Tampere University of Applied Sciences

One-man Audio Team In An Indie Game Company

How to do everything by yourself

Zsombor Bencsik

BACHELOR'S THESIS
April 2020

Media and Arts Degree Programme
Music Production

ABSTRACT

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

Zsombor Bencsik
One-man Audio Team In An Indie Game Company

Bachelor's thesis 57 pages, appendices 1 page
April 2020

The video game industry has been growing rapidly over the past decade. The need for quality video games is bigger than ever. This thesis is written for those who want to get into game audio production as a sound designer, composer, or a technical audio designer. It provides beginner, intermediate and advanced techniques in music, sound design, voice-over, and implementation.

Often a AAA game has dozens of professionals working on the title's audio side. However, employing so many trained experts requires a huge budget that a small indie game company does not have. A typical indie studio can only afford 1-2 employees at maximum. Thus there is a desire for audio professionals with a wide variety of skillsets who can handle processes ranging from recording to coding.

Furthermore, the thesis dissects one of the top-selling VR shooter game Zero Caliber's audio design and provides helpful tips and tricks for VR audio production covering sound recording, handling voice-over, using an audio middleware software, and learning basic coding in Unreal Engine. Additionally, it includes advice from the industry veteran Bence Pajor, former audio director of the Battlefield franchise.

Key words: game audio, music production, sound design, implementation, middleware

CONTENTS

ABBREVIATIONS AND TERMS

AAA	AAA is an informal classification used for video games produced and distributed by a mid-sized or major publisher
DAW	digital audio workstation
low-end	low frequencies
npc	non-player character
fps	first-person shooter

1 INTRODUCTION

Music and video games have always been a huge part of my life from a very early age. But it wasn’t until 2015 when I became acquainted with game audio in college and immediately a whole new world opened up to me. I could take my knowledge of music and sound design and apply it to video games. Emotions can be greatly influenced by sound and music in video games so it was fascinating to me how adding a soundtrack to a game can change the atmosphere so much. Since then my obsession for video games grew even bigger and ever since I have been striving to create immersive and emotional music and sounds that serve video games.

As technology advanced, more storage, memory, and processing power led to more possibilities and a focus on audio improvements in games became very important. Less repetition, more variety, randomness, interactivity, and adaptable sound behaviors become reachable goals. (Marks 2017, 425.) However, even with increased technology, new resource challenges, and the demand for greater interactivity, immersion, and impact, we still need every trick in the book to make today's experiences sing (Damian 2017). ”There are unlimited possibilities a person with audio expertise can do for a game soundscape with the right tools in hand.” (Marks 2017, 425).

The video-game industry is about to change dramatically over the next decade. Cloud gaming, digital distribution, fresh revenue models, new players, and greater regulation are likely to be some of the biggest trends. (Mohamed 2019.) The game industry market value has seen a vast increase in the current years.

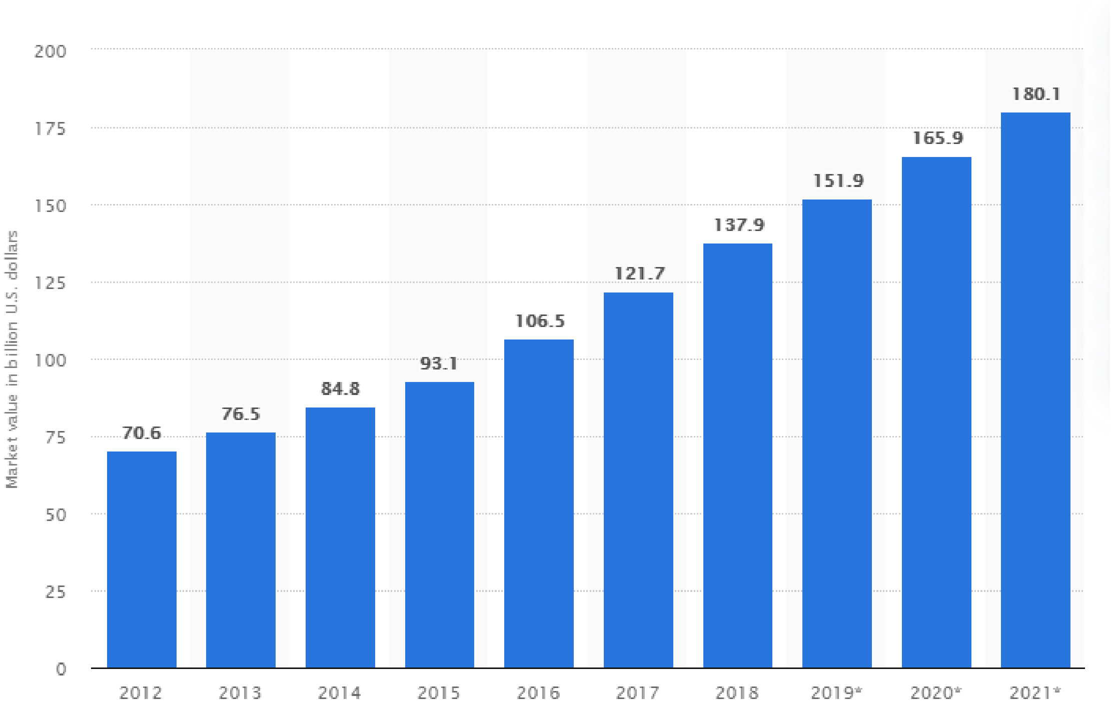


FIGURE 1. Video Game Industry Market Value (Gough 2019)

The projections show even more increase in the upcoming years, that means more game companies and more job openings in the industry. The number of Audio Designer job postings has been steadily increasing on Soundlister.com in the past few years.

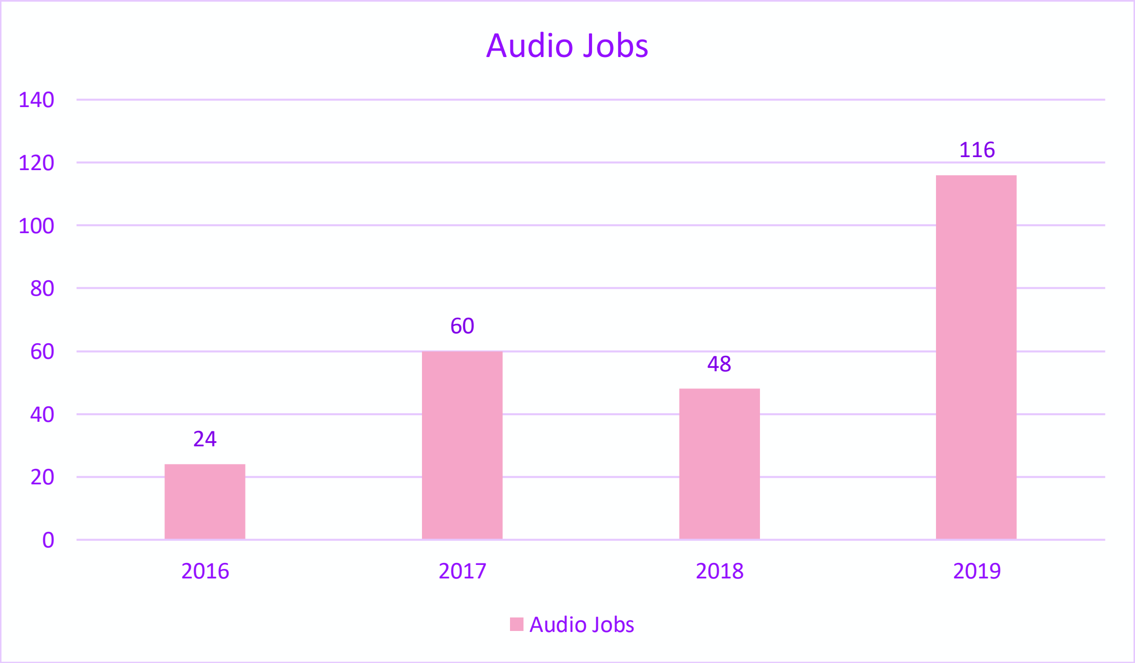


FIGURE 2. Job listings on Soundlister.com (Bencsik 2020)

I have started working in an indie game company two years ago and since then I’ve been responsible for all audio in the studio’s upcoming games, so there has been a lot to learn. Being a one-man audio team was very challenging at first since there was so much to learn about sound design, music, voice-over, implementation, and coding. Learning resources on sound design and music are widely available online, however, finding any information on implementation and audio coding was very difficult. Thus this thesis intends to aid those who are just starting as I did years ago. It will introduce the different branches and roles of game audio and discuss VR audio production techniques extensively with examples from one of the most selling VR shooter game Zero Caliber. The thesis connects audio design with its implementation to serve a bigger picture and a deeper understanding of the audio design decisions.

2 VIDEO GAME AUDIO

The games industry is a creative, exciting, and challenging place for musicians and sound designers. A soundtrack can completely immerse a player in another universe or reality. The creative freedom to alter emotions, environments, and atmospheres is limited only by the technical capabilities of the designer's imagination. (Marks 2017, 21.) However, creative freedom comes with a price. Since game audio is a non-linear media, it can be technically demanding sometimes. For instance, in linear media, a car passby sound is easy and straight forward to implement while in games, the sound designer has to be mindful of the car's speed, direction, distance, perspective, location, and many more as all these attributes affect the sound the player hears.

"Game audio is an art that goes unseen, literally. It is the ephemeral glue that binds the relationship of an experience to one of our dominant senses by rendering the action on-screen with analogous sound." (Kastbauer 2017). This quote resonates with me as in my opinion sound should be a silent support that enhances what the viewer sees on the screen. Sound has the unique ability to communicate to the player directly in a way that is impossible to represent visually (Kastbauer 2017, Dude History?, paragraph 32).

2.1 Sound Designer

Sound design is an integral part of any game, just as important as visuals, gameplay, and music. Great sound effects create an impact that rounds out the entire gaming experience, and without them, that experience would be empty. They are designed to completely absorb the player into a virtual world, making it immersive, entertaining, and satisfying all at the same time. (Marks 2017, 277.)

Sound designers are responsible for recording, editing, mixing, and mastering sound effects. Much like sound effect editors in the film industry, the main job of a sound designer is to search through commercial audio libraries to find the right sounds for a project, or, when the right sounds cannot be found or licensed, record sounds in the studio or in the field and process them through a daw to create the desired effect or tone. However, sound designers do so much more than these. They are true jacks-of-all-trades, and in addition to placing mics, recording and editing dialogue, recording and editing field sounds, sound designers are likely to mix sound and implement it in the game engine. (Maguire.)

A sound designer on a AAA game project is usually more specialized than one who works on an indie game; for example, sound designers who specialize in the dialogue are called dialogue editors, and those who specialize in building audio into the game world are audio implementers (Maguire). This is an extremely creative and rewarding job, that needs strong listening and artistic skills.

2.2 Technical Sound Designer

A technical sound designer’s interest is to bring a sound recorded in the field or designed in a DAW and getting it to play back in a game appropriately (Andersen 2014). As mentioned before, having good sounds and music is not enough for an exceptional and immersive audio experience. Great attention needs to be given to implementing the sounds and music. According to one of the biggest game studio Ubisoft, technical sound designers need to be able to design and implement audio assets within the audio engine, game engine and pipeline, script complex audio playback sequences incorporating sound design, music, and voice-overs. Furthermore, they manage the middleware project organization, structures, gameplay parameter setups, and optimizations. (Ubisoft.) Audio middlewares such as FMOD and Wwise are often used for implementing sounds. Technical sound designers are also responsible for creating a clear audio mix where important sounds can be heard during important moments during gameplay.

2.3 Composer

Video games are getting more complex and immersive, so are the soundtracks that accompany them. These scores can be just as dramatic and sophisticated as any movie soundtracks. Since music affects the players at a purely emotional level, a composer’s job is to capitalize on this and make the players feel how he wants them to feel. The role of a composer is to enhance, suggest, and paint emotions during gameplay. (Marks 2017, 235.)

“Music is like a language we knew before our birth, its intimacy both surprising and obvious in turn.” (Chance 2017, 3). The composer should know how to set the tone, support the story as well as how to orchestrate, mix, and master. Music should be written flexibly to respond seamlessly to events and player actions at any given moment. For instance, quickly change from “wandering around” music to” battling for your life,” while not sounding obvious or abrupt. Game composers also need to be able to write music that can be arranged and pieced together to not sound too repetitive even when the game is played for hundreds and thousands of hours (Schmidth 2018).

2.4 Dialogue Editor

Dialogue production for a AAA video game can often be an intense and often incredibly challenging process. Getting an actor in the booth and reading a script is in itself a monumental achievement that requires solid tools, pipelines, and communication. (Bridgett 2009.) The main job of a dialogue editor is editing, cutting, and renaming recordings. This could mean cleaning up noise, pops, clicks, and artifacts; adjusting levels; compression; equalizing voices; devising process patches to modulate voices (e.g. the voice is a robot’s voice, or it is coming from a TV), and reviewing the visuals to make sure that every minuscule mouth movement syncs with the sound. (Jones 2019.) Dialogue is crucial in video games and can ruin a game entirely when it is not properly performed and recorded.

2.5 Audio Director

Creating a video game is an enormous act of collaboration, requiring professionals from different departments to work on the same schedule and create designs that are aesthetically complementary. For instance, the animator who is developing an animation for a AAA game might never speak to the sound designer who's designing the sound for that particular animation and yet, their work has to progress at the same rate and reflect the same design principles. (Brosius 2016.)

At the head of any video game audio team is an audio director: an experienced senior audio designer who oversees the whole design and implementation process and quality of audio work on a title. Audio directors typically start out as audio designers or technical designers. After working for several years in that position, one gets promoted as a senior and can lead projects. Having experience with multiple branches of game development like game design, sound design, audio implementation is necessary to become a successful audio director. (Brosius 2016.)

3 GAME AUDIO In A VR GAME

In 2018 I got a job in a very small indie game company as an audio designer. At the time I was composing a lot of music and was making some sound effects as a freelancer. My knowledge was very limited on the technical side of game development. I was put on a first-person VR shooter called Zero Caliber to handle everything regarding audio. At first, the task was incredibly overwhelming as I had so little experience in game development. I have worked on the game for 1,5 years and learned an astounding amount about sound design, implementation, coding, music, dialogue, and teamwork.

One thing I cannot stress out enough is to listen to references. It was a huge help in being able to listen and analyze the audio of AAA titles. For Zero Caliber my main reference point was the Battlefield game franchise. Despite not being a VR game the immersion they created in their games is jaw-dropping. Guns are loud and punchy, ambient sounds are three-dimensional and rich, foley sounds are sort of dusty and real. I have had the chance to ask some questions from the former Audio Director of the franchise Bence Pajor (appendix 1). Bence is truly an industry veteran with more than 15 years of experience in creating immersive game audio. He was asked two questions.

1. What is the most important thing when creating audio for realistic shooter games?
2. All of your guns have an incredible low end and punch; are these coming from real-life recordings or are this generated synthetically?

For the first question, he emphasized the importance of first-person sounds. Pajor (2020) said that all the other sounds should support the fps perspective. The line “sounds should leave a mark in the world” (Pajor 2020) is the single most valuable advice I was ever given regarding game audio. I’ve started to re-think how sounds should work and think about how they can affect each other. In real life, if there is an explosion or a gunshot, the environment reacts immediately to the sheer loudness of those events. For instance, birds fly away, people stop speaking, shockwaves might happen, windows break, etc.. It can sell the immersion in VR especially.

Virtual Reality is a relatively new technology and it is still taking its baby steps. I'm certain that that it will define a new era of communication and entertainment. Working with VR opens up a huge amount of possibilities at the cost of being very challenging audio-wise. Imagine a simple metal barrel and the number of interactions it can have; in a traditional pc game the barrel could produce sounds like being shot, punched, exploded, and maybe rain is falling on it, so not a lot of variation sound-wise. In VR however, it can be touched with a huge variety of items with multiple velocities, it can be thrown around, swung, scraped, and can have an endless amount of interactions. Hundreds of sounds just for one simple decoration item. Not having these sounds implemented for items like the barrel will break immersion as audio feedback has to be provided when visuals are present. It is also very performance heavy to calculate the math and play the sounds accordingly.

Getting audio wrong in non-VR games is mildly annoying, but get it wrong in VR and the player will want to kill you. By that I mean badly implemented audio in VR can be so off-putting, it can seriously impede people's reception of their virtual reality, to the point that it may put some people off completely, and this is an enormous problem. Michael Abrash the chief scientist of Oculus said that 3D sound in VR is 'not an addition, it's a multiplier' (Abrash 2018). VR is about immersion and presence, but without a certain level of audio competence, there is no presence. Moreover, because it is a multiplier, there is an extremely fine line between what we would call presence, the illusion that you're actually there, and annoyance. (Taylor 2016.)

In linear media, we see through a small window so it is really important to have sounds that draw us into the world. However, in VR the world surrounds us and you have to chance to change your perception any time by just turning our head, even the slightest movement changes what you see and hear in the world. (Schütze, 2018.)

3.1 Sound Design Techniques and practices

Designing the audio for Zero Caliber was very a very long and difficult journey. The foundation of fps shooter games is the weapon sounds. Guns needed to be even louder, more aggressive, and punchy than in linear media since the gun feels very close to the player. Ambient sounds had to be rich, positional, and immersive. Creating a cohesive sound-scape with hundreds of audio libraries recorded by different people was also an obstacle that had to be tackled.

3.1.1 Gun Sound Design

I believe that everyone should visit a shooting range at least once in their lifetime to grasp the idea of how powerful and loud guns are. Guns sound nothing like the ones heard in movies and video games. In real life, they are a lot less bassy and not at all like in war movies and TV shows depicting combat. A man with extensive experience with firearms described his first impressions firing an M-79 Grenade Launcher.

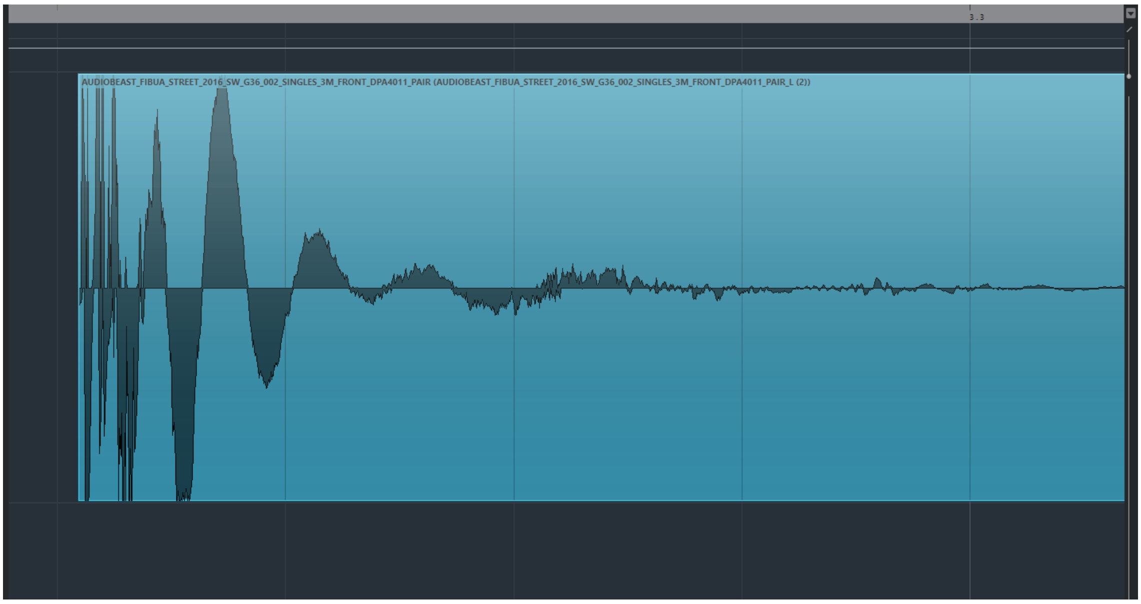
> The sound it makes is “plink.” There is no other word to describe it. It is the wimpiest sounding weapon I have ever heard. Yet once, I was watching some war movie about Vietnam and the actor on screen fired an M-79. The noise it made in the movie was like five 12-guage shotguns going off simultaneously. That’s when I knew my suspicions about Hollywood weapons sound effects were correct. (Reed 2015.)

However, throughout generations, we’ve been listening to gun sounds with a lot of bass so nowadays it has become normal to convey power through bassy frequencies. Designing punchy, aggressive gun sounds that have weight needs good source material, processing skills, and a lot of experimentation.

The type of gun should be taken into consideration. The ideal way of designing would be to get gun recordings of the same gun, but often it is not possible either for a monetary reason or simply just not having available recordings. The gun model you’re designing the sound for should be observed; how big is it, what material is it made of, what kind of shape does it have? These are all things that should be considered before starting to shape the sound. It is self-explanatory that small guns are not as loud as big guns for example and metallic looking guns should have some metal sounds layered in while plastic guns should sound more plastic-ish.

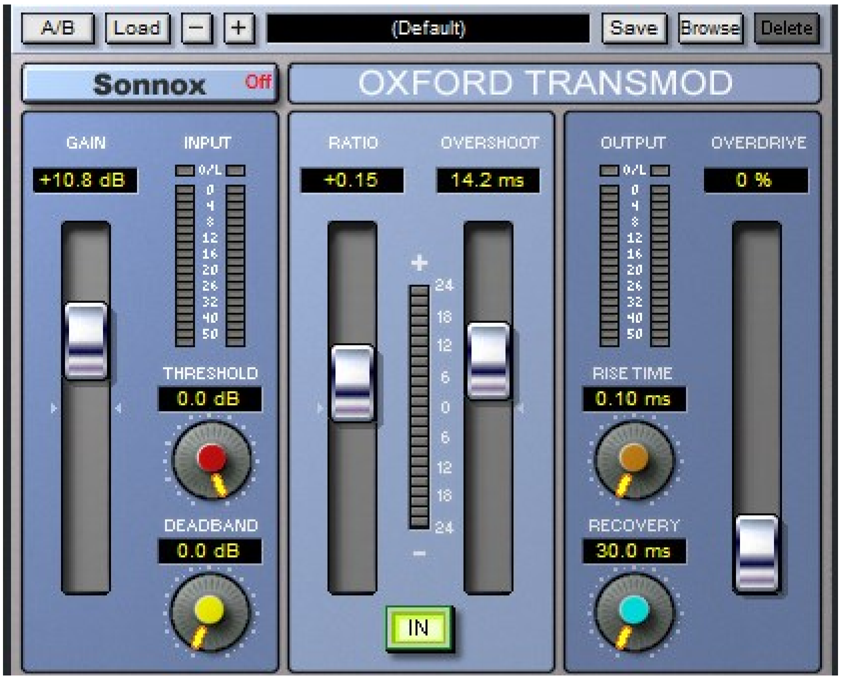
High-quality gun sounds can be found on the internet for relatively cheap, but more often than not they still need heavy processing. Sound design is all about finding good sounds and combining them. Gun sound design is no exception to that principle. Different layers are needed representing the whole frequency spectrum from 20-20k Hz (Smith 1999). Frequency analysis is a huge part of sound design in my opinion. When referencing sounds from other games, one should record the in-game sounds then import it into a daw for frequency analysis.

When layering gun sounds, there are usually 4 parts. A sub/bass/thump, a mechanical, a noise/character, and an environmental layer. The bass layer consists of a very sharp and loud high-frequency transient and a short bass sound that resembles a kick drum.


3.3
AUDIOBEAST_FIBUA_STREET_2016_SW_G36_002_SINGLES_3M_FRONT_DPA4011_PAIR (AUDIOBEAST_FIBUA_STREET_2016_SW_G36_002_SINGLES_3M_FRONT_DPA4011_PAIR_L (2))

PICTURE 2. Bass layer audio waveform. (Photo: Zsombor Bencsik 2020)

A lot of people in the industry use kick drums as this layer, but it sounds too synthetic for me so I just heavily process a gunshot sound. The source sound is a mono gunshot to which I apply my processing chain that consists of 4 plugins. Three of those are adding bass essentially. Rbass from Waves is enhancing around 80 Hz, Lowender is adding just a touch of sub at 40Hz and Maxxbass brings out frequencies around 120Hz. This is very important as most of the time players use low-end budget headphones and speakers that can’t reproduce low frequencies properly, so having that 120Hz boost makes sure that the punch is present even on budget gear. After bass enhancement, I apply a very aggressive transient shaper called TransMod from Sonnox.



(Default)
GAIN +10.8 dB
THRESHOLD 0.0 dB
DEADBAND 0.0 dB
RATIO +0.15
OVERSHOOT 14.2 ms
RISE TIME 0.10 ms
RECOVERY 30.0 ms
OVERDRIVE 0 %

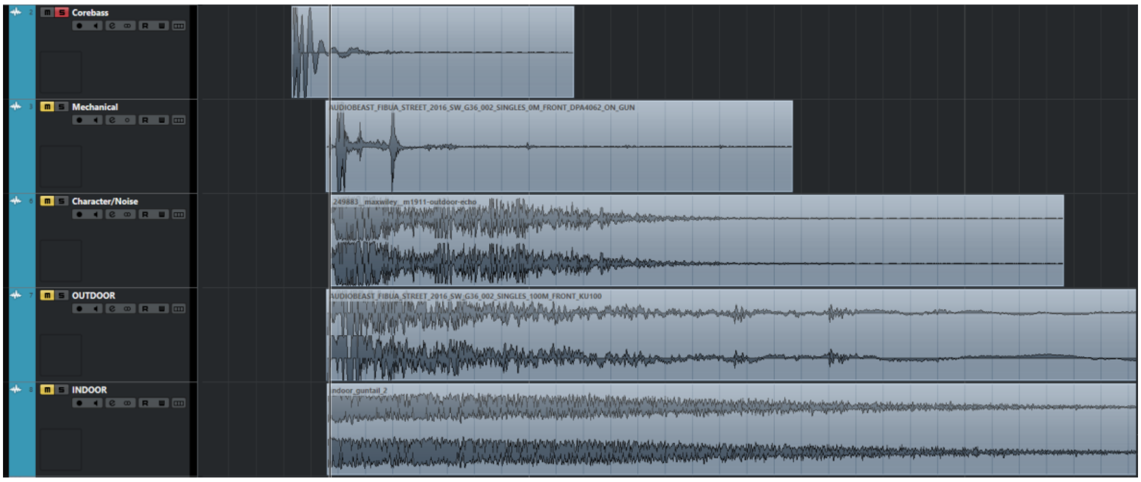
PICTURE 3. Adding sharp transients with Transmod (Photo: Zsombor Bencsik)

This plugin does the bulk of the work. I’m adding 10db input gain and putting the threshold to 0 dB which means it is only going to affect the beginning of the gunshot. It makes the sound clip very hard in a sonically pleasing way.

The second part is the mechanical layer when the bolt kicks back and forth on the gun. That can be anything from a gun reload sound to recording some metal clank sounds at home. This layer represents the high and mid-frequency spectrum so as for processing; I tend to add some high frequencies, compression and some multiband distortion that is driving the mid-frequencies. The noise/character layer is one of the most important. It is really hard to find good source material that can be used as this layer. I found that low-quality stereo recordings of gunshots work great. I usually highpass around 300 Hz with a 6db/oct slope, compress, limit and distort them.

After firing the gun an echo can be heard as the sound is bouncing off surfaces. The environmental layer suggests the location where the gun has been fired. It sounds very different when fired indoors and outdoors. As for processing, I tend to highpass filter from 100 Hz with a 12db/oct slope.

The timeline position of these elements is crucial. If these layers are played at the same time, the power of the shot is lost. Instead, it is recommended to place the bass layer before everything else.


Corebass
Mechanical
Character/Noise
OUTDOOR
INDOOR
AUDIOBEAST_FIBUA_STREET_2016_SW_G36_002_SINGLES_0M_FRONT_DPA4062_ON_GUN
249883__maxwiley__m1911-outdoor-echo
AUDIOBEAST_FIBUA_STREET_2016_SW_G36_002_SINGLES_100M_FRONT_KU100
indoor_guntail_2

PICTURE 4. Layer placements for a gunshot in Cubase 10 (Photo: Zsombor Bencsik)

I have placed the bass layer 44ms before every other layer as you can see in the picture above (picture 4). This way the pop and the bass cuts through even more. It is advisable to use multiple variations of each layer and randomizing the layer timings helps with the authenticity and realness of the sound as every shot of a real gun sounds slightly different than the previous one.

As mentioned before, the Battlefield franchise has the best gun sounds in my opinion. My second question (appendix 1) for Pajor (2020) was about creating punchy and high fidelity gun sounds. He said for guns to have a punchy low end, everything else must contain less low frequencies. This was an immensely valuable answer because I had a look at my ambient sounds and they had some very low frequencies that were taking up the frequency range so I high-pass filtered them immediately.

3.1.2 Environmental sounds

Listen, the snow is falling... In addition to that, my computer is humming, traffic is driving by outside, birds are intermittently chirping, not to mention the clacking of my "silent" keyboard. Life is full of sound. We've all spent time basking in the endless variation and myriad ways in which the world around us conspires to astound and delight with the magic of its soundscape. (Kastbauer, 2017.) We are very aware of how the real world sound and cannot be fooled easily. Real-world environmental sounds are extremely important in VR for this reason. Without them, VR feels lifeless and empty. Most of the traditional games use 2D ambient sounds with spot sounds (e.g bird chirping, river flowing, human speech) that is enough for that medium. However, in VR the player can rotate his head which makes a world of difference as it becomes much more important to have more 3D sounds in the world like rain hitting a barrel, a fly buzzing around, lights flickering, etc.. for the player to feel he's in a real world surrounded by real objects and living beings.

One of the most important tools for environmental sound in VR is ambisonic audio. Ambisonics is a 3D recording and playback method that is based on the representation of the sound field excitation as a decomposition into spherical harmonics. This representation facilitates spatial sound production that is independent of the playback system. The adaptation to a given playback system (loudspeakers or motion-tracked headphones) is achieved by a suitable decoder. (Frank, Matthias & Zotter, Franz & Sontacchi, Alois. 2015.)

Throughout the years I have been building my sound library so I have gathered a good amount of ambient sounds. The processing I usually do with them is only some gain-adjustment and eq. I have found that using a lower noise base sound can work great as a sort of glue that holds the other sounds together. Using your own recordings can also be rewarding. Last winter I was in Lapland and happened to be in a snowstorm that I recorded with my portable recorder, then later implemented into one of Zero Caliber's snowy levels. It turned out great, even though it was only a 2D layer.

Wind sounds are very important in VR as they enhance immersion greatly. Everybody knows the feeling of smooth wind blowing in their faces. It is tricky to find good and isolated wind sound sources that sound in your face, so to emulate that, I have recorded myself blowing gently in the microphone and layering that with other wind sound sources. The technique turned out to be very effective.

Zero Caliber is an apocalyptic VR game that takes place in a world where clans are fighting for drinkable water. The player is in a constant warzone so I had to somehow convey that through audio. I took a bunch of gunshots, cannon shots, mortars, explosions, and lowpass filtered them so they sound distant. The player always hears these in random sequences so even if he’s not in an actual fight he still feels the pressure of war.

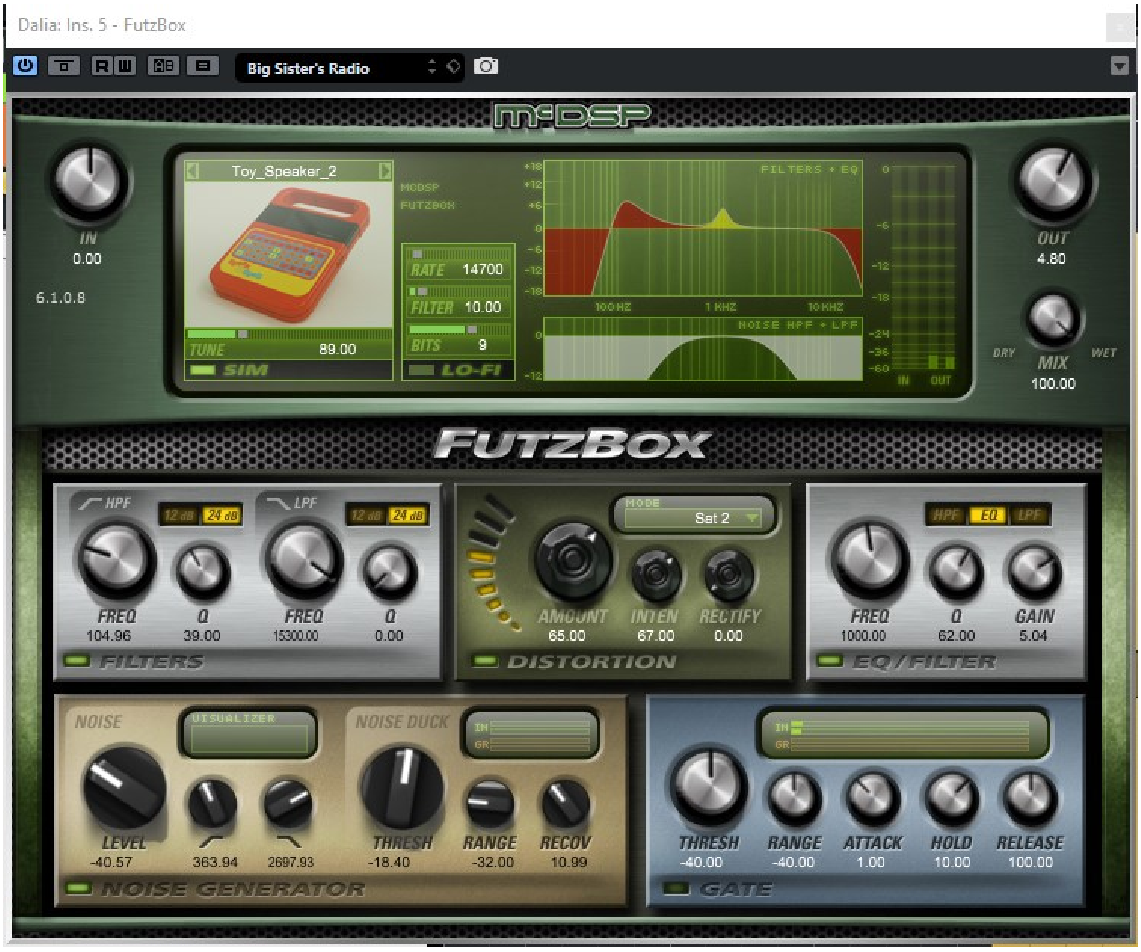
3.1.3 Foley

In the film world, Foley is the art form that adds believable sound garnishment to on-screen character movements. For instance, imagine a man walking down the street on a wet road. Foley artists would perform this scene while viewing the dry footage by using props in a studio, then the recording would be mixed, edited, and perfectly synced to the picture. Doing Foley work for video games is slightly different. If sounds were to be added to the same scene from the previous example, most game sound designers would record the sounds separately and sync it up to the character’s walking animation. (Marks 2017, 279.) Unfortunately, the player does not have animations in VR so doing foley gets even more challenging.

The primary foley sounds in Zero Caliber were footsteps, cloth movement, opening, closing doors, touch sounds, breathing, etc.. Footsteps were quite challenging to make despite having a wide range of sounds from third-party libraries as they sounded too dry and isolated, probably because they were recorded in a studio and not outdoors. So I recorded some of my own footsteps with my portable recorder and layered that in with the library footsteps and suddenly much more noisy and dirty which made it more immersive for VR. For processing only some noise-removal, eq, and Rbass were used to give it a little more weight. I also recorded some indoor footsteps that have some reflections and reverb which is being used when the player is indoors in Zero Caliber.

3.1.4 Voice Over

In Zero Caliber the narration gets a lot of attention, so it was important to get clear and understandable lines. We used Fiverr to get voice actor professionals who did a fantastic job in my opinion. There were diegetic and non-diegetic voice applications. Diegetic sound is any sound that emanates from the story-world of the game (e.g. dialogue, footsteps) The source of diegetic sound doesn't necessarily need to be seen on screen, as long as the audience understands that it is coming from something within the film. Non-diegetic sound, also called commentary or nonliteral sound, is any sound that does not originate from within the game’s world (e.g. 2D music, UI sounds). (Masterclass 2019.) In Zero Caliber the storyline was non-diegetic as it was simulated to be coming from a military radio walkie-talkie. Before processing the recordings to sound like they are coming from the radio, I compressed, eq, noise reduced them since they were coming from different voice actors with different recording equipment and quality. I used FutzBox from Mcdsp and Echoboy to achieve the radio sound.


Dalia: Ins. 5 - FutzBox
Big Sister's Radio
Toy_Speaker_2
TUNE 89.00
RATE 14700
FILTER 10.00
BITS 9
IN 0.00
6.1.0.8
OUT 4.80
MIX 100.00
FILTERS: HPF FREQ 104.96, Q 39.00; LPF FREQ 15300.00, Q 0.00
DISTORTION: MODE Sat 2, AMOUNT 65.00, INTEN 67.00, RECTIFY 0.00
EQ/FILTER: FREQ 1000.00, Q 62.00, GAIN 5.04
NOISE GENERATOR: NOISE LEVEL -40.57, 363.94, 2697.93; NOISE DUCK THRESH -18.40, RANGE -32.00, RECOV 10.99
GATE: THRESH -40.00, RANGE -40.00, ATTACK 1.00, HOLD 10.00, RELEASE 100.00

PICTURE 5. Futzbox radio emulation settings. (Photo: Zsombor Bencsik)

3.1.5 Vehicles

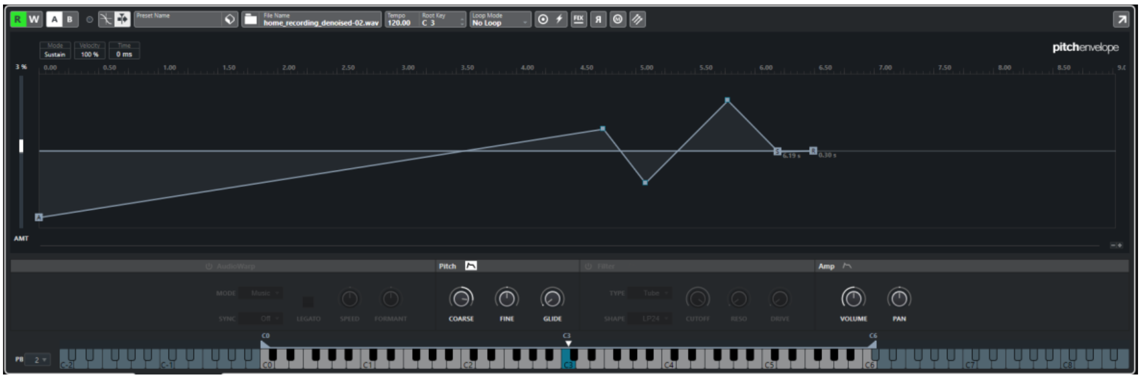
Vehicles were used extensively in Zero Caliber to enhance immersion in VR. Huge helicopters, tanks, military jets were flying by the player, and their sound needed to be extremely loud and in the face. Just as any sound I have discussed, vehicles need layering as well. A vehicle usually has multiple elements like motor, wheels, body. These layers should represent the frequency spectrum from low/sub to high frequencies. Having these separate layers, it allows for detailed control over their characteristics while implementing. For instance, the bass layer can be heard from a greater distance than the high-frequency layers. There are several types of tanks and helicopters in Zero Caliber. In this section, I will go into detail about helicopter sound design.

It is crucial to look at the in-game model of the vehicle first and start designing the sound after as the same principles apply; big vehicle more bass, louder and lower pitch, small vehicle less bass, and higher in pitch. I have found that the rotor of the helicopter is great for low-end representation so I have searched for sounds that were consistent in volume and pitch, which was difficult since most of the helicopter recordings are passbys with ascending and descending volume and pitch. After finding an acceptable recording I used bass enhancement, gating to get a rhythmic quality, and some eq to clean up the muddiness of the sound. For midrange, I tried to find loud and aggressive helicopter engine sounds and for high-end, I used a turbine recording with added distortion. The reason why I don't render them together is distance attenuation which will be discussed in the implementation section.

3.1.6 UI

> Oftentimes, I hear people say that sounds in non-game apps are rude and annoying. There is a rationale behind such an opinion due to the many examples of failures. However, a sound can significantly improve the users' experience and be useful where GUI can not. All that it takes – to put good sounds in the right places. (Zimarev 2018.)

User Interface sounds were kind of magical to me as I had no idea how one makes them. Having searched through multiple forums it seems that making UI sounds is considered as the most challenging aspect of sound design. I have experimented a lot and found a recipe that works most of the time. There are different kinds of UI sounds of course, but for Zero Caliber I wanted to go for something organic since the game is supposed to be realistic and not futuristic. For that reason, I have used a lot of foley sounds and everyday household items that I heavily processed. Here is an example of that. I recorded some door creaking and imported them to Cubase. I converted the sample to a sampler track where I modified the pitch envelope of the file.



File Name home_recording_denoised-02.wav

Tempo 120.00

Root Key C 3

Loop Mode No Loop

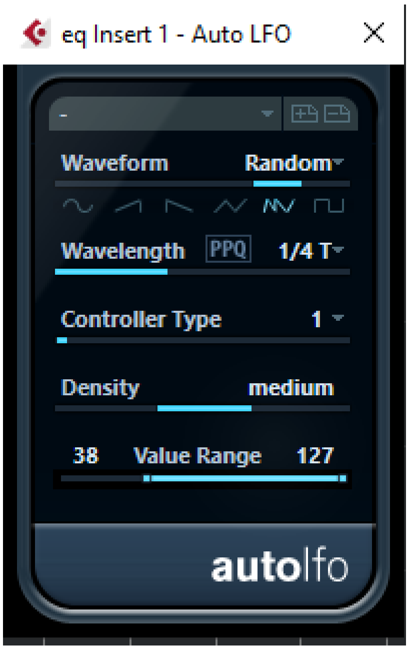
Mode Sustain

Velocity 100 %

Time 0 ms

PICTURE 6. Applying pitch envelope to the door creek sound (Photo: Zsombor Bencsik)

Beside pitch shifting, I used automated ring modulation, phaser, compression and after all these, I had a frequency shifter, a lowpass filter with high resonance, and another ring modulator, but this time I sent an auto LFO that randomly modulates their parameters and this way I got a lot of variation due to its randomness.



eq Insert 1 - Auto LFO

Waveform Random

Wavelength PPQ 1/4 T

Controller Type 1

Density medium

38 Value Range 127

PICTURE 7. Random LFO is modulating plugin parameters. (Photo: Zsombor Bencsik)

3.2 Music in VR

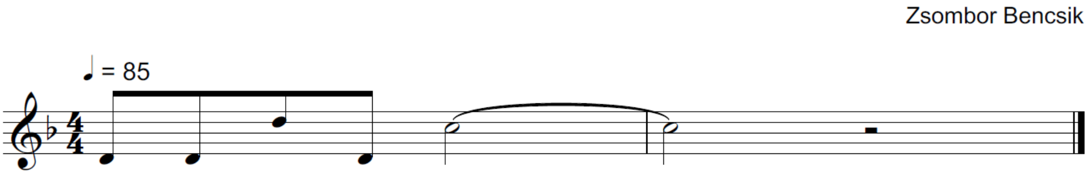
As mentioned before, there are two types of sound sources, diegetic and non-diegetic. Most of the time music falls into the non-diegetic category and being played 2D. There have been several experiments with spatialized music in VR, but due to lack of time, I went with the traditional non-diegetic route so the music in Zero Caliber is being played back in 2D stereo. I have had the luck to work on the game 6 months with extensive testing before starting to write the soundtrack which was incredibly important to get the overall feel and atmosphere of the game. When I started the soundtrack I already had a vision for harmony, pacing, and instrumentation which can be difficult if the composer doesn't know much about the game. The soundtrack album consists of seven tracks not including three game states in each one of them. The implementation of these states will be discussed in the implementation section. In my opinion, the main theme is one of the most important tracks in every game, so in this section, I will go into detail about creating it. Before going into details here is the soundtrack, so there's a clearer understanding of the analysis.
https://www.youtube.com/watch?v=UP4cufNtfgg

Music consists of known components, patterns, and variables that can be identified, learned, and mastered. Imagine this scenario. An auditorium is filled with music students. A large projection screen displays an image from an obscure video game level. None of the students have seen the image before. The lecturer asks, "How would you score this level? What would your music sound like?" After the students collected the ideas, the professor played a piece of music that closely resembled the students' ideas. (Chance 2017, 13.) We all have preconceptions about how different things sound. Years of gaming and watching movies secretly planted how war, military, violence, and action sound. I'm no exception to that, so before starting Zero Caliber's soundtrack I knew I wanted epic drums, guitars, synths, and an orchestra. Capturing some in-game footage and importing it into the daw while composing can be a tremendous amount of support as seeing the gameplay itself while composing helps to decide what works and what doesn't.

As mentioned before my initial instincts about instrumentation were epic drums, guitars, synths, and orchestra so I began with drums. The main rhythm is being performed by layers of surdos, taikos, kick drums, and military footsteps. I was looking for something out of ordinary and found that military footsteps were creating a militaristic tone. The main rhythm is being accompanied by atmospheric synths, low brass, cellos, reversed gun echo sounds, police walkie-talkie conversation, and a helicopter sound. I felt that creating a battlefield atmosphere in the music made everything sound tense and anticipating. This soundscape reminded me of an old fender Rhodes as it has a gritty and dirty sound quality. After some tweaking, it became the best candidate to play the main melody which is recurring throughout the whole album. The track is slowly building up to its climax with raising synth sounds then it reaches it around 1 minute.

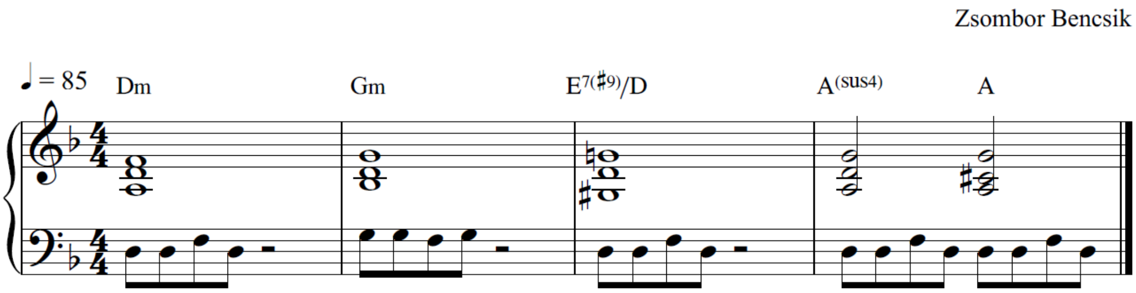
Music is oftentimes referred to as a “language of emotion”, and it is natural for us to categorize music in terms of its emotional associations. Myriad features, such as harmony, timbre, interpretation, and lyrics affect emotion, and the mood of a piece may also change over its duration. (Kim et al. 2010, 1.) Using the wrong instrumentation, rhythm, or harmony would immediately break the immersion of the game. Keeping this in mind, I started to experiment with chord changes and soon figured that the key center is going to be D minor. The minor 7th (D, C) interval in the scale had that mellow and militaristic quality I was looking for.

The harmonic content is quite sparse up until the chorus. The main motif that is played by a guitar is the following.



PICTURE 9. The recurring main motif in Zero Caliber. (Photo: Zsombor Bencsik)

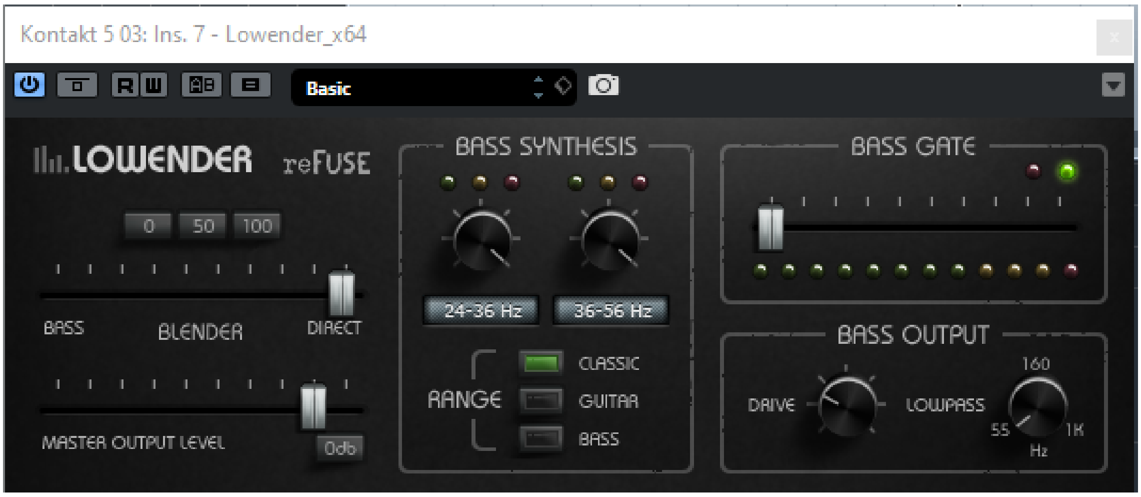
In the chorus, there’s a harmonically busier section. I have experimented a lot with the harmony and settled with using only four chords with a distinct bassline.



PICTURE 10. Chord progression throughout the chorus. (Photo: Zsombor Bencsik)

The chord E7#9/D might seem odd at first for two reasons. First, E7#9 is typically used in jazz, but the way it is orchestrated makes it blend in the soundscape. Secondly, the bass plays a D note that is very unusual in any type of genre, however, the bass is playing a repetitive pattern, so it doesn’t stand out in this case. The chorus is followed by a bridge buildup, where I added more textural elements to enhance the feeling of being in a war before reaching its climax again which is just a repetition of the first chorus.

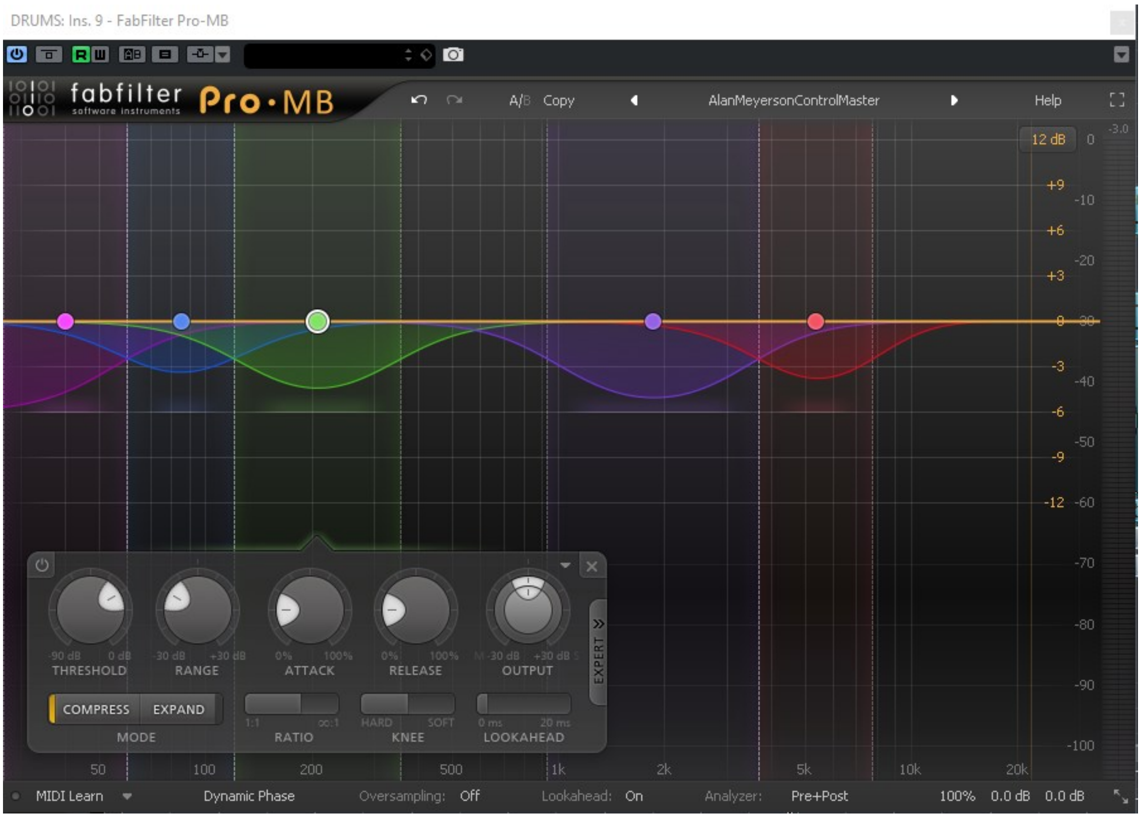
Mixing the album was incredibly challenging as I typically had more than a hundred tracks in each song. At first, the mixes were muddy, boomy, dark, and barely audible in the game itself. I realized that handling a lot of amounts of tracks required grouping, so I grouped my drums, guitars, basses, synths, and effects to separate groups in Cubase. Mixing became immediately easier as I did not have to filter and process dozens of tracks separately as I can do that on the group channel. Getting punchy, full drum sounds were the most challenging aspect of mixing the soundtracks. Just like in sound design, layering is key, drums need low/sub, mid, and high-frequency layers. Once I found the layers I had to blend them, with eq and bus-compression. For low-end, I added some sub using Low-ender.



PICTURE 8. Lowender for sub-bass enhancement. (Photo: Zsombor Bencsik)

Without sub-bass, tracks feel weightless and weak, I only realized this after getting headphones with good bass response.

As for group processing, I applied Fabfilter Pro-MB multiband compression to almost all the groups. With the settings, I have learned from the great score mixing engineer Alan Meyerson.



PICTURE 9. Multiband compression settings (Photo: Zsombor Bencsik)

It is crucial to control low and high frequencies as they tend to be problematic in sample libraries. With a relatively quick attack and quick release, I'm compressing a few dB at 200 Hz and 2000 Hz.

Mastering for video games is slightly different than mastering for other types of media. As discussed before it is very important to have clarity in the game mix. However, having music, narration, loud guns, screams, explosions at the same time is making everything mask each other. In general, I tried to make guns punchy and bassy just like explosions, but where is the room for music then. The way I solved the issue is adding a bit more high/mid frequencies and less bass while mastering, this way the music pops out of the gameplay sounds. "By utilizing opposing frequencies, or ones that are not overloaded by other audio, the game can be experienced as the designer intended." (Marks 2017, 406).

3.3 Audio Implementation with FMOD

What is a successful audio implementation? According to the composer and voice actor Elspeth Eastman, it is the attention to detail.

> Remember that it's a very dynamic process and you have to go over everything 500 times. The difficulty will vary depending on the environment or engine you're working with, but for me, this is a form of audio that requires patience and trial and error. (Eastman 2017, 104.)

Implementation was the most challenging side of game development I had to learn when I was beginning my journey as an audio designer. It required programming logic as well as the ability to think ahead while designing sounds. It took a great amount of time to figure out even the basic implementation techniques, since tutorials, books, and online resources were difficult to come across. This chapter aims to introduce basic and intermediate implementation concepts using FMOD and Unreal Engine. For coding, Blueprints are going to be used in Unreal Engine. The Blueprints Visual Scripting system in Unreal Engine is a complete gameplay scripting system based on the concept of using a node-based interface to create gameplay elements from within Unreal Editor (Unreal Engine Docs).

Fmod events are used to play sounds and music. An event is an instanceable unit of sound content that can be triggered, controlled, and stopped from game code. As a rule, every situation in the game that produces a sound should have a corresponding event. An event contains and is primarily composed of tracks, instruments, and parameters. The parameters trigger the instruments, which route audio content into the tracks. The tracks route into other tracks, or into the event's master track. The output of the event's master track routes into the project mixer. An event can contain an infinite number of audio tracks in theory, but for performance reasons, this is not recommended. In addition, the event's parameters can control and manipulate most properties of the event, of the event's instruments, and effects on the event's tracks; and the event's logic markers can manipulate the event's timeline parameter. (FMOD 2019.) These principles are the pillars of FMOD and the following techniques and practices will use them extensively in the next sections.

3.3.1 Grouping Sounds

Before going into detail about implementation techniques, it is crucial to understand what grouping is and why is it so important. Grouping is used to group similar sounds to achieve a clear mix. There are several ways of doing this, but for Zero Caliber I choose a technique that involves grouping sound by loudness priority. It is important for the proper use of side-chain compression. In Zero Caliber I named these groups ‘duck’ as in game audio, it is very similar to side-chain compression. Ducking, at a base level, is essentially the practice of lowering the volume of all elements of the soundscape except for the track we want to highlight. This allows more headroom in the final game mix, that will provide important information to the player that may otherwise be missed due to the crowdedness of what is going on within the audioscape at that particular moment (Quarles 2009).

Here is the grouping structure in Zero Caliber.

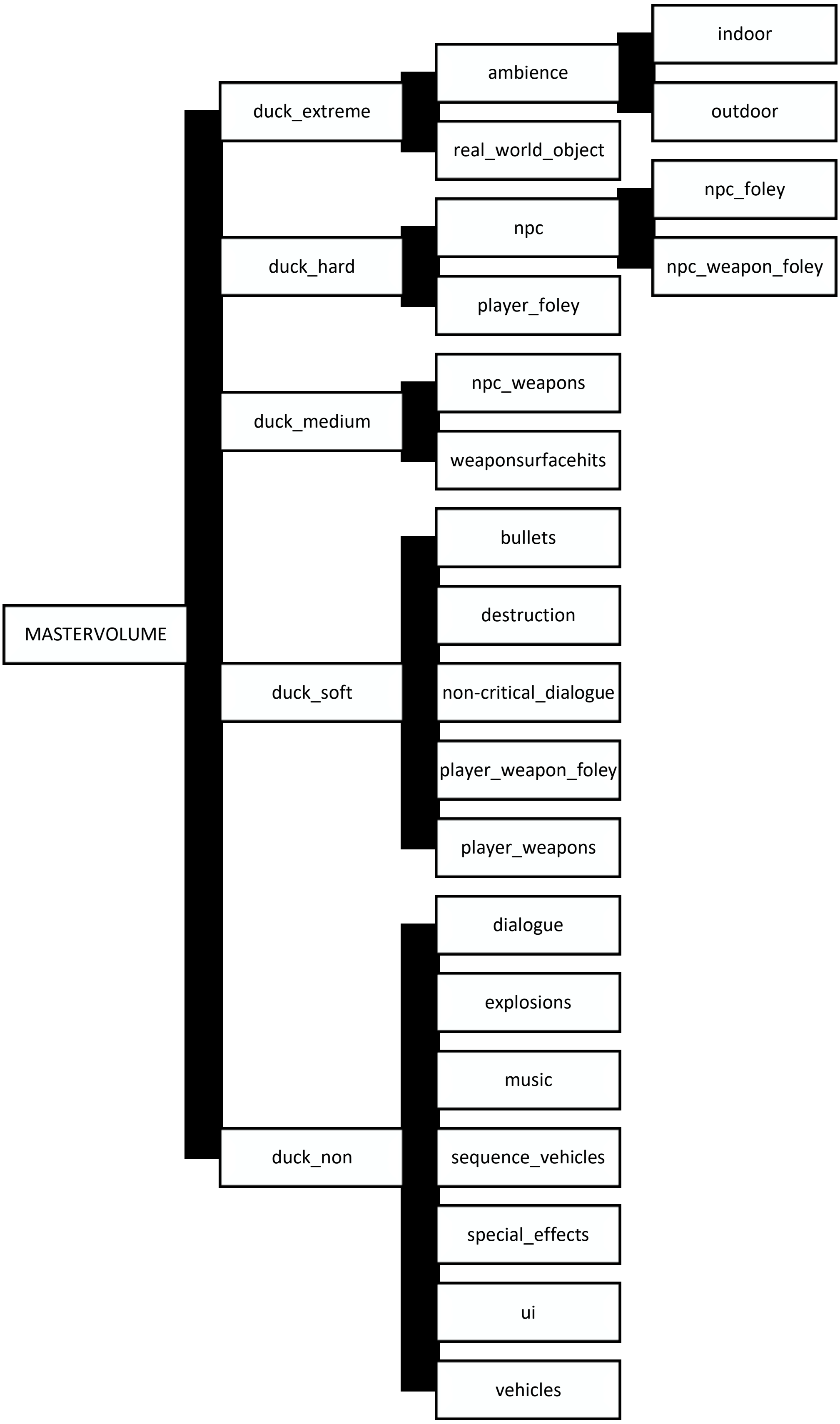
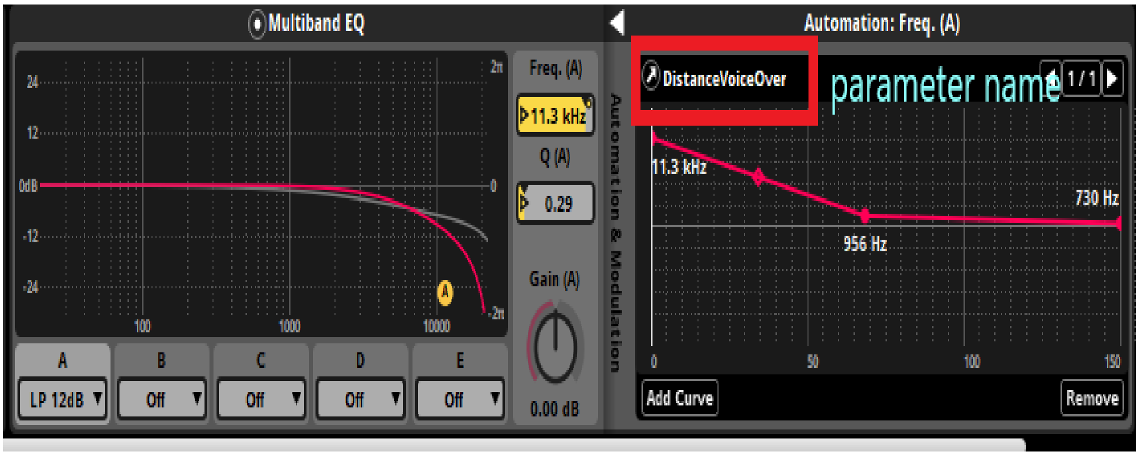


FIGURE 3. Grouping structure in Zero Caliber (Zsombor Bencsik 2020)

As displayed on the figure (Figure 3), everything is routed to the MASTERVOLUME group as it is needed for setting the master volume in-game. Then there are five groups categorized by importance namely duck_extreme, duck_hard, duck_medium, duck_soft, duck_non. After the duck groups, the subgroups are categorized by attributes (ambient, foley, weapons, explosions, music, voice-over, UI, etc..). Groups are valuable for performance reasons as well since they have DSP (Digital Signal Processing) effects like distortion, eq, reverb, and compression on them so it is not necessary to place them on every event separately.

3.3.2 Parameters

For dynamic audio, it is indispensable that the game-engine communicates with the audio middleware. This process happens through the settings of parameters. For instance, switching audio samples based on distance, or altering a car's pitch with an RPM parameter, etc.. Parameters have limitless applications in FMOD, as almost everything can be automated by them. In Zero Caliber they are used extensively, for example in a battle chatter voice sound.

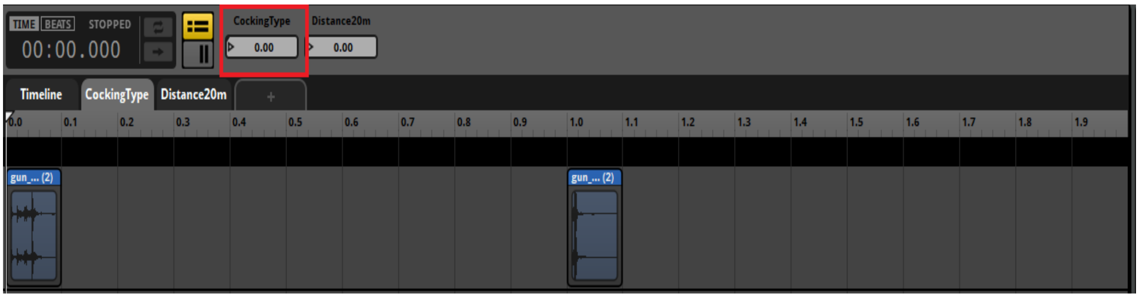


Freq. (A) 11.3 kHz
Q (A) 0.29
Gain (A) 0.00 dB
A LP 12dB
B Off
C Off
D Off
E Off

PICTURE 10. DistantVoiceOver parameter automation in FMOD Studio (Photo: Zsombor Bencsik)

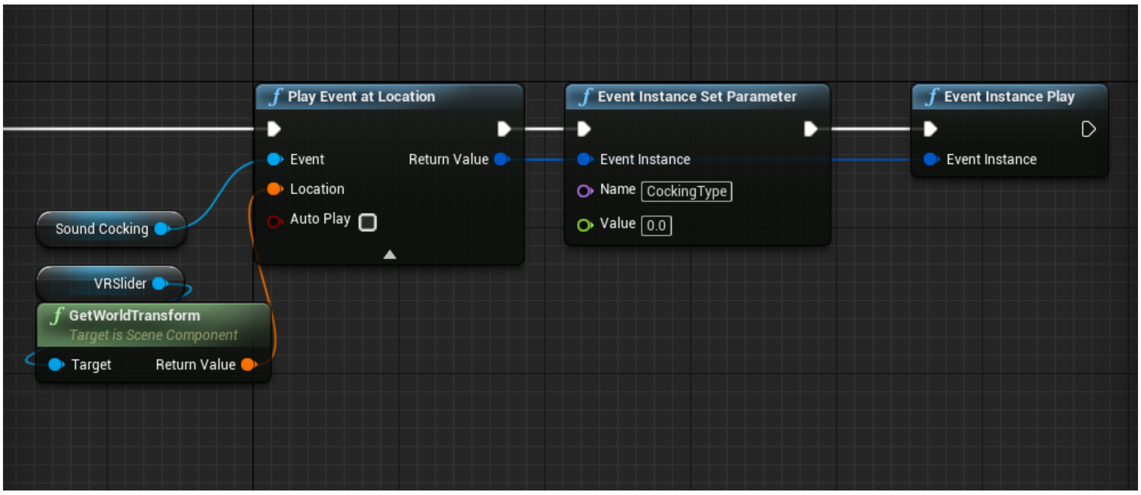
The 12db/oct low-pass filter is being applied on the chatter and the frequency is automated by the parameter called DistanceVoiceOver, so that the further the chatter is, the lower the filter will go. FMOD makes this even easier as all the distance parameters are being called automatically so it is not necessary to set them manually in the game engine.

However, sometimes custom parameters have to be used. In this example, a parameter is needed to decide if the player is pulling the gun bolt back or forth.



PICTURE 11. Parameter in FMOD (Photo: Zsombor Bencsik)

The parameter is called CockingType and it has a value of 0 to 2. If the event gets a 0 as a parameter the event plays back the sound at 0, while if it gets 1 as a value it plays the audio at 1. As this is a custom parameter, it has to be set in the game engine.



Name CockingType
Value 0.0

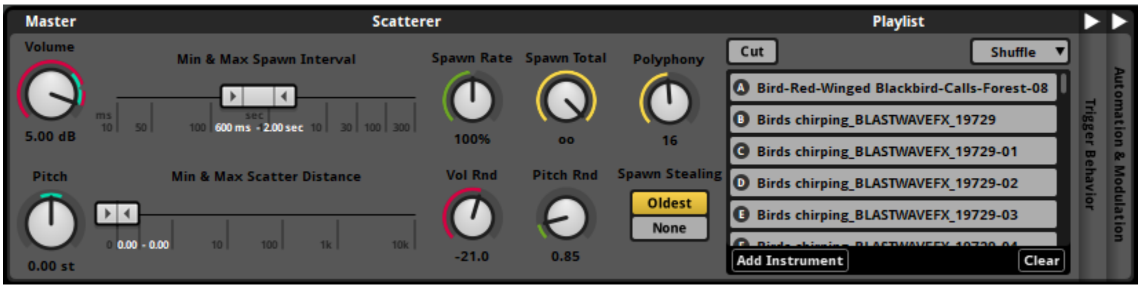
PICTURE 12. Setting parameters in Unreal Engine. (Photo: Zsombor Bencsik)

The event is played at location, then its return value’s parameter, that is an event instance is being set as displayed above (Picture 12). It is fundamental to understand this process before diving into more advanced implementation techniques.

3.3.3 Dynamic Ambiences

Ambiances can be used to great effect in communicating the idea of space. In a video game where you can spend countless hours walking and trying out new guns, one of the keys to extending the experience is the idea of non-repetitive activity. If we can help to offset that from a sound perspective by introducing dynamic ambiance it can help to make the experience more immersive. I'll be presenting various implementation techniques that help to make the game audio less repetitive, boring, and more immersive for the players.

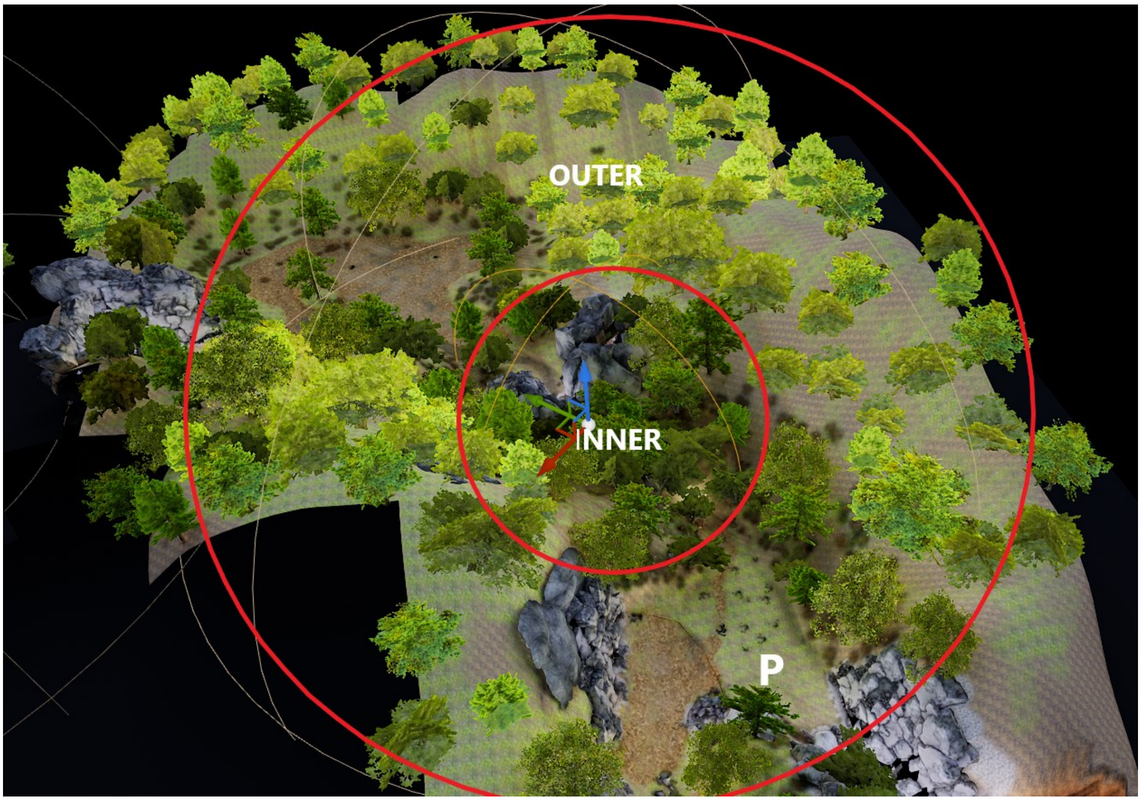
During soundscape creation, there are four implementation techniques we must consider. Spotsounds, 2D sounds, ambisonics, and area sounds. Spotsounds are positioned sounds in the world such as gas pipe leaking, rats squeaking, birds chirping, wind passing through tree branches, etc.. These sounds are crucial for creating an immersive VR space since during head-rotation, players going to hear these sounds positioned. It is very important to use variation with these sounds as repeating the same sound over and over again can be tiring and immersion-breaking. In FMOD it is a good practice to use 'scatter sounds' as they have a lot of randomization options.



PICTURE 13. Scatter instrument in FMOD Studio (Photo: Zsombor Bencsik)

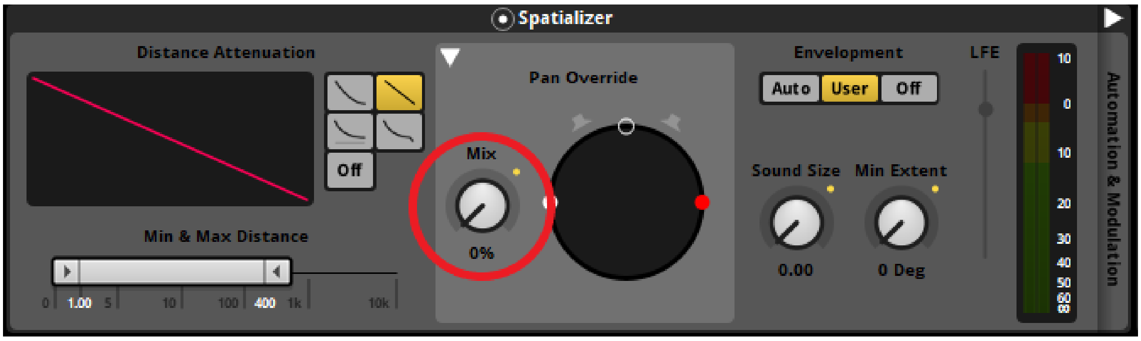
In this example (Picture 13), short bird sounds are being played every 600ms-2sec with random volume and random pitch. Having more than 30 bird samples makes the sound less repetitive and more believable. 2D sounds are layers that are being played back without any positioning. There are no 2D ambient sounds in reality so in VR 2D sounds are must be used carefully and sparsely, as when the player moves his head, the 2D sounds move with the rotation and this effect can be very apparent and audible. In Zero Caliber, ambisonics audio is used extensively on all the levels. For the playback of ambisonic files, Google's Resonance Audio plugin is used with the combination of FMOD. The ambisonic events only need to be placed on the game level to start playing automatically. However, they must be used carefully as they can have a huge file size, which can be a problem for mobile platforms as they have a limited amount of space. A 12-second 1st order ambisonic audio file is 4,823 Mb, while the stereo version of the same file is only 2,413 Mb. This might not seem much, but a 10-minute file can be 500 Mb in size, so it is important to shorten these files.

The last ambient implementation technique is using so-called area sounds. These sounds are 2D and 3D at the same time. In this example, there is a forest ahead of the player. When the player is outside of that forest, the sound is positioned and filtered.



PICTURE 14. Area Sound in a Zero Caliber level in Unreal Engine. (Photo: Zsombor Bencsik)

A forest atmosphere sound is looping continuously in this example. The letter P stands for the player's location. In the inner circle, the sound can be heard 2D without any filtering and positioning. However, from the outer circle, the sound is positioned and filtered depending on the proximity to the inner circle. The event in FMOD has a parameter named PlayerInSoundArea, so when the player entered the outer area the parameter is being set between 0-1 until he reaches the inner circle where the sound switches to 2D, non-positional.



Mix 0%
Sound Size 0.00
Min Extent 0 Deg

PICTURE 15. Mix automation on the spatializer of the event in Fmod (Photo: Zsombor Bencsik)

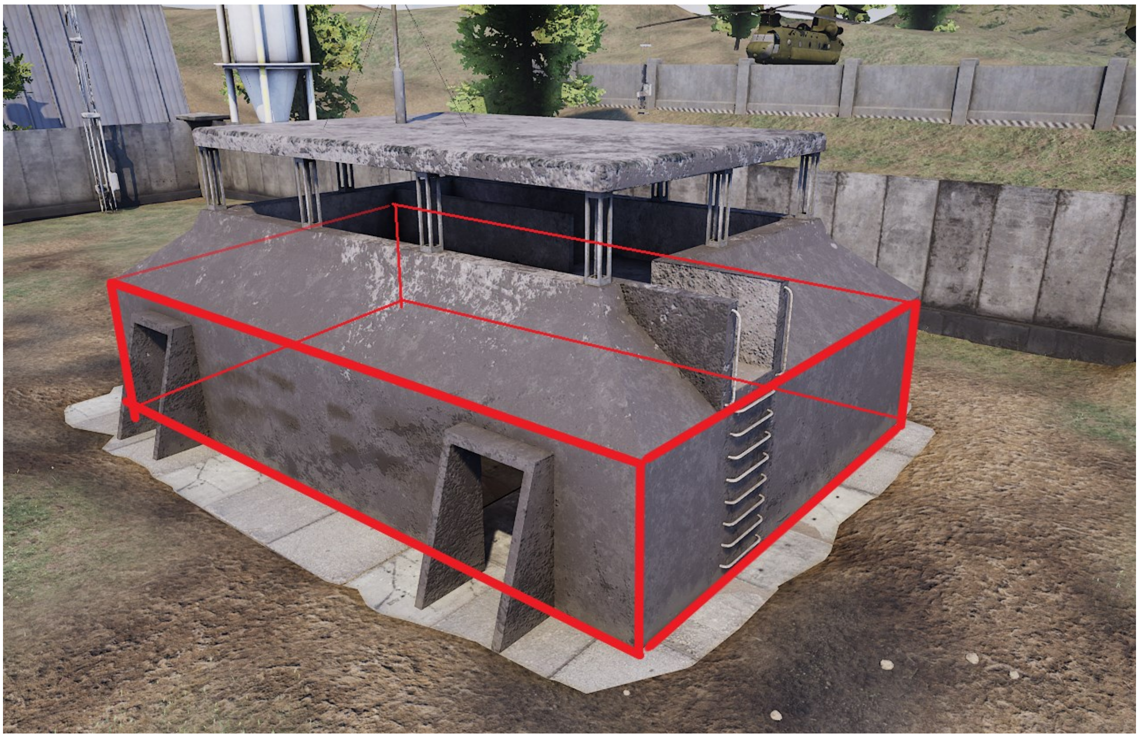
If the PlayerInSoundArea parameter is 0 then the Mix amount is going to be 0 as well, but when the parameter is more than 0 then it starts to mix in the 2D sound so when it reaches 1, the sound becomes stereo 2D. Another important thing is setting the Sound Size and Min Extent with the same logic. These control the stereo width of the sound, where 0 is completely mono and max is fully stereo, so as the player is approaching the inner circle the stereo width of the forest sound is getting wider and wider.

Setting the parameter in Unreal Engine happens with a timer that updates the parameter in various periods based on the player's proximity to the soundarea, so the further the player is, the less frequently this parameter setting happens. The sound is being low-pass filtered as the player is leaving the soundarea simulating real-life acoustics.

3.3.4 Indoor and Outdoor spaces

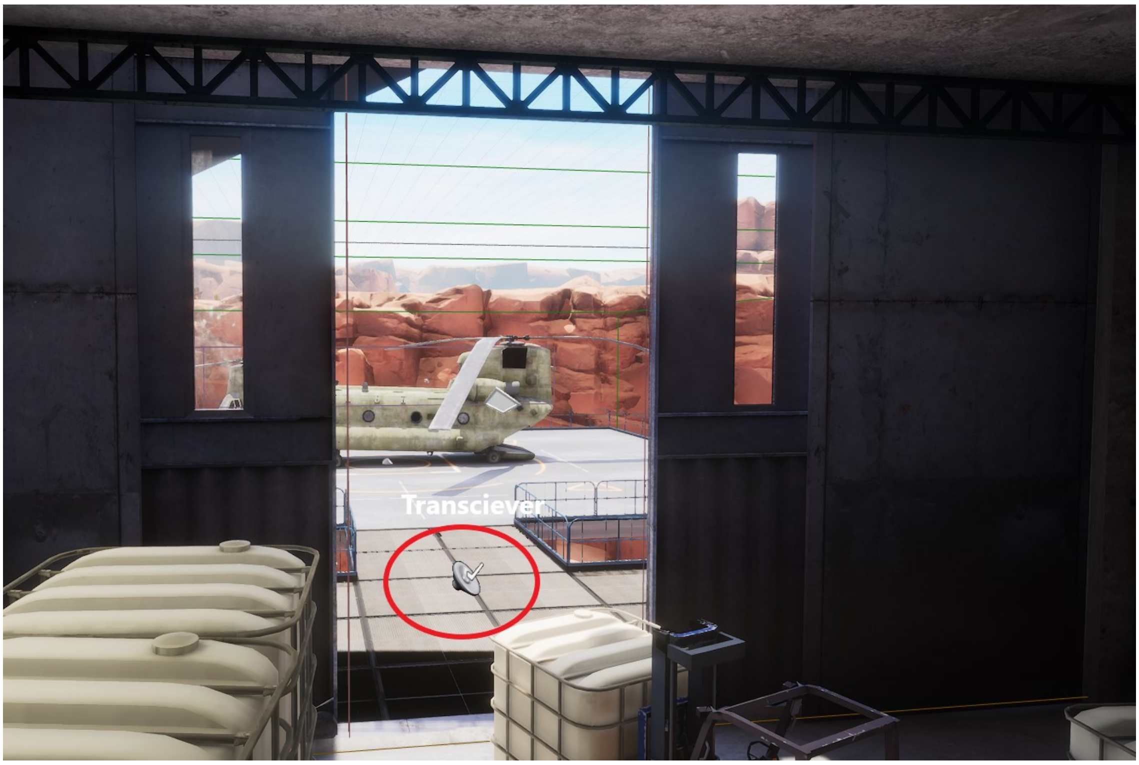
There are several ways of differentiating indoor and outdoor sounds, and there is no right or wrong solution. In Zero Caliber, sample switching based on being indoor/outdoor, algorithmic reverb, and convolution reverb were used. Combining these three had the most believable effect in VR. Swapping sounds and reverbs are quite easy to set and tweak in FMOD, as they can be automated by parameters. For the player, there is a global parameter that is ranging from 0 - 1, 0 for outdoors, and 1 for indoors. The current and set values of a global parameter are shared between all parameters that reference the same global preset parameter. but the game engine implementation was the more difficult aspect (FMOD 2019). So setting this once will affect all the events that implemented this parameter.

Solving the implementation in the game engine was more challenging, as the logic needed to be updated very frequently which can have an impact on the game's performance. Unreal Engine's collider box actors were used to create a re-sizeable box that triggers on the entrance of the player as well as on exit. This way, it was possible to set the PlayerIndoors parameter whenever the player entered and exited the box. Unfortunately, this meant placing this box in every indoor place, which quickly became laborious and time-consuming on levels with lots of complexly structured buildings as those needed more than one box.



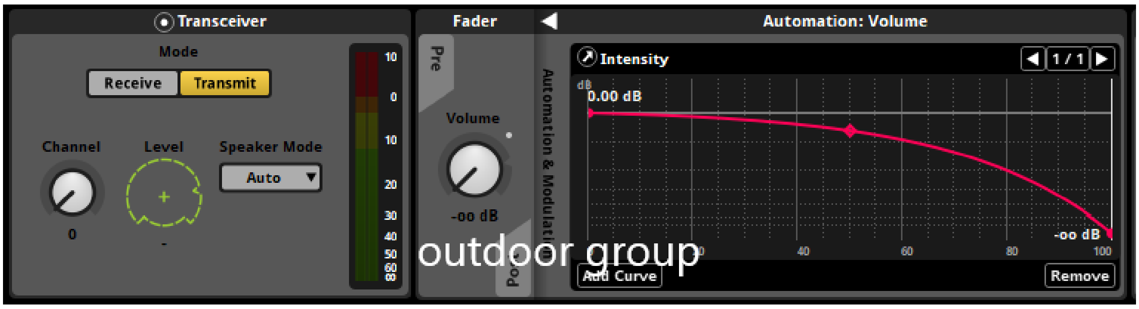
PICTURE 16. Indoor/outdoor box in Unreal Engine 4. (Photo: Zsombor Bencsik)

As displayed in the Grouping Sound section, indoor and outdoor sounds have their respective groups. The reason for that is, when the player enters the box (Picture 16), the outdoor sounds are low-pass filtered and lowered in volume but there are certain cases, however, when the player still should hear the sounds coming from outside, for example, a huge opening. In that case, it is advisable to use FMOD’s transceiver tool.



PICTURE 17. Transceiver at the gateway in Unreal Engine (Photo: Zsombor Bencsik)

The transceiver works just like a send in a DAW, as events can be sent to it and other events can receive it. This is important because in this way it is possible to position the transceiver. Essentially the outdoor sounds are being sent to a transceiver before lowering their volume.



PICTURE 18. The transceiver is sending sounds before lowering the group volume (Photo: Zsombor Bencsik)

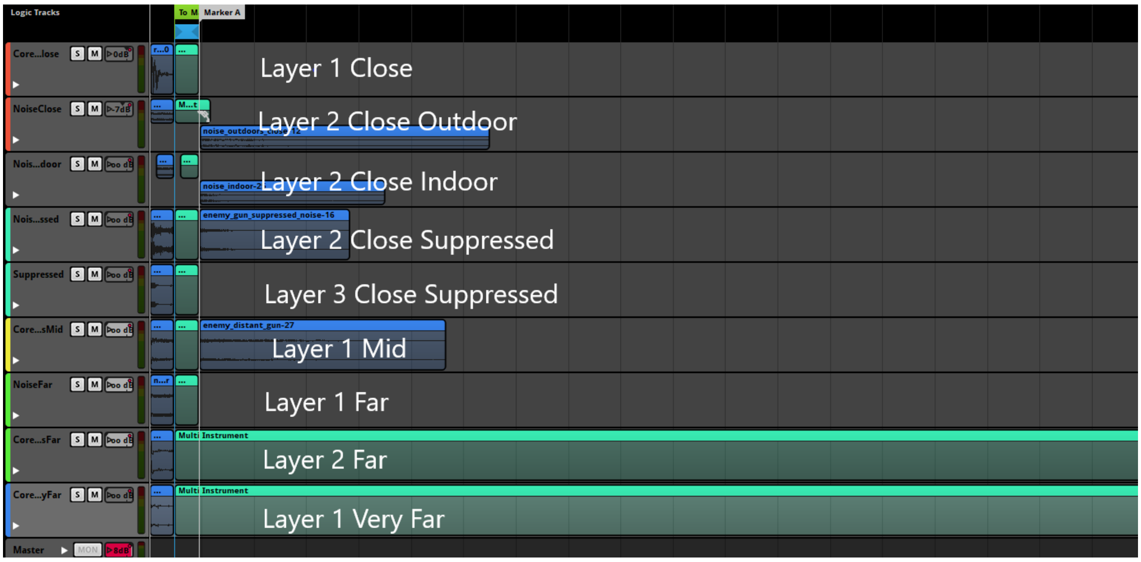
Intensity here is dependent on the global indoor/outdoor parameter. So even if all the outdoor sounds are cut when the player enters the box, the transceiver is still receiving all the sounds.

The box does not only provide information about whether the player is indoors or outdoors, but it also calculates its size, which is used to set the reverb size. The bigger the box is the more reverb and longer tail size is applied. These boxes also have to ability to choose between different types of reverb such as sewers, hangars, and small rooms. Reverbs are used sparsely as they can sound too artificial. They are placed on the separate group channels and being automated by the global indoor/outdoor parameter. Mainly algorithmic reverbs with short pre-delay were used where the reverb-time was dependant on the box size the player's position. However, for footsteps and weapon tail sounds, reverb was not applied instead sample switching was used to them. This not only saves valuable CPU power but also sounds more realistic.

3.3.5 Gun Sound Implementation

Implementing gun sounds might seem straight-forward, but in Zero Caliber it is quite a complex system. Just like everything else that had been discussed so far it came with a lot of experimentation and tweaking. There are dozens of parameters that are being driven during gameplay that are enhancing the immersion. First, it is very important to have the player gun sounds separated from the non-player gun sounds. This way, it is easier to edit and organize the events in FMOD and the player's sounds can belong to a different ducking group. Thus the player's weapon side-chain compresses the non-player weapons so there is more emphasis on the first person sounds.

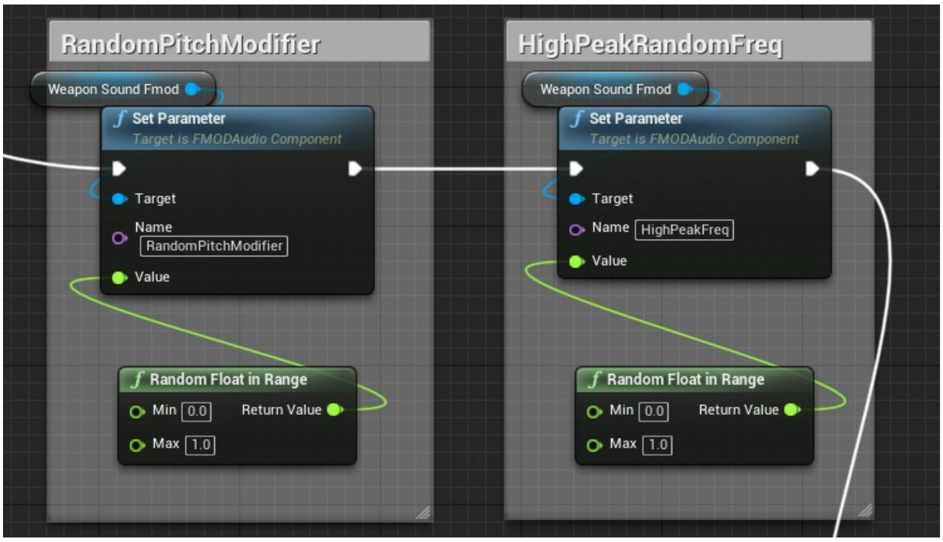
These events are built up by layers in FMOD. Player and non-player events are slightly different, as non-player weapons need distance-based sample switching. Non-player gun sound implementation is going to be discussed first.



PICTURE 19. Gun sound layers in FMOD Studio (Photo: Zsombor Bencsik)

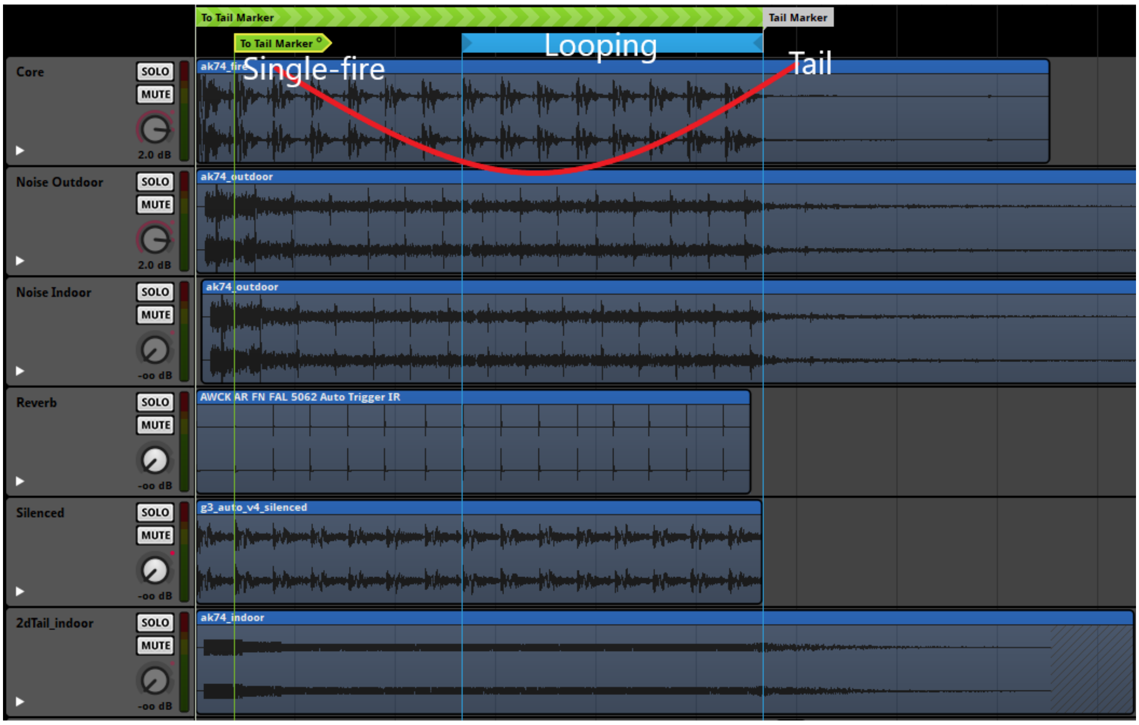
As examined in the Gun Sound Design chapter good gun sounds are made of layers. There are four distances for each layer: Close, Mid, Far, and Very Far and they are being crossfaded based on the proximity to the player. Besides volume automation, there is a low-pass filter that is being triggered by distance, so the further away the sound is the more filtering is applied.

Zero Caliber has approximately 25 guns, so having so many distance layers for non-player gun sounds was not an option due to a lack of time and resources. Instead, six events were used for each gun type: heavy, rifle, SMG (Sub Machine Guns), HMG (Heavy Machine Guns), pistol, and sniper. To add more variety and randomization, each event instance gets a random number between 0-1 at creation that sets their pitches based on this parameter along with some eq bumps.



PICTURE 20. Setting parameters in Unreal Engine. (Photo: Zsombor Bencsik)

The firing sound consists of two separate events, one being the firing sound itself and the other one is a tail reverberation. The reason they are separated is mainly for easier editing. Firing sound is going to be discussed first.



PICTURE 21. Player gun event in FMOD Studio. (Photo: Zsombor Bencsik)

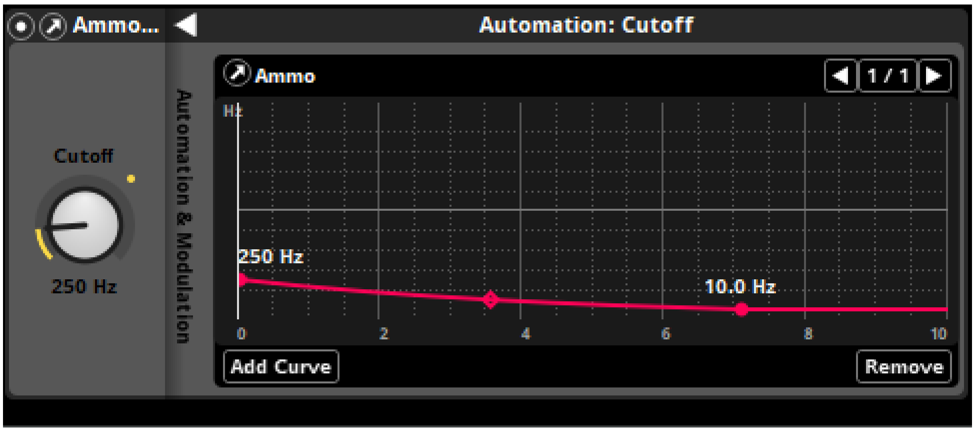
As mentioned before, player gun sounds are somewhat different in both Fmod and unreal engine implementation. There is no need to use the distance layer as the sound can only play in the player's arm length. The events in Fmod are primarily made of 5 layers: Core, Noise Outdoor, Noise Indoor, Reverb, and Suppressed (Picture 21). However, noise indoor and suppressed only being added if their parameter is called and set. Instead of looping 1 gunshot sound, a longer firing sound is used then looped in the end.

When firing is stopped, the playhead in the event jumps to the Tail Marker, and the tail of the gun sound is being played back. Since Zero Caliber is a realistic shooter all the automatic guns have a single-fire mode. If single-fire is selected, a parameter is passed to FMOD and the playhead jumps to Tail after reaching To Tail Marker as displayed on the picture above (Picture 21).

The first layer is the core sound that consists of the layers discussed in 3.1.1 Gun Sound Design. Following that, there is a noise layer that plays back the gun sort of blooming in its environment, whether if it is indoors or outdoors. The reverb layer (Picture 21) is quite important as it places the gun into a space. For this purpose, convolution reverb is used as it provides more realistic acoustic space than any algorithmic reverb. When the whole gunfire audio was sent to the convolution reverb, the sound was very muddy and unclear, so instead of sending everything, a trigger pop sound aligned (Picture 21) with the gunshots was sent only to the reverb. This produced the most believable and sonically pleasing sound. The last layer is the silencer layer that mixes in some silencer sounds with the core layer and significantly lowers the amount of noise the gun has since when the gun is suppressed there isn't as much acoustic feedback from the environment.

Since Zero Caliber is a VR game, spatialization is a key factor to achieve immersion. I experimented dozens of hours trying to get player gun sound spatialization right and concluded: that the noise and reverb layers should be 2D without any positioning, while the core and silencer layer should be 3D. This way the acoustic feedback can be heard as a wide stereo sound, while the punchy characteristic core sound is spatialized and attached to the gun itself.

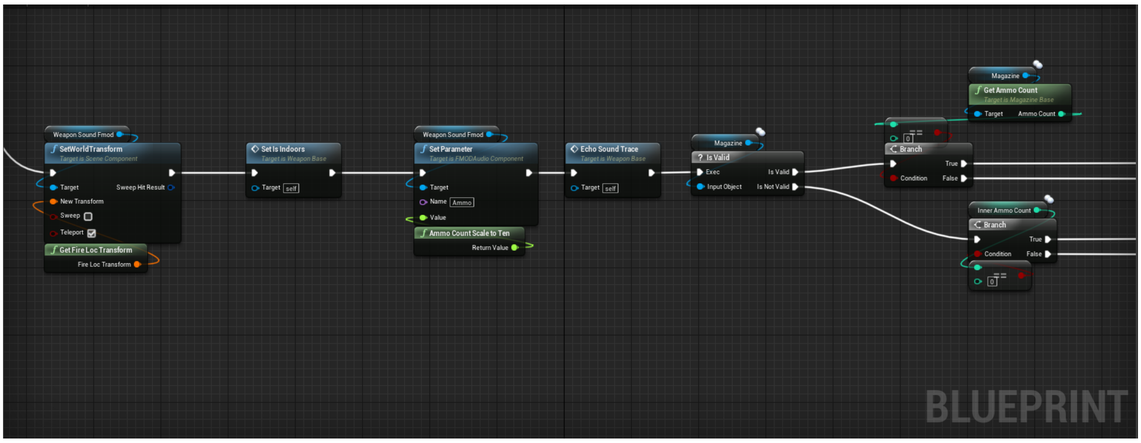
There is no UI in the game so the player doesn't know how much ammo he has left in the chamber. I thought that audio feedback is going to be necessary to provide some kind of information about the ammo. To tackle this problem, I used high-pass filtering that filters more low end as the ammo gets lower. This happens through an Ammo parameter that is being set every time a bullet leaves the gun, then the filter is automated via the param.



PICTURE 22. Highpass filtering via Ammo parameter.

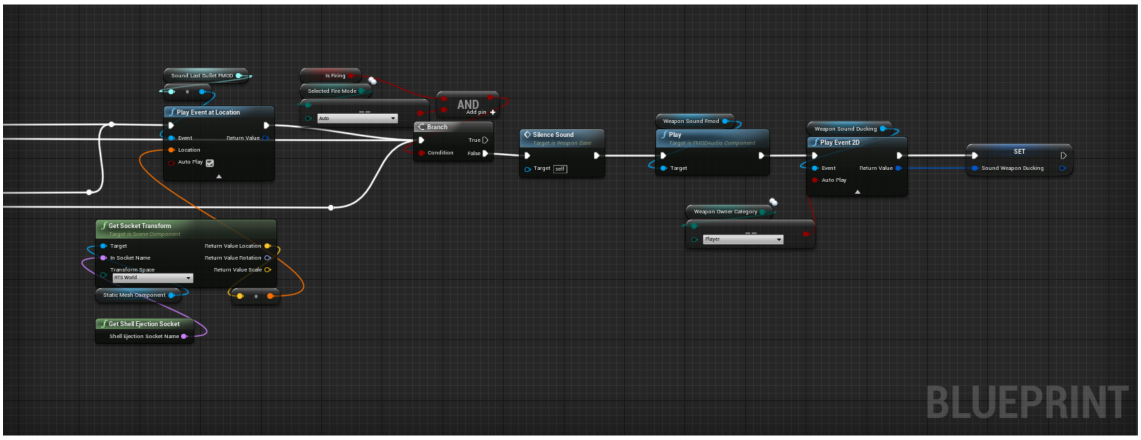
When the parameter value reaches 0 the filter is around 250 Hz, so the gun gradually loses its power as being fired.

The shooting logic happens in Unreal Engine using Blueprints. It can seem overwhelming and complicated first, but once one understands the basics, it becomes an easy and handy tool.



PICTURE 23a. Playing gun fire and setting parameters in Unreal Engine (Photo: Zsombor Bencsik)

Numerous parameters are being set while firing each bullet in the blueprint. As seen in the pictures (Picture 23a) the transform of the weapon is being set first since the position of automatic gun sounds needs to be updated every shot, otherwise, they would remain in the same position during firing. After that, the indoor/outdoor parameter is set, depending on the player’s location and the Ammo parameter (Picture 23a) is set as well.

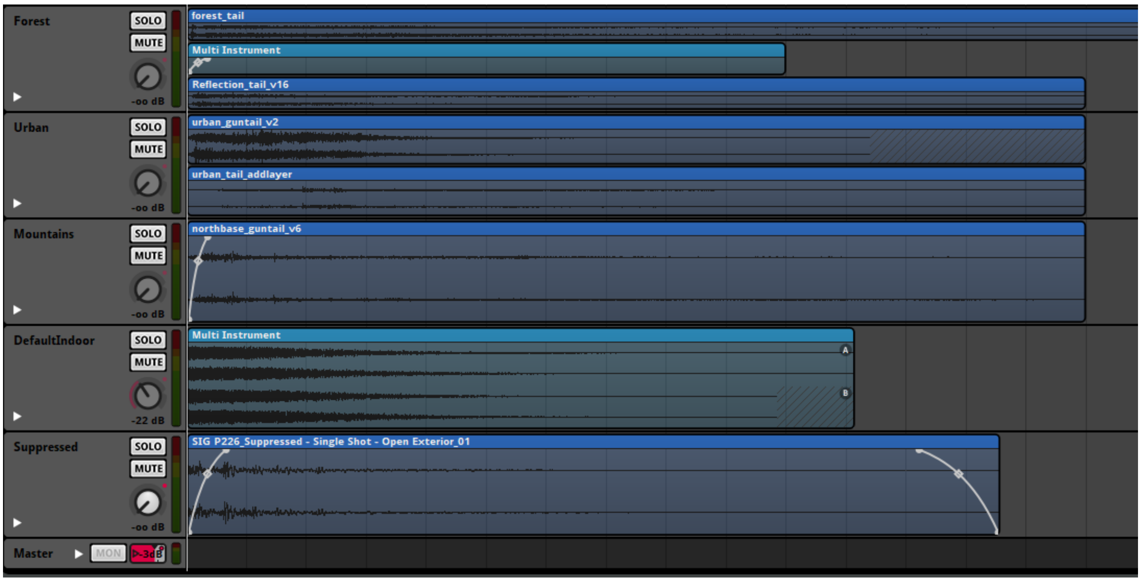


PICTURE 23b. Playing gun fire and setting parameters in Unreal Engine (Photo: Zsombor Bencsik)

Next, there is a check to see if there’s any ammo left. If not, an ammo-out sound is played (Picture 23b) which is crucial, for the player to know that he is out of ammo. If the gun has a silencer on, the silencer parameter setting happens here, and at last, the sound is played with the configured parameters.

The second event is the gun tail sound which is played after firing the gun. It is 2D for the player, while 3D for the non-player characters. Setting the 3D attributes for non-player characters was incredibly challenging because a mono gun-tail didn’t sound right. Ultimately, it was solved by automating the stereo width of non-player tail sounds by distance. Other than this automation and some distance-based filtering, the events were almost identical for the player and non-player events in Fmod.

Firing a gun in an urban area or by a mountain produces a very different sound in real life. Thus in Zero Caliber there are 4 types of environments that have a different sound: Forest, Urban, Mountains, and Indoor. When firing the gun, a parameter is set to select one of these environment types.



PICTURE 24. Tail sounds in Fmod Studio (Photo: Zsombor Bencsik)

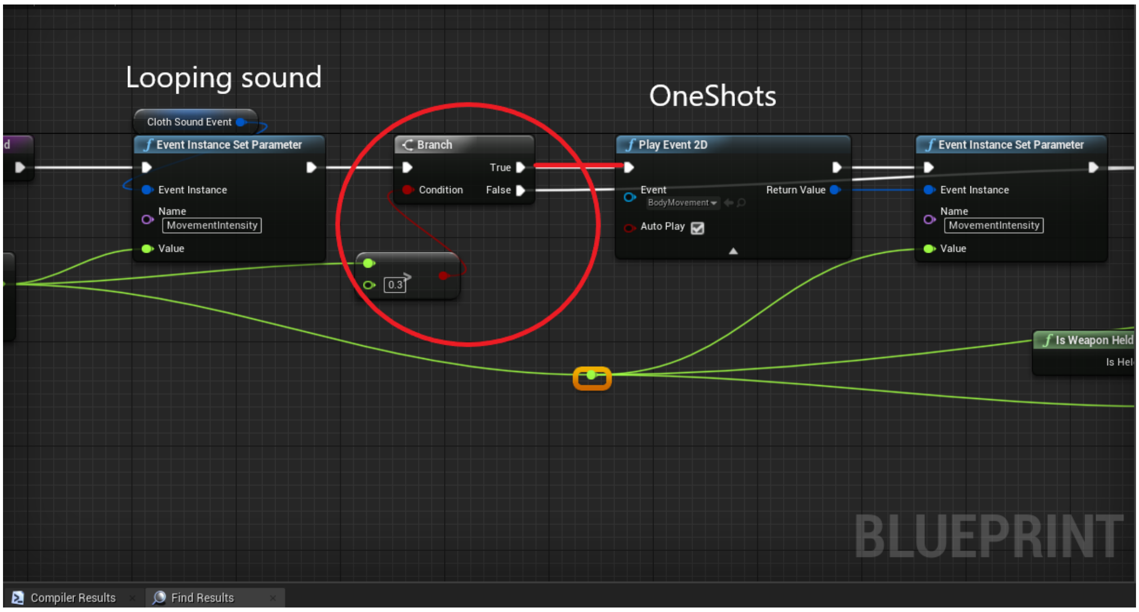
When the parameter set, the respective audio channel is un-muted and played. If the player changes his environmental location, for instance, goes indoors, the previous channel is muted and only the indoor sound plays. If the gun is silenced, the tail sound will be heavily lowered in volume and a Suppressed layer will be mixed in (Picture 24).

To differentiate this sound for each gun, a Strength parameter was given to control the tail sound's volume and pitch. For small pistol, the volume is lower and the pitch is higher, while it is the exact opposite for bigger rifles.

3.3.6 Character Foley Sound Implementation

Most of the VR games on the market use only hands and no body as the player character, however in Zero Caliber the player has a full body that responds to controller movement. In non-VR games character foley is implemented 2D for the player such as footsteps and cloth movement sounds. If the player is standing still, the sound is usually attached to its idle animation, but in VR there is no animation for it and the player has more input than just a mouse and a keyboard. When moving the VR controllers the character model moves its arms while rotating the elbows and shoulders. The VR games I have tried did not have sound for these kinds of movements. For Zero Caliber movement detection was implemented, which calculates the controller's velocity every 0.1 sec. This way when the player moves his arms it produces audio feedback that makes the experience more real and immersive.

In FMOD there are two movement sound events for this purpose. The first one is a loop of the sound of clothes rubbing against each other. A parameter called MovementIntensity is being set every 0.1 sec and the volume of the cloth rubbing is being increased or decreased depending on the MovementIntensity parameter. There is also another event that contains cloth whooshes one-shots. However, it only triggers if the MovementIntensity reaches a certain threshold.

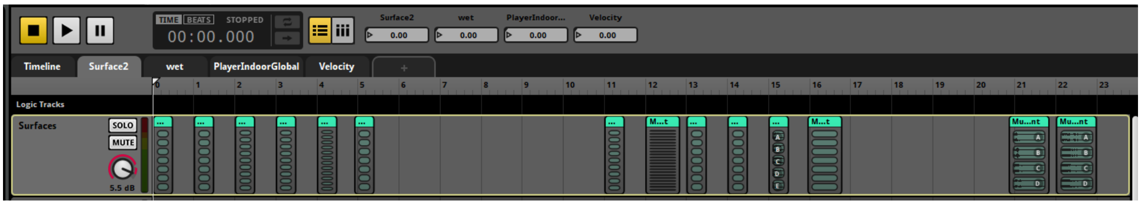


PICTURE 25. Movement sound blueprint in Unreal Engine (Photo: Zsombor Bencsik)

The one-shots only play if the intensity is more than 0.3 as displayed on the picture (Picture 25). For instance, reloading a gun quickly produces the sound of cloth and gear rubbing together. It's worth mentioning that this feature is only implemented for the player character for performance reasons.

To enhance realism, heavy breathing sound added to the player after a certain time passed while running. This creates the illusion that the player doesn't have any superhuman capability as he runs out of air.

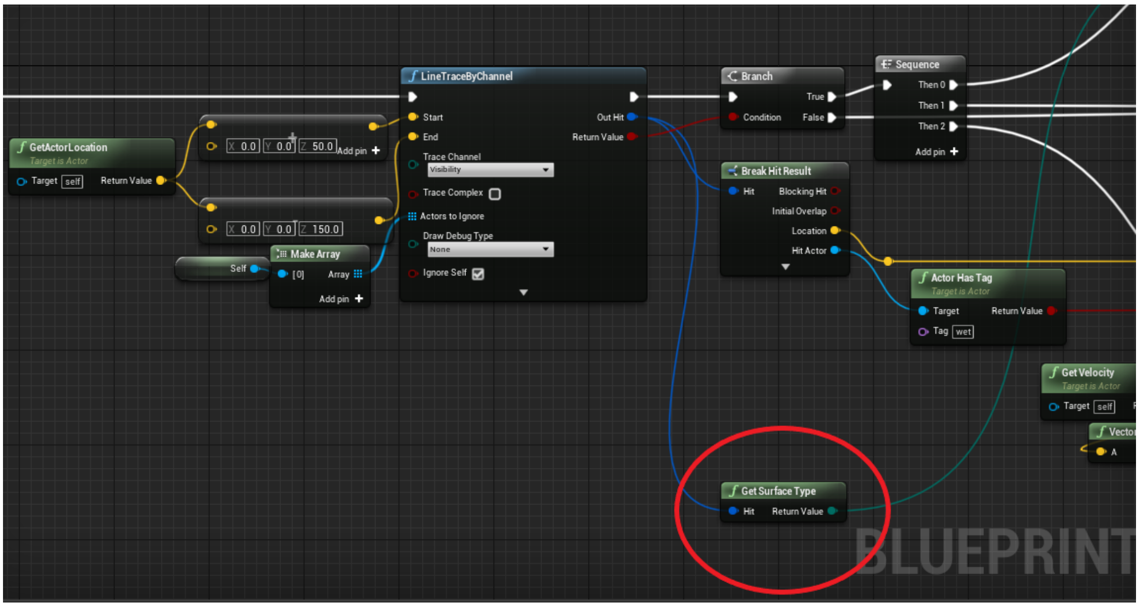
Good footstep sounds are key in VR, the audio design of these were discussed in 3.1.3 Foley chapter. In non-VR games, player footsteps are usually played 2D non-positioned, but in Zero Caliber they are 3D positioned so when the player is moving his head, he can hear a subtle difference in the positioning of the footsteps.



PICTURE 26. Footsteps on different materials in FMOD Studio (Photo: Zsombor Bencsik

Each surface type has its own sound (e.g. wood, metal, water, dirt, concrete, glass), that is being switched with a parameter called Surface. There are also three more parameters: Wet triggers some water sounds if enabled, PlayerIndoors adds reverb and swaps to an indoor footstep sound, Velocity adds volume to the sound if increased depending on the velocity of the player (Picture 26).

Implementing the footstep sounds in Unreal Engine consists of multiple elements. First, the surface material needs to be acquired to set the parameter for Fmod.

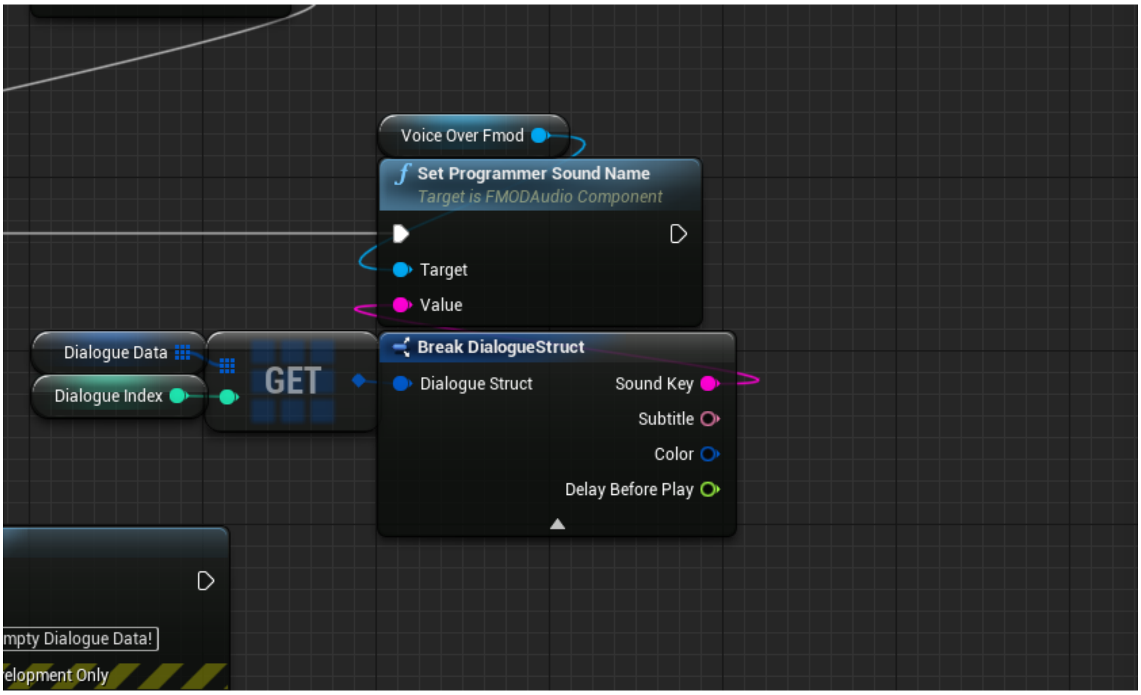


PICTURE 27. Line-tracing for surface materials in Unreal Engine (Photo: Zsombor Bencsik)

To do this, a simple line-trace is being cast down from the player's position and on collision with the ground, it gets the surface material (Picture 27). After confirming the player's foot hit the ground, the footstep event gets configured with parameters mentioned before and then played. The gear foley sound that happens for every footstep gets played in this logic as well.

3.3.7 Voice Over

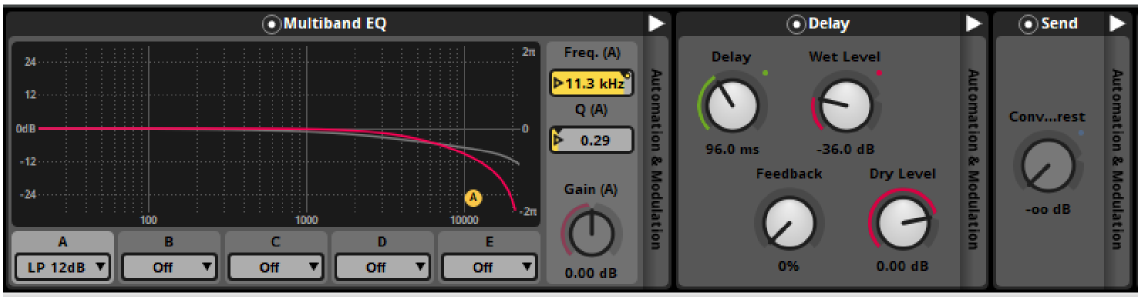
Voice over is an essential part of a video game's audio design. There are two types of voice-overs in Zero Caliber: important narration and battle chatters. The two are implemented slightly differently as story narration has to be heard above all, while battle chatters belong to a different less important group channel.
The main storyline narrations are short 2-5 second lines that are imported into an AudioTable in FMOD. Narration is played back in 2D stereo. A Programmer Instrument is used to play the narration, as it can choose and play files from the audio table. This means that there is only one FMOD event for all the narration instead of having to use hundreds of single events.



PICTURE 28. Setting the Programmer Instrument Soundkey (Photo: Zsombor Bencsik)

Soundkeys need to be set for FMOD to know which audio file to choose from the AudioTable. This means every voiceover that is placed on the level needs its soundkey to be configured.

Battle chatters are the phrases and sounds of soldiers asking for cover, medic, ammo, dying, shouting, etc.. These sounds are attached to the non-player characters and are 3d spatialized. As there are enemy soldiers and friend soldiers, these have separate events in FMOD. A parameter is passed through the event that selects the chatter type (reloading, cover, backup, flanking, etc..). Processing the sounds to sound distant was a challenging task. Voiceovers that are recorded in the studio were very difficult to process to sound as if they were outdoors.


Multiband EQ

Freq. (A) 11.3 kHz

Q (A) 0.29

Gain (A) 0.00 dB

A LP 12dB

B Off

C Off

D Off

E Off

Delay

Delay 96.0 ms

Wet Level -36.0 dB

Feedback 0%

Dry Level 0.00 dB

Send

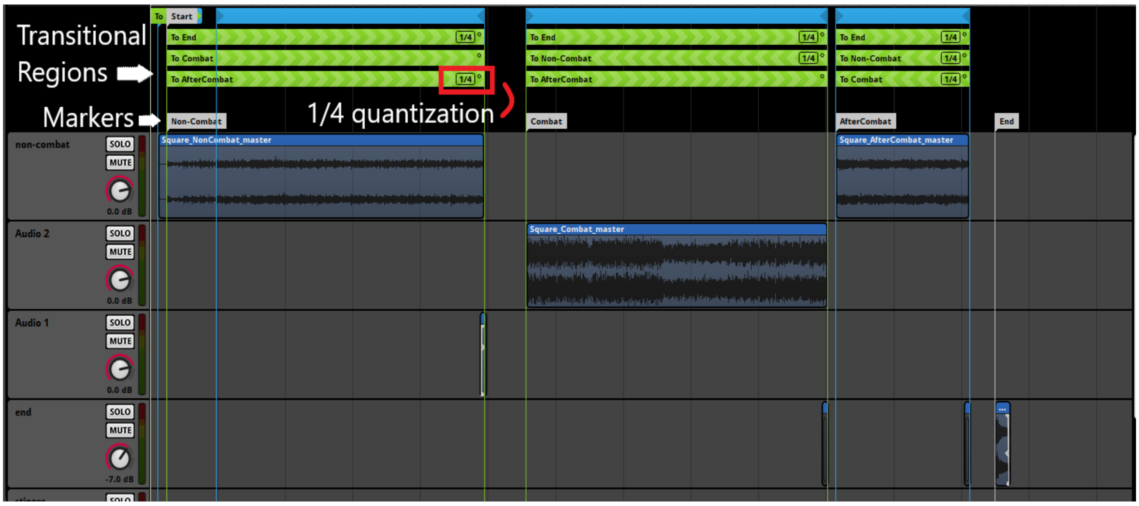
Conv...rest -oo dB

PICTURE 29. Processing chain for battle chatter in FMOD Studio. (Photo: Zsombor Bencsik)

Filtering, slap-delay, and convolution reverb were used to emulate distance. The amount of effects depends on the distance to the player, so if the player is further away the more filtering, delay, and reverb is applied. In Unreal Engine, chatter type, indoor/outdoor, and indoor parameters are set up before playing the event and if the NPC is indoors the convolution reverb is switched to indoor convolution.

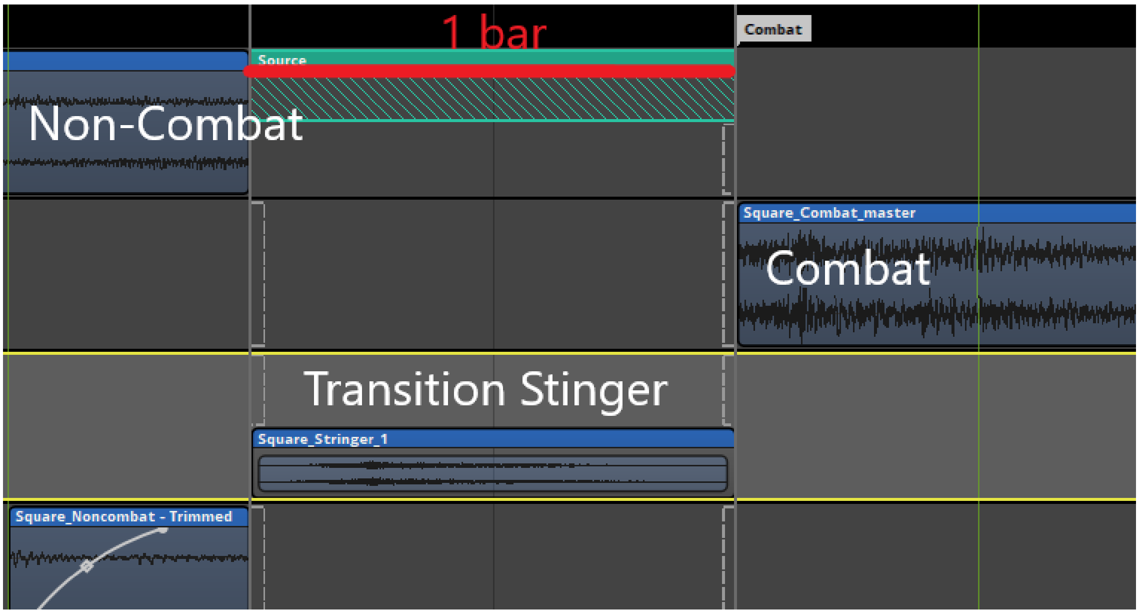
3.3.8 Music System

Music Systems in video games are very intricate systems that take information from various events in-game. In Zero Caliber four states are driving the music: non-combat, combat, after combat, and end play. During non-combat, more relaxed and anticipating music is playing while combat music makes the whole experience more fast-paced and intense. The After-Combat state is keeping up the tension after combat ends while End-Play plays a stinger that suggests that the level has been completed.



PICTURE 30. Adaptive music event in Fmod Studio (Photo: Zsombor Bencsik)

The FMOD events seem quite overwhelming at first but once understood, they're easy to grasp. Four loops are imported to the event (non-combat, combat, after-combat, and end-play) that are placed next to each other so they won’t play at the same time. Markers are placed at the beginning of each audio file. Markers are the points to jump when the state changes (Picture 30). For the duration of the audio files, there are several transitional regions (Picture 30) that if given parameters are jumping to the selected marker. For example, the non-combat sound file has three transition regions above it: To Combat, To After Combat, To End. This means when non-combat music is playing, and the state parameter is changed to combat, the playhead will jump to the combat marker. However, jumping immediately would sound offbeat and too sudden. For this reason, transition regions have quantization options built into them (Picture 30), so after changing the state parameter, the playhead would only jump on the next ¼ beat. Additionally, transition stingers are used to make transitions sound more natural.



PICTURE 31. Transition timeline in FMOD Studio. (Photo: Zsombor Bencsik)

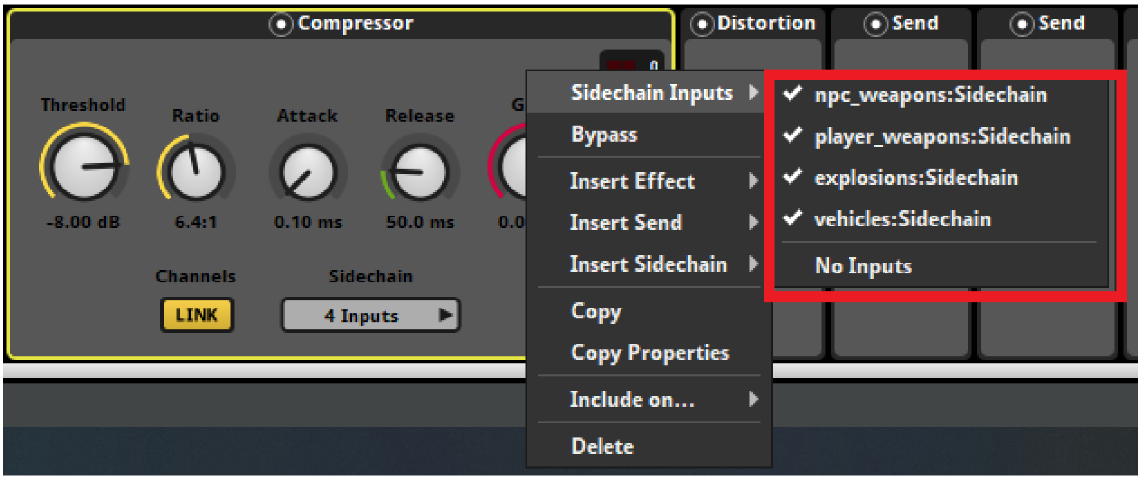
Transition Timelines exist only between transitions, so when the state parameter is changed, the playhead waits for the next ¼ quantization point then jumps to the transition timeline (Picture 31) where a stinger is played while the previous state continues for 1 bar jumping only after to the new state. This makes a seamless and natural transition.

The music system in Unreal Engine is flexible and can be controlled manually and automatically. If it’s set automatically, the different states change based on the number of enemies near the player. Sometimes manual control is necessary for artistic or technical reasons, for example, some level doesn’t need non-combat music only combat music. This can be set up in the Music System blueprint.

3.3.9 Clear Mix

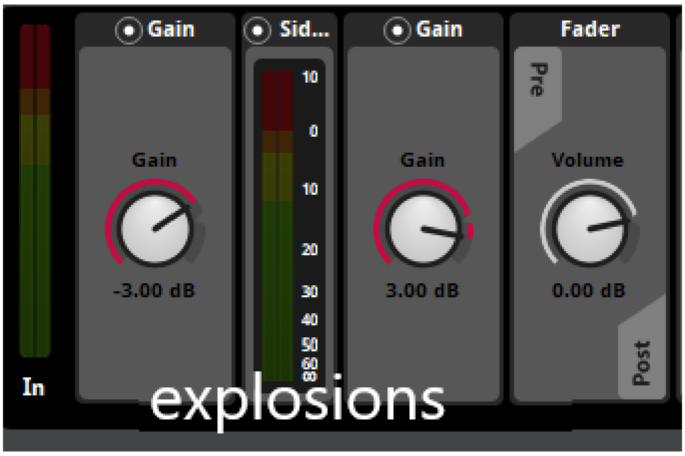
Mixing a game is usually the last process during audio production. However, the importance of mixing cannot be underestimated. Having great sounds individually is not enough, as the sounds have to work as a cohesive soundscape during gameplay. The key to an immersive VR audio experience is dynamics in my opinion. Quiet sounds should be very quiet while loud sounds like guns and explosions should be very loud. Ambiances are around -45 dB, music is -25 dB, player gun sounds are -10 dB, and explosions are hitting the 0db on a limiter. The overall loudness of the game is -17 LUFS. It is important to use LUFS instead of true peak when measuring long term loudness of a video game.

A lot of prioritization decisions have to be made during mixing. In Zero Caliber, the main focus is on player guns, so they have high importance while shooting. Ambient sounds, foley, battle chatter, weapon foley gets heavily side-chained when firing a gun.


Threshold -8.00 dB
Ratio 6.4:1
Attack 0.10 ms
Release 50.0 ms
Channels LINK
Sidechain 4 Inputs
✓ npc_weapons:Sidechain
✓ player_weapons:Sidechain
✓ explosions:Sidechain
✓ vehicles:Sidechain

PICTURE 32. Sidechain inputs on the duck_extreme group channel (Picture: Zsombor Bencsik)

The compressor on the duck_extreme group has 4 inputs; npc_weapons, player_weapons, explosions, and vehicles (Picture 32). The reason for not grouping those subgroups is to have greater control over how much I want to compress. Since it is not possible to control each input's gain in the compressor. The workaround for this issue is adding a Gain plugin before the side-chain send on the input groups.

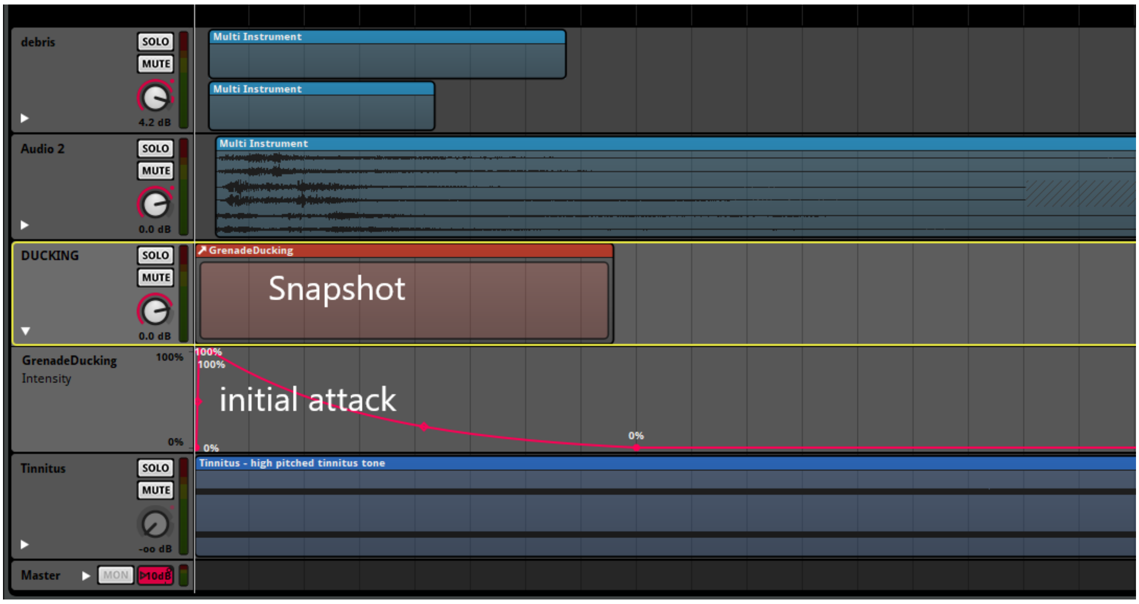

Gain -3.00 dB
Gain 3.00 dB
Volume 0.00 dB

PICTURE 33. Side-chain input gain in FMOD Studio (Picture: Zsombor Bencsik)

To avoid too much sidechain input, the gain is lowered by 3 dB before the sidechain send and re-added after on the explosions group (Picture 33). Other than that, the Ambience group is also being high-pass filtered, to leave space for gunshots and explosions in the lower frequencies.

Besides the ducking system, mixer snapshots are used to control individual group volumes. Snapshots are sets of values for bus properties that can be applied to your game's mix on the fly, allowing your game to change its mix as it runs. (FMOD Documentation 2019.) They can alter the game mix instantly. For instance, there is a WeaponDucking snapshot in Zero Caliber that turns down 6db of ambient sounds. This is a static volume reduction addition to sidechain compression which is a dynamic volume reduction.

When playing extremely loud sounds such as explosions, a short explosion duck snapshot is triggered, so for the initial transient of the explosion, every other sound is killed.



PICTURE 34. Triggering snapshots in FMOD Studio (Photo: Zsombor Bencsik)

Snapshots can be triggered in Unreal via Blueprint or in Fmod via snapshot instrument as displayed on the picture (Picture 34). The explosion snapshot intensity is automated, so in the first 100ms it is 100%, after it fades out bringing all the ducked sounds back. The snapshot intensity is also automated by distance, so it only triggers if the explosion happens close to the player. Snapshots are also used to lower music during voiceover playback, trigger reverbs, and apply filters to different groups. They are incredibly handy and a must-have tool during game mixing.

Fmod has the ability to monitor and tweak during gameplay which means it is possible to set levels, reverbs, and filters while playing in-game. Soloing and muting individual channels and groups is extremely helpful also. More often than not, soloing and muting are used for debugging, as hundreds of sounds could be playing at once and sometimes there are going to be out of place sounds that need to be found and fixed.

Reality doesn’t sound right sometimes. During a firefight, there can be 20+ non-player characters shooting at each other making whole mix overcrowded and unclear. Setting the event’s maximum instances parameter can solve this issue. For enemy guns, a maximum of 8 events can play at once. Choosing which one to play can be dependant on several things but for this example, it’s based on volume. So louder guns are going to be prioritized and quieter ones are going to be silenced. The maximum instances technique is applied to several sounds such as non-player footsteps, battle-chatter, gear-foley, bird sounds, etc.. This not only makes the soundscape clearer but improves performance as well. During gameplay, a total of 64 events can play simultaneously, all the rest are silenced.

4 DISCUSSION

There has been a steady growth in the video game industry resulting in the demand for more and more audio professionals according to the articles, data, and projections I’ve read. The number of indie game studios opening and job listings are getting higher every year, so making a living as a game audio designer is now more viable than ever.

A lot of indie game companies cannot afford to get multiple people on their audio team, so they often opt for designers who have expertise in several fields of audio such as sound design, music, implementation, voice over. Being the only person responsible for audio in a game company is a very demanding job. However, when working alone, one can decide about artistic and technical decisions without any conflict. Being a jack of all trades is a must in this situation. It is not enough to be good at music and sound design as implementation is a huge part of audio production in a game as well. Being quick and efficient with an audio middleware is indispensable.

Bad audio design and implementation can be incredibly off-putting in Virtual Reality, thus the greatest care and effort need to be put into audio production when developing a VR video game. Sound Designers have to think about implementation before design. According to the audio director of the Battlefield franchise, dynamics and first-person sounds are the most important aspects in game audio when it comes to immersion. For the player to experience a real war-like scenario, guns, and close explosions should be extremely loud, while environmental, and foley sounds should be low in the mix. Music plays a huge part in immersion and must be used with the intention of magnifying emotions.

Sound design, music, and implementation can be taught, but because of competition, a lot of professionals keep their techniques and tricks to themselves. Me having to learn that by myself gave a purpose for this thesis. Music and sounds are very subjective, but I believe that the techniques presented in this thesis are a good starting point for those who began their journey on becoming an audio designer for games.

REFERENCES

Abrash, A. 2018. Chief Scientist at Oculus. Virtual Reality Predictions. Key-note presentation. Oculus Connect 5. on 26.09.2018. San Jose McEnery Convention Center.

Andersen, A. 2014. Game Audio: 4 golden rules for technical sound design. Read on 24.04.2020. https://www.asoundeffect.com/game-audio-4-golden-rules-for-technical-sound-design/

Bencsik, Zs. 2019. Audio Designer. Zero Caliber Soundtrack Playlist. Published on 21.08.2019. https://www.youtube.com/watch?v=EzQyPc9LpBU&list=PLcLn1mwVYe5XxtT_kX34hxqiy1JR_F_ha

Bridgett, R. 2009. A Holistic Approach to Game Dialogue Production. Read on 24.04.2020 https://www.gamasutra.com/view/feature/132566

Brosius, E. Audio Director at Harmonix. What Does an Audio Director Do. Read on 29.04.2020. https://www.berklee.edu/careers/roles/audio-director

Chance, T. 2017. Composing Music For Games. CRC Press, p.3.

FMOD. 2019. Documentation about Global Parameters. Read on 15.04.2020. https://www.fmod.com/resources/documentation-studio?version=2.1&page=parameters.html

FMOD. 2019. Documentation about Snapshots. Read on 17.04.2020. https://www.fmod.com/resources/documentation-studio?version=2.1&page=mixing.html#snapshots-and-the-tracks-view

Frank, M. &. Z. F. &. S. A., 2015. Producing 3D Audio in Ambisonics. s.l., s.n., p. 8.

Jones, A. Dialogue Editor. 2019. How to get work in video game audio. Video. Published on 29.08.2019. Referred on 29.04.2020 https://thinkspaceeducation.com/how-to-get-work-in-video-game-audio-p1-g1/

Kastbauer, Damian. 2017 Game Audio: Tales of a Technical Sound Designer Volume 01 . Kindle Edition

Kim, Y. Schmidt, E. Migneco, R. Morton, B. Scott, J. Speck, J. Turnbull, D. 2010. State of the Art Report: Music Emotion Recognition: A State of the Art Review.

Maguire, P. Sound Designer (Games and Tech). What Does a Sound Designer Do. Read on 10.04.2020. https://www.berklee.edu/careers/roles/sound-designer-games

Marks, A. 2017 Complete Guide to Game Audio: For Composers, Sound Designers, Musicians, and Game Developers. 3rd Edition.

Masterclass, 2019. Diegetic Sound and Non-Diegetic Sound: What's the Difference?. Read on 10.04.2020. https://www.masterclass.com/articles/diegetic-sound-and-non-diegetic-sound-whats-the-difference

Mohamed, Theron. 2019. 5 ways the video-game industry will change in the next decade. Read on 24.04.2020. https://markets.businessinsider.com/news/stocks/5-ways-the-video-game-industry-will-change-in-the-next-decade-2019-7-1028394586#streaming-gameplay1

Osgyani, D. 2020. Zero Caliber Gameplay. Video. Published on 11.05.2020. https://www.youtube.com/watch?v=_-5hscuHkwY

Pajor, B. Audio Director. 2018. Designing punchy gun sounds. Email message. Read on 16.08.2018

Pajor, B. Audio Director. 2020. Creating immersive and realistic sounds for a shooter game. Email message. Read on 21.04.2020

Quarles, Z. 2009. Game Audio Theory: Ducking. Read on 02.05.2020. https://www.gamasutra.com/view/feature/132310/game_audio_theory_ducking.php

Reed, J. 2015. Hollywood vs. real weapons sounds. Read on 05.05.2020. https://johntreed.com/blogs/john-t-reed-s-blog-about-military-matters/65654339-hollywood-vs-real-weapons-sounds

Schmidt, B. 2018. 6 Surprising Facts About Composing Music for Video Games. Read on 01.05.2020. https://www.gamesoundcon.com/post/2015/08/08/6-surprising-facts-about-composing-music-for-video-games

Schütze, S. 2018. New Realities in Audio: A Practical Guide for VR, AR, MR and 360 Video. s.l.:CRC Press.

Smith, S, PhD. 1999. The Scientist and Engineer's Guide to Digital Signal Processing Read on 24.04.2020 https://www.dspguide.com/ch22/1.htm

Taylor, G. 2016. Audio Director at Sony Interactive Entertainment Europe. How to unlock the creative power of audio in VR. Read on 02.05.2020 https://www.asoundeffect.com/creative-audio-for-virtual-reality/

Ubisoft Entertainment. Technical Sound Designer Job Listing. Read on 01.05.2020. https://jobs.smartrecruiters.com/Ubisoft2/743999694757294-technical-sound-designer

Unreal Engine. Blueprints Visual Scripting. Read on 16.04.2020 https://docs.unrealengine.com/en-US/Engine/Blueprints/index.html

Zimarev, R. 2018 UI Sounds: From Zero To Hero. Read on 24.04.2020 https://icons8.com/articles/ui-sounds/

APPENDICES

Appendix 1. Interview Bence Pajor

1. **What is the most important thing when creating audio for realistic shooter games?**

Make sure that all the sound has the same style, everything should be believable, sounds should be built around the most important sounds without drawing attention from it. Weapon, explosion, and first-person sounds should sound the best and all the other sounds should accentuate and highlight them. It should be joyful to fire a gun. It is important that every sound share the same space in the world and sounds should leave a mark in the world, this is important for the player to belive that the sounds are really loud as simply turning up the volume is not enough.

2. **All of your guns have an incredible low end and punch; are these coming from real-life recordings or are this generated synthetically?**

Everything is real recordings, we always use a lot of microphones. We manage to create the low end punchy because we take out the low-end from everything else.