, one is able to separate components by placing them in different locations in physical space, thus making one able to hear them through the mix. (Oculus 2017; Smurdon & Stirling, 2015.)

That said, the sounds should have a wide frequency spectrum for HRTF to work as well as possible. Humans are able to localize high frequencies (HF) better (especially above 1500 Hz), and on the contrary, localizing low frequency (LF) rumble can be difficult, even so that some LF sounds do not need to be spatialized and using panning or attenuation is enough. A more practical reason for using wide frequency spectrum is that it also helps masking audible glitches that may occur due to dynamic changes

involved with panning, attenuation or HRTF processing. (Oculus 2017; Smurdon & Stirling, 2015.)



PICTURE 11. Frequency response of white noise running through a spatializer, based on an HRTF response of a human subject. The Y-axis values are between 10 dBs so the changes above 5kHz seen here are quite heavy. (screenshot from video lecture by Smurdon & Stirling, 2016)

One should also avoid spatializing pure tones such as sine waves, since they lack harmonics or overtones, making them difficult to spatialize as HRTF functions by filtering frequency content. The lack of frequency content also makes pure tones more vulnerable to glitches, especially having the sound implemented in a moving object. (Oculus 2017; Schmidt 2017.)

Not all sounds need to be spatialized. These include the aforementioned LF rumble, and especially non-diegetic sounds, for example user interface elements and narration. Additionally, some sounds are relative to head tracking, such as body sounds like breathing and heart beats. One may also have to prioritize on what to spatialize due to limitations of processing power (CPU) available. (Oculus 2017; Schmidt 2017.)

4.2. Sound Mixing

Although there is the underlying concept of maximum immersion in VR sound design, usually based on the idea of reflecting the real world, it should be noted that realism (in terms of physics and how sound waves actually function) is not necessarily what works best for a game. In some cases one has to bend the rules of physics to create a more immersive and exciting experience. (Oculus 2017; Smurdon & Stirling, 2015.)

This does not concern only the sound material but also implementation issues such as dynamic range, attenuation curves and direct time of arrival. For example, to create excitement, one could hear a monster bellowing afar, even though one would not actually be able to hear it in reality because of the distance. (Oculus 2017; Smurdon & Stirling, 2015.)

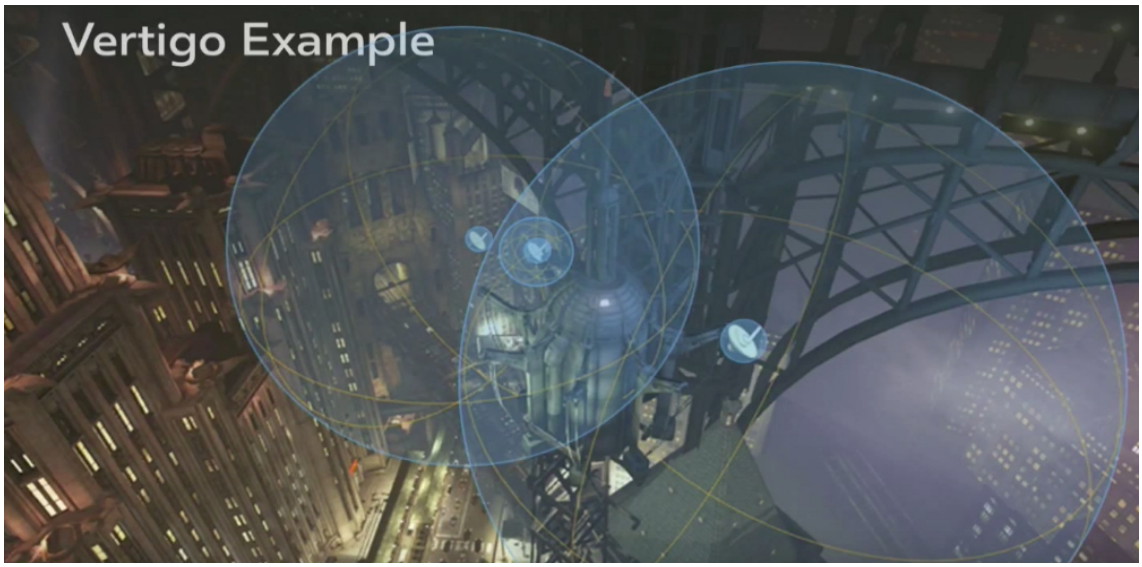
On the other hand, because of more specific localization and especially elevation, one may have to pay more attention to the placement of sounds. In 2D lateral panning one would place footsteps on the root of an object (for example on the hip of an NPC), however with 3D sound and localization this may not be accurate enough. Placement can be taken further towards realism by dividing a sound effect to frequency lanes and placing the lanes accordingly. For example with a creature roar sound, the high frequencies would be placed on the mouth and low frequencies on the body of the creature. (Oculus 2017; Smurdon & Stirling, 2015.) This said, experiencing a VR game can be quite overwhelming with the visuals alone, so one should be careful not to excessively distract the player with audio too much.

4.2.1 Ambience

Ambience usually includes a basic background track that “glues” the audio together, for example room hum (interior) or wind (exterior) that would be stereo in a non-VR game. A stereo track is locked “inside the head” which may pose a problem with VR, breaking the immersion. Problems occur especially if the background tracks include noticeable sound sources such as birds or crowd noises. (Oculus, 2017; Smurdon & Stirling, 2015.)

One solution to this is to avoid noticeable sound sources in the background tracks and place and spatialize them as separate mono files in the scene. Combined with this, a stereo loop of low hum in the background may work well enough without breaking the immersion. Then again some, for example Tom Smurdon (Audio Content Lead at Oculus) (2017), argue that one should entirely discard the idea of using stereo loops in any cases.

Avoiding stereo, one may use multiple mono tracks for the background ambience, placed around the player. This works well especially if the ambience needs to stand out at some point, becoming more than just a “glue”. An example of this method is presented at Oculus' “Vertigo Example”, in which the player stands on a platform high above the ground. As the player gets closer to the edge, a heavy wind sound starts to blow a lot harder, creating an immersive experience looking down the edge (picture 12). (Smurdon, 2015.)



PICTURE 12. Vertigo Example. The ambience includes three mono wind loops. One of the loops in front of and another behind the player, with large radiuses. The third loop is on the edge of the platform, with a small radius, creating a “wow-effect” as the player gets closer to the edge. (Screenshot from the video lecture by Smurdon, 2015)

Another solution for background ambience is using ambisonic audio. Ambisonics is a 3D surround sound technique and a large field of research on its own. It uses the same principle as the M/S (mid/side) stereo technique, but with additional signals (of figure 8

polar pattern) to create the elevation and front/behind. With VR, this proves useful since an ambisonic signal set is not head-locked, enabling head-tracking (as long as the plugins used are supporting ambisonics). An ambisonic set is also practical to implement as it is based on the decoding of one multi-channel audio track, instead of using separate mono tracks. (Blue Ripple Sound, 2015; Smurdon & Stirling, 2016.)

4.2.2 Music

It has been argued that music, even if non-diegetic, should be spatialized as well or in some way adapted to three dimensional VR environment, instead of sticking to a 2D stereo mix. The “safest” way would be using diegetic music only. According to sound designer Aki Silventoinen (2017), especially with video applications it has been noted that even though music would not be head-locked stereo, it might still break the immersion. Nevertheless, similarly to ambience, there has been suggestions about placing multiple mono sources around the player, and most recently, implementing music as ambisonic files.

In the trailer for “Henry”, a game by Oculus, the music was broken apart into stems (different instrument sections) and placed around the player in an arc. This method was found functional when the context was a black room, having “phantom sources” (non-diegetic, non-visual) playing music for the player. However, when the room was visible, the phantom source issue became more noticeable. (Bible, 2016.)

The same method was tried in a demo by Oculus called “Audiobot”, with quite similar results, including the phantom source issue. There were multiple (8-9) phantom sound sources around the player, playing orchestral music separated as different orchestra sections (picture 13). According to Tom Smurdon, although this was “cool”, the problem was that as the player moved around, one would hear sounds coming from a specific point in space, yet nothing visible actually being there. (Smurdon 2015; Smurdon & Stirling, 2016.) Later in the 2017 Oculus Connect lecture, having developed new ways of implementing music, Smurdon commented about the amount of music tracks: “in Audiobot we gave nine [tracks], that's too many” (Smurdon & Stirling, 2017).

PICTURE 13. Audiobot demo (screenshot from the video lecture by Smurdon & Stirling , 2015)

The approach of using fewer point-source mono tracks was tested in Oculus' “Farlands”, which has the music mixed in quad (four channels), instead of stems based on instrumental sections. Based on an idea of a compass, these four mono channels were placed into the scene in four static directions, direction depending on head tracking. The sounds sources' levels were however staying constant, not depending on how close the player would be to the sound source (picture 14). (Smurdon & Stirling, 2016.)

PICTURE 14. Farlands, a compass for four music channels (screenshot from the video lecture by Smurdon & Stirling, 2016)

Naturally this quite unorthodox implementation of music, having the player change the directionality, would affect the music's mix as well. The mix should stay relatively constant between the four channels, not having huge dynamics between the channels or instruments dropping out in a channel without being replaced by another instrument. (Smurdon & Stirling, 2016.)

In the end of 2016 Oculus introduced support for first-order ambisonics in the AmbiX format. In the case of music this would mean that the quad mix could be implemented in a game as an AmbiX file instead of four separate and static point-source mono tracks. Additionally, ambisonics create a smearing effect, reducing the phantom source effect. (Oculus 2016; Smurdon & Stirling, 2017.)

5 OVERVIEW AND COMPARISON OF SPATIALIZER PLUGINS

This chapter covers the implementation of a few spatializer plugins by different manufacturers using Unity game engine. The focus is in covering the features and differences between the plugins and how to implement them in Unity, not so much in how to make actual sound design for a VR game. This may require some knowledge about Unity's basic functions, however nothing that one could not learn from manuals and instructional videos in one evening.

Instead of using sample projects, this overview is based on a demo scene in Unity that anybody can create from scratch. This facilitates understanding the process and (for more practical reasons) avoids problems that may occur between different versions of sample projects, Unity and spatializer plugins. The idea is that anybody can test the plugins at home; all of the plugins and the basic version of Unity are free of charge.

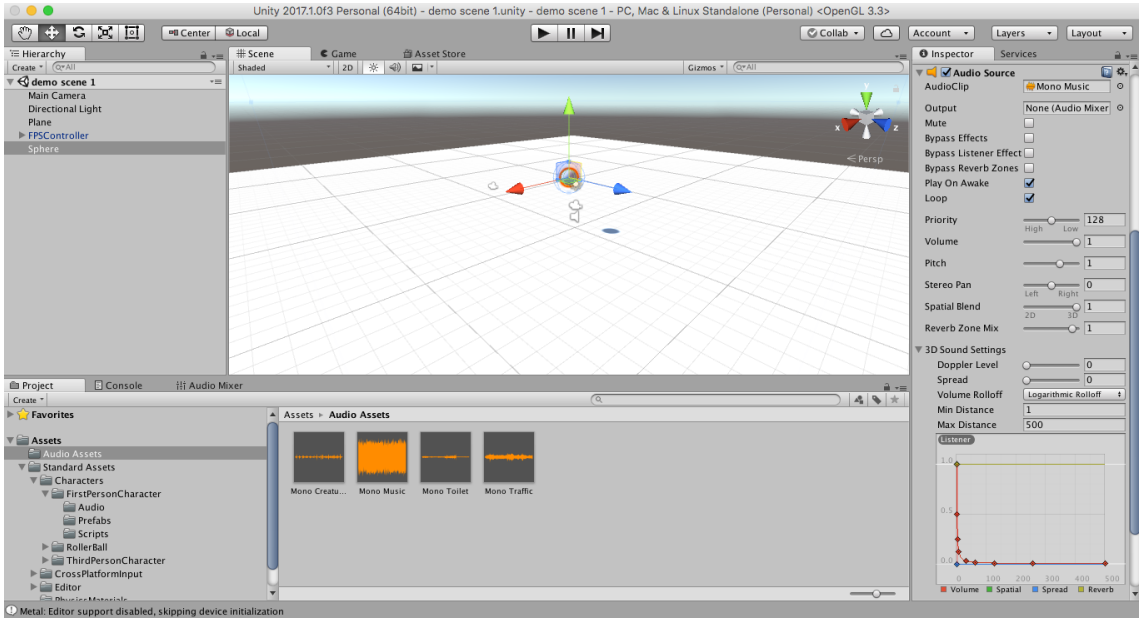
The scene does not require the use of VR-glasses, instead focusing on the audio and implementation. All of the information is available at the manufacturers' websites, however this chapter provides the basic information in creating the demo scene that is used here and making spatializer plugins work in Unity. In order to differentiate Unity terminology from other text, terms such as Audio Listener and Main Camera are written in capital letters.

5.1. Overview

Although there are differences between spatializer plugins, they have similarities in their implementation in Unity. With this very simple demo scene, one is able to hear and experiment how the spatializer plugins and HRTF processing affect a sound. In the demo scene there is a Plane (surface), an FPSController (first-person perspective player) and a Sphere (a ball). The Sphere will be the sound source and the FPSController will be the player. The following paragraph explains the process in detail. There is also a walk-through video by the author in Youtube (click here, full link at appendix 2).

To create the demo scene (picture 15), one is to

- create a Plane (surface) via “GameObject → 3D Object → Plane”
  - adjust the scale/size of the plane to one's liking (in this demo x=8, y=8, z=8)
- import Characters via “Assets → Import Package → Characters”
  - drag and drop the FPSController into the scene
- create a Sphere (a ball) via “GameObject → 3D Object → Sphere”
- choose Main Camera and in the Inspector untick the Audio Listener
- choose FPSController and in the Inspector “Add Component → Audio Listener”
  - tag the FPSController as Player
- create a folder “Audio Assets” in the Project window via “Create → Folder”
  - import whatever audio files one wants to use in the scene, however only mono files are fitting to represent HRTF and its functions
- Add an Audio Source to the Sphere in the Inspector window via Add Component
- add an audio file to the Sphere in the Inspector window under the Audio Source's AudioClip by clicking its tiny ball next to the name of the audio file (in default “None (Audio Clip)”)
  - under Audio Source, adjust the Spatial Blend to 3D (value 1).



PICTURE 15. Demo scene (screenshot of Unity by Julius Nuora, 2017)

5.1.1 Oculus

In addition to advancing the entire VR industry, Oculus has been in the forefront in the development of 3D audio implementations and spatializer plugins. The company is constantly improving their knowledge about VR sound design, with new advances represented at the Oculus Connect event every year. Oculus also has freely downloadable audio packages at their website, including ambisonic soundscapes. Similarly to the demo scene, there is a walk-through video by the author on how to implement the Oculus plugin in Youtube (click here, full link at appendix 2). The process is also described in the following paragraph.

When implementing the Oculus plugin in Unity, at first one needs to download the Oculus spatializer (Oculus Audio SDK Plugins) at Oculus' website. To implement the plugin in Unity, one is to

- import the package via "Assets → Import Package → Custom Package"
  - choose the "OculusNativeSpatializer.unitypackage" in the folder one has downloaded it
- "activate" the plugin via "Edit → Project Settings → Audio", where one can choose in the Inspector which spatializer plugin to use (picture 16).

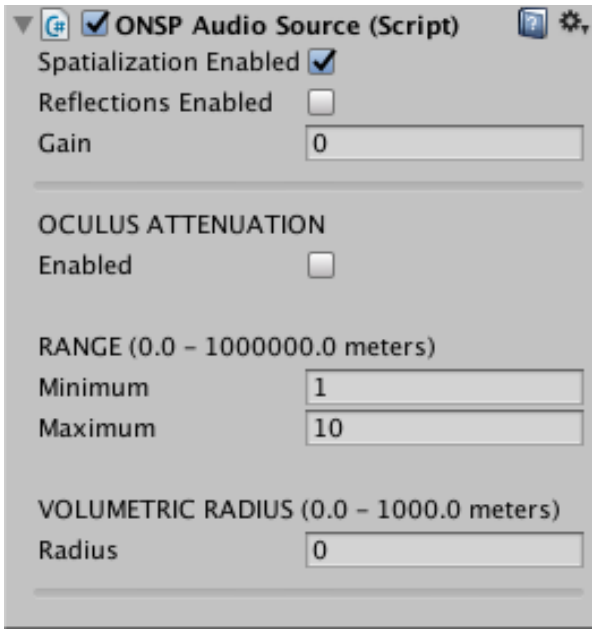
PICTURE 16. Choosing the spatializer plugin (screenshot of Unity by Julius Nuora, 2017)

To apply the plugin into the sound source (in this case the Sphere), one is to

- Add Component (in the Inspector window), selecting the “ONSP Audio Source”
    - most easily found by searching for “ONSP” which stands for all the plugins included in the Oculus spatializer package.

The ONSP Audio Source plugin includes certain options (described in more detail in Oculus' website), which can be adjusted to fit the needs (picture 17). The plugin itself is however applying the HRTF automatically into the component as long as the “Spatialization Enabled” is ticked.

The effects of the plugin can be heard instantly, initially as lower volume of the sound source, which is a common effect when applying HRTF (one can add Gain in the plugin window). With further listening one can hear the effects of HRTF as one moves around the Sphere and compares it with the original sound (also during Play Mode by ticking and unticking the “Spatialization Enabled”, using Esc to free the cursor).



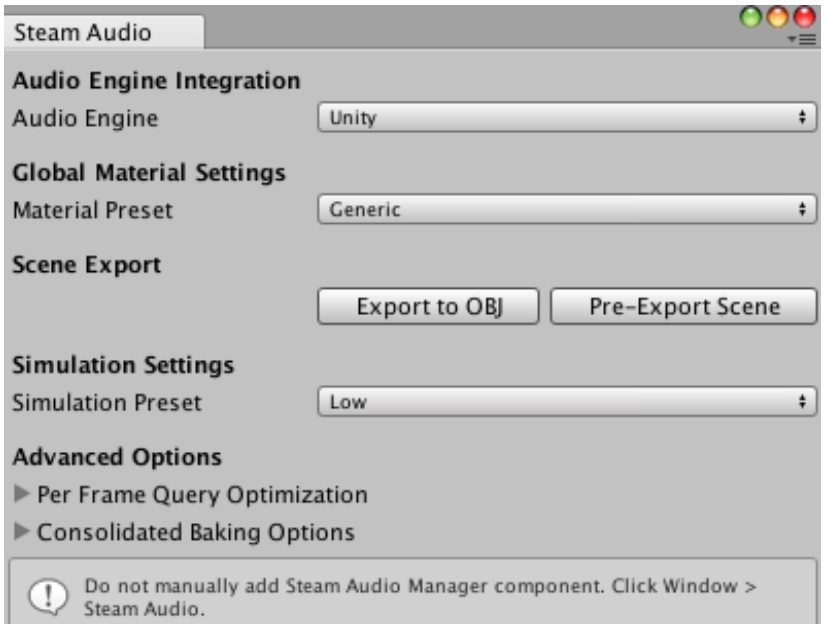
PICTURE 17. ONSP Audio Source plugin in the Inspector window in Unity (screenshot of Unity by Julius Nuora, 2017)

5.1.2 Steam Audio (Valve)

Steam Audio spatializer plugins are created by Steam, a company and platform best known for their online gaming services. An instructional video by the author on how to implement the Steam plugin can be found in Youtube (click here, full link at appendix 2). Similarly to implementing Oculus spatializer, one has to first import the package and choose the spatializer plugin via the Audio Manager. After adding the plugin via Add Component, one may have to click "Window → Steam Audio" for the plugin to work, opening the Steam Audio Manager Settings (on a separate window, the window being added to the Hierarchy) (picture 18.1 and 18.2).

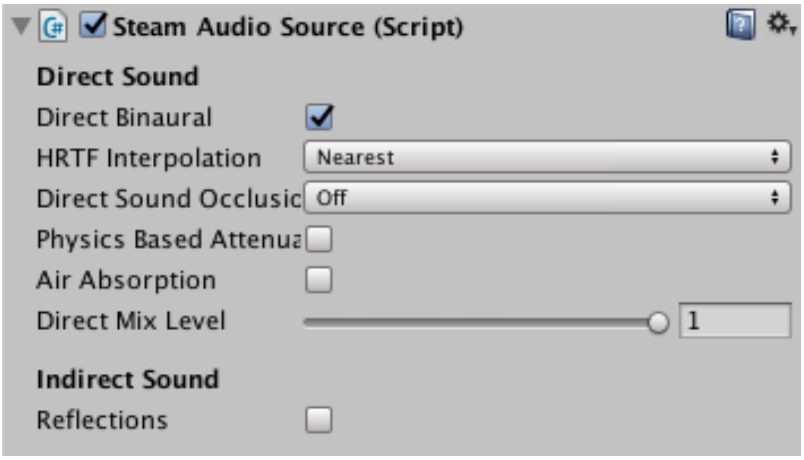
Differing from Oculus spatializer, one has to click "Pre-Export Scene" in the Steam Audio Manager Setting once going to Play Mode, to ensure that whatever changes made in the scene are also available to Steam Audio. This may be irritational at times, making the plugin less intuitive.

PICTURE 18.1. "Window → Steam Audio" (screenshot of Unity by Julius Nuora, 2017)



PICTURE 18.2. Steam Audio Manager (screenshot of Unity by Julius Nuora, 2017)

The Steam Audio Source plugin includes options (picture 19) that are explained in more detail in the manual included with the package. However, the Direct Binaural option is to be ticked in order to enable HRTF-based 3D audio for direct sound, otherwise regular panning is used. Similarly to Oculus Audio SDK, the Steam Audio SDK package also includes other plugins for spatialization (including room geometry, acoustics etc.).



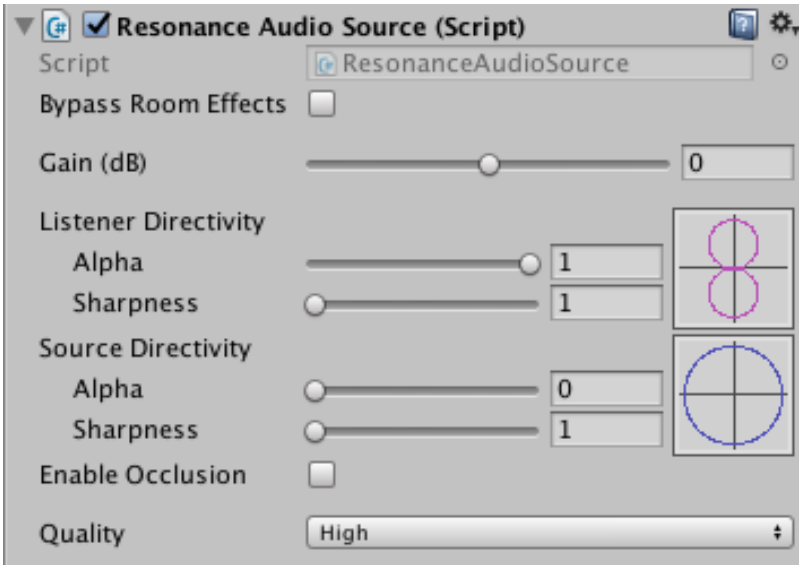
PICTURE 19. Steam Audio Source plugin in the Inspector window in Unity (screenshot of Unity by Julius Nuora, 2017)

5.1.3 Resonance Audio (Google)

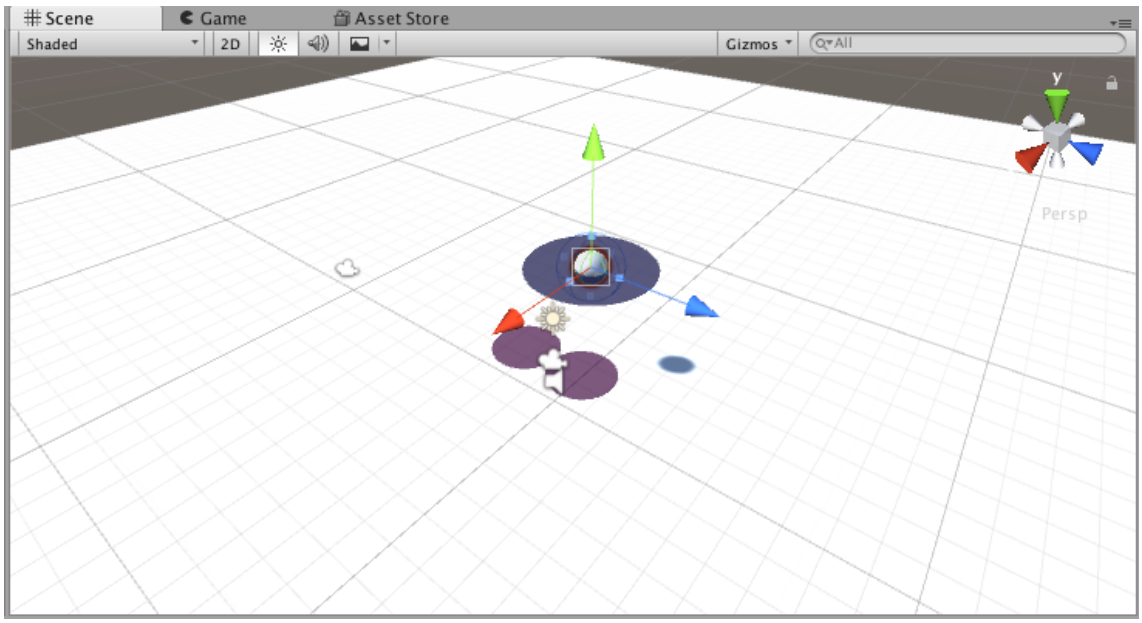
In the VR field, Google has been focusing mostly on mobile applications and low-cost products such as the Google cardboard VR-glasses. However, the company has been creating their own audio spatializer package, formerly included in the Google VR SDK, recently (in the late 2017) newly branded and released as Resonance Audio SDK. Resonance is compatible with all of the common platforms, including similar plugins as its competitors.

An instructional video by the author on how to implement the Resonance plugin can be found in Youtube (click here, full link at appendix 2). Implementing Resonance in Unity follows the same guidelines as the others mentioned earlier. The Resonance Audio Source plugin includes some interesting options, such as Listener Directivity and Source Directivity, ranging the directivity from a full circle (omni) to a figure of eight

(as in microphone polar patterns) (picture 20). It also includes a more visual approach in Unity, one being able to see the directivity changes in the Scene window (picture 21).



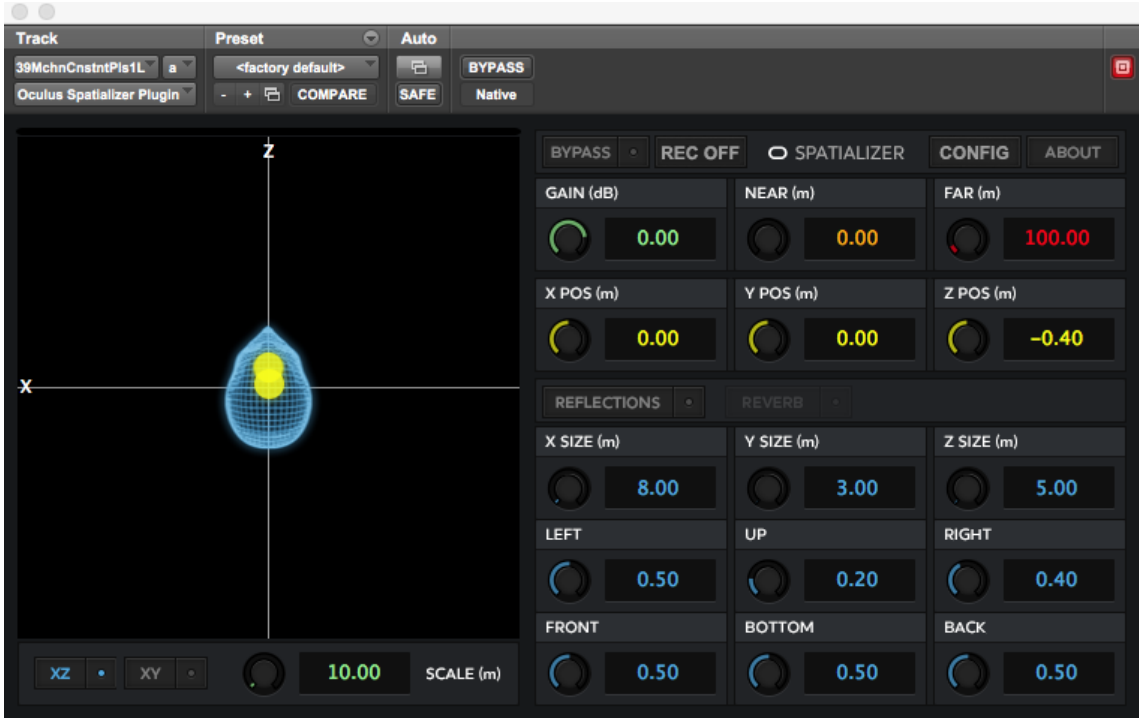
PICTURE 20. Resonance Audio Source plugin in the Inspector window in Unity (screenshot of Unity by Julius Nuora, 2017)



PICTURE 21. Resonance Audio Source plugin in the Scene window in Unity, showing the directivities of the Listener below (figure-8) and the Source above (omni) (screenshot of Unity by Julius Nuora, 2017)

5.1.4 DAW Spatializer Plugins

In addition to plugins used in game engines or middleware softwares, there are spatializer plugins that can be used in a DAW (picture 22). In game audio development these plugins are helpful as testing equipment when creating the audio material before implementing the audio in a game engine or middleware software. One is able to hear and assess how well the sound will work with HRTF processing already when creating the sound. This saves time (not having to switch between softwares) and enables troubleshooting (for glitches or dead spots) in an early phase. These plugins should be bypassed or made inactive when exporting the audio, since the spatializer plugins at further phase (in the game engine or middleware software) will handle the actual spatialization and HRTF processing.



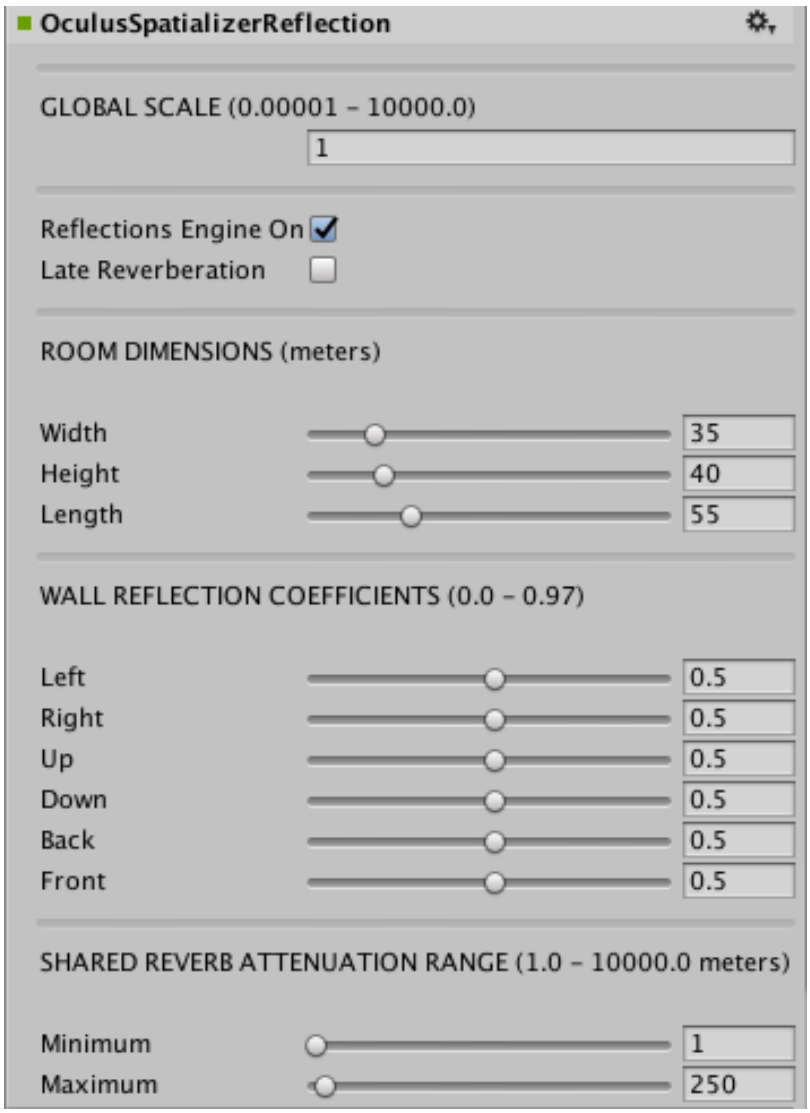
PICTURE 22. Oculus Spatializer plugin in Pro Tools (in default setting) (screenshot of Pro Tools by Julius Nuora, 2017)

5.2. Comparison

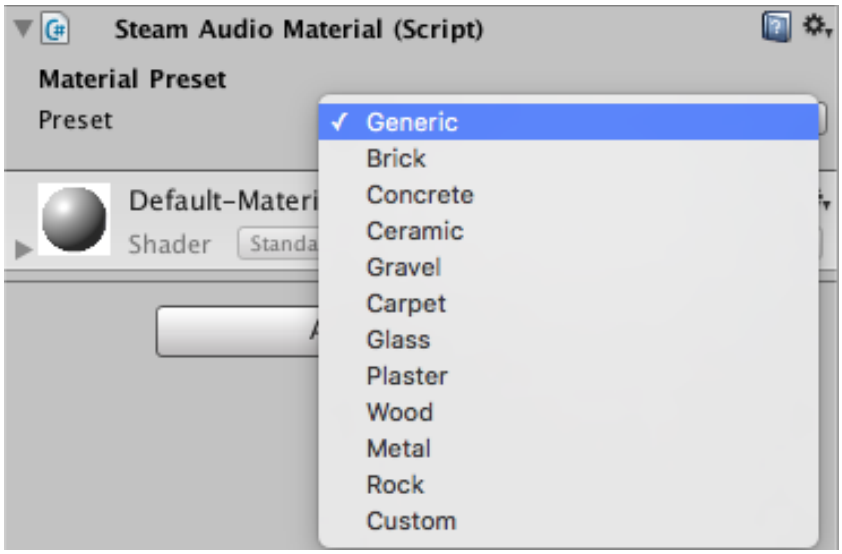
It can be quite difficult to compare the sound quality of different spatializer plugins. One could do measurements about the quality of different plugins' HRTF processing, but in practice it is more useful and interesting to compare their properties, compatibility with host softwares and their additional content. The previous chapter presented the plugins that are added to the AudioSource, however the spatialization packages include several other plugins as well, especially for spatialization of different rooms (acoustics, reverbs and reflections).

All of these packages include an acoustics (named either as reflection, reverb, room etc.) plugin in their package. One is able to adjust the sound absorption properties of the room, either by choosing between different wall materials or more technically by adjusting the absorption coefficients. Absorption coeffiecient is an acoustics term, referring to how different materials absorb (and reflect) sound. Value 1 stands for full 100% sound absorption and thus reflecting nothing (for example a sound going straight out of an open window), whereas value 0 stands for a surface that fully reflects a sound without any absorption (Gracey & Associates).

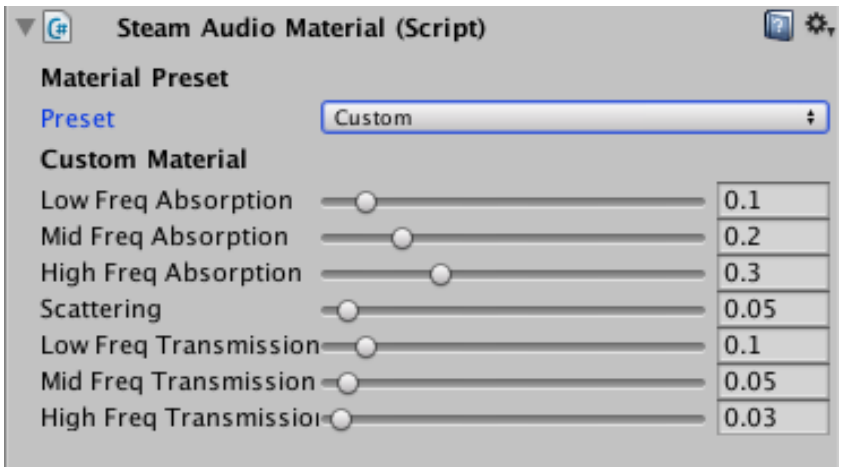
These manufacturers' plugins offer varying solutions, for example Oculus is more precise about wall absorption options, one being able to adjust the coefficient values separately on each wall. Additionally, the Oculus plugin offers options about room dimensions and reverb attenuation range (picture 23). Steam Audio (picture 24.1) and Resonance (picture 25) on the other hand offer options between different wall materials, leaving the coefficient calculations for the plugin. Additionally, the Steam Audio plugin offers a custom material option, enabling one to define the absorption between different frequencies (picture 24.2). Which one of these plugins is the “best” depends both on the project's requirements and the sound designer's preferences.



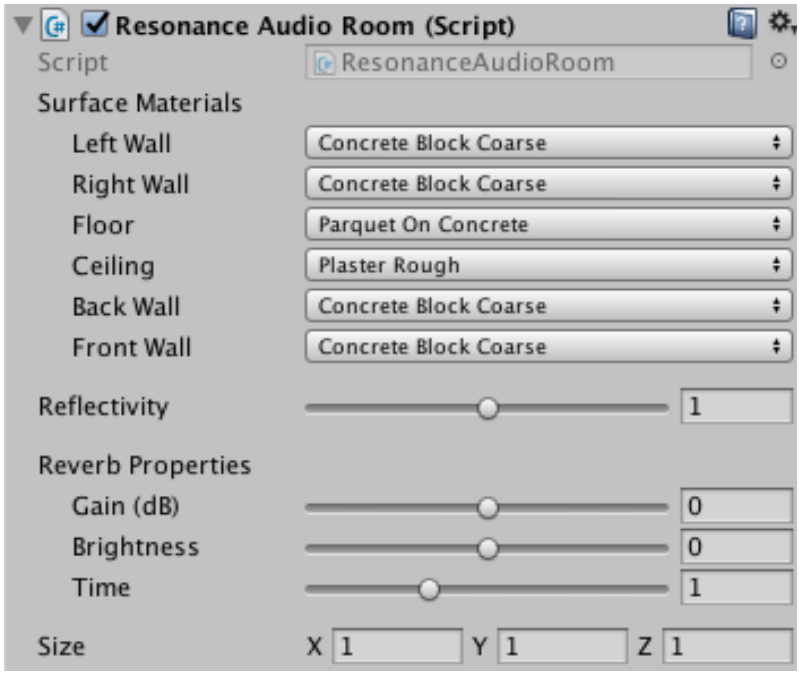
PICTURE 23. Oculus room plugin in Unity (screenshot of Unity by Julius Nuora, 2017)



PICTURE 24.1. Steam Audio room plugin in Unity (screenshot of Unity by Julius Nuora, 2017)



PICTURE 24.2. Steam Audio room plugin in Unity with custom settings (screenshot of Unity by Julius Nuora, 2017)



PICTURE 25. Resonance Audio room plugin in Unity (screenshot of Unity by Julius Nuora, 2017)

In addition to comparing the properties of these plugins, one can view them and their manufacturers in a larger scale, as in what is their focus. All of these plugins are compatible with Unity, but they have differences in their compatibility with other host softwares, for some quite obvious reasons. Oculus and Steam Audio focus on game development, whereas Google is an all-around company, developing VR applications for all all platforms. Google's Resonance is compatible with almost all platforms,

whereas Oculus and Steam Audio are compatible especially with platforms involved in game development (table 2).

TABLE 2. Compatibility of spatializer plugins on different platforms (based on the information provided on the manufacturers' websites in January 2018).

	OCULUS	STEAM	RESONANCE
Unity	x	x	x
Unreal	-	x	x
Wwise	x	”coming soon”	x
FMOD	x	x	x
Mobile	-	-	x (Android Studio)
iOS	-	-	x
Web	-	-	x
Native	x	-	-
DAW Plugin	x (VST and AAX)	-	x (VST)

6 CONCLUSIONS AND DISCUSSION

Creating sound design for VR games differs quite a lot from sound design for non-VR games. Non-VR games usually include a screen with speakers or headphones, whereas VR games require the use of HMDs and headphones, providing isolation that enables a truly immersive experience. Audio may either break the immersion or contribute to it. With 3D audio and spatialization it is possible to create soundscapes that not only support the immersion but also provide cues and information to the player.

However, every game is different and not all VR games require spatialized audio. If a game is not meant to be that immersive and localizing sound sources is irrelevant, a two-dimensional and head-locked sound design may be entirely adequate. There are no strict rules on how to create sound design for VR games. The methods depend on the style of the game, possible technical limitations and personal preferences, often with a trial-and-error workflow.

Yet the way we as humans hear and are able to localize sound sources is consistent. Spatializer plugins used in sound design for VR games are based on the concept of our ears receiving a sound and our brains interpreting the information and determining the location of the sound source. Knowing about the theory behind 3D audio and localization helps with understanding how spatializer plugins function and how to use them. When looking for information about 3D audio, one should be aware that scientific facts are usually mixed in the content with new ideas and experiments. One should try to filter and differentiate the facts from new theories.

The development of VR games and 3D sound design has been very fast. During the writing of this thesis there has already been new “breakthroughs” in for example implementing ambisonics and spatializing near-field sounds. We are also living in a transitional period where VR games are not yet that common for the consumers. The gear and computer power required to run a proper VR game have still been quite expensive for them to become mainstream. Although this is going to progress gradually, it may also affect and limit VR sound design for some time.

This thesis reviewed a few spatializer plugins and their implementation in Unity game engine. All of the plugins functioned well, including mostly same functions with minor differences. Although the comparison did not provide much concrete results, the process may be useful for somebody who is studying the subject and learning about different options between spatializer plugins. For a more purposeful overview it would require an actual game project to see how the plugins work in action. Although demo scenes such as the one presented in this thesis are useful for testing and experimenting, real-life game projects with their possibilities and constraints provide a more realistic overview.

Similarly to DAWs and audio plugins, the field of 3D audio packages and spatializers will probably include various manufacturers having different products and solutions as well. Currently most of the plugins are still free of charge, mostly because the companies want customers to become used to their products. As the market grows and the products become more refined, we may see a change in this. Nevertheless, VR games are here to stay.

REFERENCES

Angus, J. & Howard, D. 2006. Acoustics and Psychoacoustics. 5th edition. Elsevier.

Bible, T. 2016. Binaural Audio for Narrative VR. Blog post. Released on 31.5.2016. Read on 12.11.2017. https://www.oculus.com/story-studio/blog/binaural-audio-for-narrative-vr/

Blue Ripple Sound. 2015. Binaural Audio and Head Tracking. Article. Read on 12.11.2017. http://www.blueripplesound.com/head-tracking

Brown, L. 2017. A Brief History of Virtual Reality. Article. Released on 12.4.2017. Read on 12.11.2017. https://filmora.wondershare.com/virtual-reality/history-of-vr.html

Collins, K. 2008. Game Sound: An introduction to the history, theory, and practice of video game music and sound design. Cambridge: The MIT Press.

Collins, K. 2013. Playing With Sound: A theory of interacting with sound and music in video games. Cambridge: The MIT Press.

Gracey & Associates. Acoustic Glossary.
http://www.acoustic-glossary.co.uk/sound-absorption.htm

Google. 2017. Google VR SDK for Unity. Updated on 9.8.2017. Read on 12.11.2017.
https://developers.google.com/vr/unity/guide

Hook, B. Audio Engineer, Oculus. 2014. Oculus Connect: Introduction to Audio in VR. Video lecture. Uploaded on 20.9.2014. Viewed on 12.11.2017.
https://www.youtube.com/watch?v=kBBuuvEP5Z4

Kall, A. 2011. A Brief History of Binaural Audio. Article. Released on 16.8.2011. Read on 12.11.2017. http://www.kallbinauralaudio.com/a-brief-history-of-binaural-audio/

Kuivalainen, A., Tiitinen, P. & Tikka, Pia. 2017. Pää edellä virtuaalitodellisuuteen. VR tekijän näkökulmasta. Avek-lehti 1/2017. 8-15.
http://www.kopiosto.fi/avek/avek_lehti/fi_FI/avek_lehti/

Modern Mixing. 2013. Plugin Presets: Should You Use Them While Mixing?. Article. Read on 2.3.2018. https://modernmixing.com/blog/2013/07/04/plugin-presets-should-you-use-them-while-mixing/

Oculus. 2016. Oculus Ambisonics Starter Pack 1.0. Document. Read on 12.11.2017. https://developer.oculus.com/downloads/package/oculus-ambisonics-starter-pack/

Oculus. 2017. Introduction to Virtual Reality Audio. Article/document. Read on 12.11.2017.
https://developer.oculus.com/documentation/audiosdk/latest/concepts/book-audio-intro/

Packer, R. & Jordan, K. 2001. Multimedia: From Wagner to virtual reality. 1st edition. New York: W.W. Norton & Company, Inc.

Potisk, T. 2015. Head-Related Transfer Function. University of Ljubljana. Faculty of Mathematics and Physics. Article. Released on 9.1.2015. Read on 12.11.2017. http://mafija.fmf.uni-lj.si/seminar/files/2014_2015/Seminar_Ia_Head-Related__Transfer_Function_Tilen_Potisk.pdf

Salen, K. & Zimmerman, E. 2004. Rules of Play: Game design fundamentals. 1st edition. Cambridge: The MIT Press.

Schmidt, B. Creative Director, EarGames. 2014. 3D Audio: Back to the Future. Video lecture. Viewed on 12.11.2017. http://www.gdcvault.com/play/1020451/3D-Audio-Back-to-the

Schmidt, B. 2017. 9 Things You Should Know When Creating Sound for Virtual Reality. Article. Read on 12.11.2017. https://www.gamesoundcon.com/single-post/2017/09/03/9-Things-You-Should-Know-When-Creating-Sound-for-Virtual-Reality

Silventoinen, A. Sound designer. 2017. Interview on 13.9.2017. Interviewer Nuora, J. Transcribed. Helsinki.

Smurdon, T. Audio Content Lead, Oculus. 2015. Oculus Connect 2: 3D Audio: Designing Sounds for VR. Video lecture. Uploaded on 26.10.2017. Viewed on 12.11.2017. https://youtu.be/IAwFN9sFcso

Smurdon, T. Audio Content Lead, Oculus & Stirling, P. Audio Manager and Lead Architect, Oculus. 2016. Elevate Your VR Experience with Ambisonic Audio Design. Video lecture. Uploaded on 12.10.2016. Viewed on 12.11.2017. https://www.youtube.com/watch?v=a43AvvHwW70

Smurdon, T. & Stirling, P. 2017. Oculus Connect 4 | Breakthroughs in Spatial Audio Technologies. Video lecture. Uploaded on 26.10.2017. Viewed on 12.11.2017. https://www.youtube.com/watch?v=l7mhXRB9PA4

Recording Revolution. 2011. Is it OK to Use Plugin Presets?. Article. Read on 2.3.2018. https://www.recordingrevolution.com/is-it-ok-to-use-plugin-presets/

Virtual Reality Society. 2017. History of Virtual Reality. Article. Read on 12.11.2017. https://www.vrs.org.uk/virtual-reality/history.html

APPENDICES

Appendix 1. Interview with Sound Designer Can Uzer

1. Could you tell a little bit about your background on sound design for VR games, how did you get started, which game titles you have worked on etc.?

I've been working on various games since I've moved to Finland 4 years ago, including lots of mobile titles, some PC and desktop games. I've also have a background in film sound, so when VR started to become hot, I naturally found myself in a bunch of projects. Some VR short movies, game demos and even research projects. I tutored in a VR filmmaking workshop, went to events organized in Helsinki and met with the community. VR is a very fresh field and everybody is experimenting. If you have some technical experience to top your sound design skills with, people appreciate it as a valuable asset. Eventually I started working with 3rd Eye Studios in their VR space adventure game, Downward Spiral and that's been extremely interesting project to work on.

2. What would you consider are the main differences between creating sound for non-VR games and VR games (especially concerning the audio material, implementation and workflow)?

First of all there are some technical complexities you have to consider, such as spatialization of sounds, room models and using mono sources for spot sounds. And it need quite a bit of testing to find good results. The main reason is that in VR, you are isolated as a user and the authenticity of the soundscape becomes very prominent, making every little detail intuitively audible to even untrained ears. Workflow-wise it's more or less the same with regular game audio, but there are practical details such as working with a headset and its and always working with headphones. Another big difference is the fact that you can use your hands freely, so that the Foley work has to be more fluid than, say an FPS. Also, it makes everyone touch everything, thus forcing you to create sounds for all the small props.

3. What do you think are the biggest challenges in designing sound for VR games?

Making it sound authentic and fluid, without over or under localizing elements.

4. Could you tell a little bit about your workflow? Do you have preferences between different game engines, audio middleware and spatializer plugins?

I like how Unity is compatible and flexible, but when it comes to engines, it's not up to me to decide, so I try to adapt as best as I can to the engine that is chosen for a given project. For middleware, I would go with Wwise as my first choice, but native middlewares have their advantage of being sculpted for the specifics of the project. On the DAW side, I love REAPER for its stability, customizability and community. I've been enjoying Oculus' spatialization so far.

I usually first play the game and explore the project, define the required sounds. I then create a pallet of sounds for each category of sound (ambience, weapons, foley, etc.). This involves recording material and defining useful libraries. Then I create the assets, implement them, test it and iterate for one category (I usually start with ambience), and then move on to the next category. Once the first pass is over for each category, it's time to get feedback, evaluate and iterate further.

5. What is your approach on designing and implementing ambience tracks for VR environment? Do you prefer mono, stereo or ambisonic tracks? Do you feel that "traditional" stereo tracks break the immersion of VR (considering head tracking and ambience being "inside the head" non-diegetically)?

Using ambisonic recordings (or surround recordings converted to ambisonics) would be ideal but that is not always possible to technical limitations. In that case, I use stereo pairs of ambience exported as dual mono, place them in the world equidistantly, end then blend a bit of 2D in the spatialization to reduce increase the spread and diffuse the sound. I might also make the sounds origin follow the listener to make the ambience more ubiquitous.

6. Similarly, what is your approach on music for VR environment? Is a stereo mix adequate, or does VR require for example a quad mix (with four tracks located in four fixed directions)?

This depends on the project a lot. Best thing is to test and see what works the best. There are no rules such as 2D won't work ever. It depends on what role the music play in the narrative.

7. How do you think that sound design for VR games is going to develop in the future? Do you have some hopes regarding the field, for example finding solutions to some technical issues?

I reckon (and hope) that the frameworks will be more standardized and tools will be more comprehensive and compatible, but it will take some time and iteration.

8. What advice or tips would you give to a sound designer working on his/her first VR game?

Opting out of this one :)

Appendix 2. Links to the Instructional Videos

Implementing Audio in Unity: Creating a Demo Scene:
https://youtu.be/NPiDCFvmRZE

Implementing Audio in Unity: Oculus:
https://youtu.be/MHlHdjmDFuY

Implementing Audio in Unity: Steam Audio (Valve):
https://youtu.be/kLopPji0uDs

Implementing Audio in Unity: Resonance (Google):
https://youtu.be/v5a3xwWVQBc