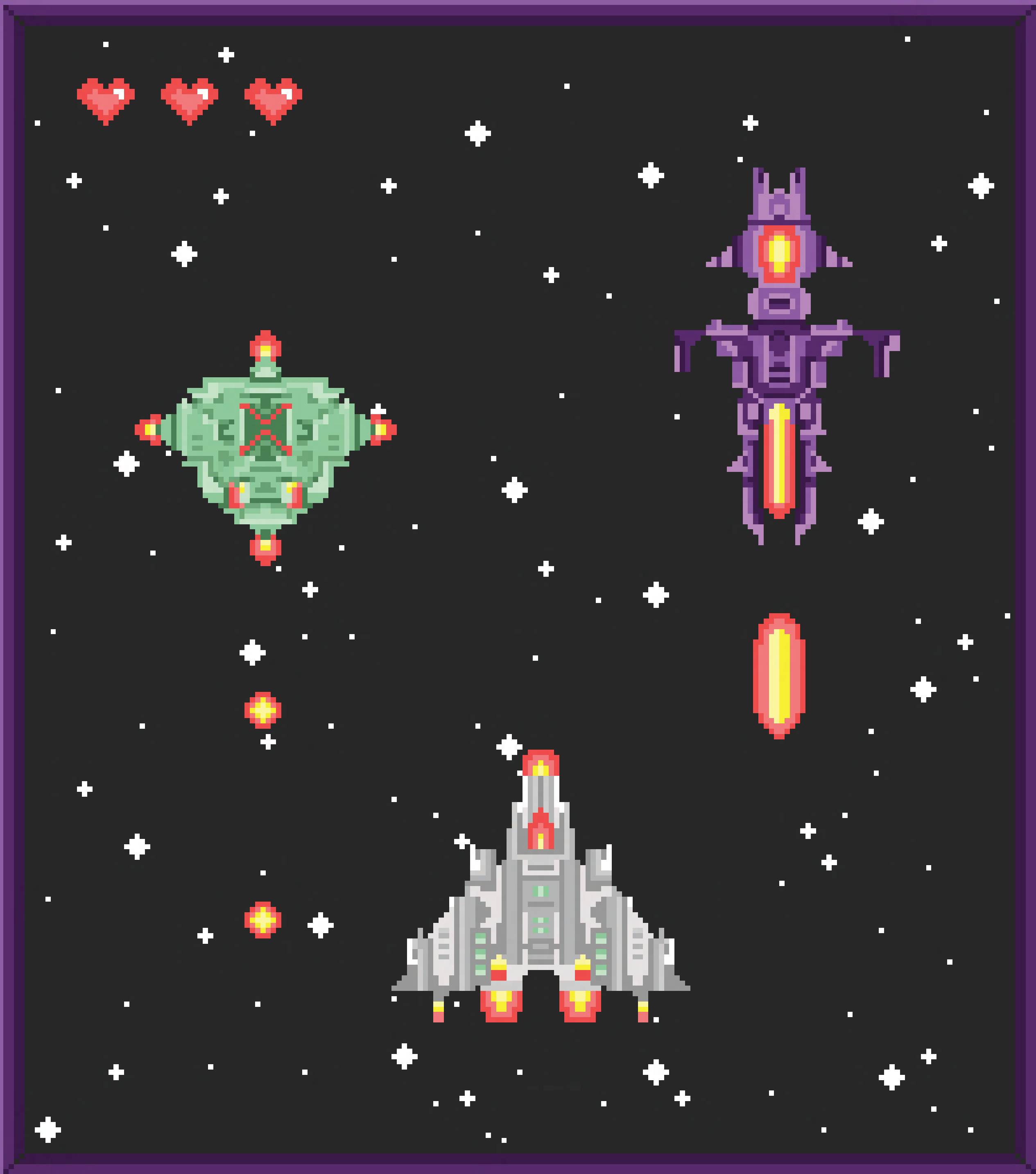


Wolfgang Beucher

Changing the Difficulty of Video Games through Sound Design

Changing the Difficulty of Video Games through Sound Design

Wolfgang Beucher

Master Thesis in Sound Design
FH Joanneum University of Applied Sciences Graz &
University of Music and Performing Arts Graz
Supervision Josef Gründler

September 9, 2019

Keywords:
Video Game, Sound Design, Game Sound, Game Audio, Game Design, Difficulty, Transdiegetic, Auditory Display, Informative Sound, Sonification

Abstract

The interactive component of video games demands that the subject of difficulty is one of the chief points in designing them. In their beginning the focus was to make the most money off arcade cabinets or to extend the playtime of otherwise short games. Nowadays the main concern should lie on creating the best experience possible, which can be accomplished by a high, low or balanced difficulty. Sound design is already widely used to form the difficulty level, but there is insufficient study about this.

Through well-executed sound design, the player is not only influenced in his or her aesthetics perception of the game's story or world, on top of this he or she automatically takes in various hints like warning sounds. Through the auditory layer the game designer can forward important information to the player without cluttering the graphical interface. Game mechanics can be made more clear or easier learnable, and general player performance can be boosted.

In this master thesis the existing research surrounding this topic will be summarized and presented. For the workpiece a video game was created in which many practices from the theoretical part were included.

Abstract

Durch die interaktive Komponente von Videospielen liegt einer der zentralen Punkte des Designs auf dem Schwierigkeitsgrad. Einst lag der Fokus darauf, möglichst viel Geld am Arcade Automaten zu verdienen oder kurze Spiele künstlich zu strecken. Heutzutage sollte mehr darauf geachtet werden, das bestmögliche Spielerlebnis zu schaffen, sei es durch einen hohen, niedrigeren oder ausgewogenen Schwierigkeitsgrad.

Sound Design ist eine viel verwendete aber noch nicht gründlich erforschte Variable beim Formen des Schwierigkeitsgrades. Durch gut ausgearbeitetes Sound Design wird der/die Spieler*in nicht nur in der Wahrnehmung der Geschichte und Welt des Spieles beeinflusst, sondern nimmt auch automatisch verschiedene Hilfestellungen wie Warnlaute auf. Durch die auditive Ebene können wichtige Informationen vermittelt werden, ohne das grafische Interface zu überladen. Spielmechaniken können eindeutiger oder leichter erlernbar gemacht werden, außerdem kann die generelle Leistungsfähigkeit verstärkt werden.

Die vorhandene Literatur rund um das Themengebiet wird in dieser Masterarbeit zusammengefasst und präsentiert. Im praktischen Teil wurde ein Videospiel erstellt, in dem viele Praktiken aus dem Theorieteil exerziert werden.

Eidesstaatliche Erklärung

Ich erkläre ehrenwörtlich, dass ich die vorliegende Masterarbeit selbstständig angefertigt und die mit ihr verbundenen Tätigkeiten selbst erbracht habe. Ich erkläre weiters, dass ich keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle aus gedruckten, ungedruckten oder dem Internet im Wortlaut oder im wesentlichen Inhalt übernommenen Formulierungen und Konzepte sind gemäß den Regeln für gutes wissenschaftliches Arbeiten zitiert und durch Fußnoten bzw. durch andere genaue Quellenangaben gekennzeichnet.

Die vorliegende Originalarbeit ist in dieser Form zur Erreichung eines akademischen Grades noch keiner anderen Hochschule vorgelegt worden. Diese Arbeit wurde in gedruckter und elektronischer Form abgegeben. Ich bestätige, dass der Inhalt der digitalen Version vollständig mit der gedruckten Version übereinstimmt. Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben kann.

Acknowledgements

I want to thank my family for supporting me all this time, even when pursuing weird niche topics like game sound design.

Thanks to my supervisor Josef Gründler for his feedback and for helping create this master study in the first place.

Thank you, Christina Cossee, for helping with the visuals of the game and for being there for me.

And thanks to all the people doing research and sharing knowledge.

Table of Contents

Preface

Difficulty in different games takes various forms. *Dark Souls* (2011, Bandai Namco) is said to be very hard, with no possibility to change this fact, apart from "getting good". *Cuphead* (2017, Studio MDHR) features an easy mode, but not playing the hard mode restricts you from finishing the game. *Resident Evil 4* (2005, Capcom) has a hidden difficulty meter where the game adjusts itself depending on the skill of the player. Five different modes are selectable in *Wolfenstein: The New Order* (2014, MachineGames), but the game taunts you for choosing the lower two. But how does sound play into this subject?

There has been some research on the effect of sound design in this area, for example in the form of perceived difficulty depending on audio (Darth 2015) or supporting the player through musical cues (Östlund 2015). The related subject of sonification has also been getting more attention in the last few years (Hermann et al. 2011).

As it is the fact in other media, sound plays an important (and often overlooked) part in games as well as in game design. Displaying certain information through the auditory layer cleans up the visuals and helps players to focus on other important visual cues. If this is executed well, it should make the game easier or harder depending on the sound designers' intentions.

In this master thesis the research surrounding this topic will be presented and tested in form of a video game developed for this thesis. Various practices on how to change the game's difficulty just through sound will be explored. Also, the sound design and audio implementation work will be documented.

1. Research
1.1 Introduction

There has been extensive research on game sound regarding it being a relatively new and niche topic. Video games have been a big industry for some time and are a fascinating theme for research due to their differences to old media like film. The player is given control to alter what is happening in various degrees instead of just following the story. This leads to a different kind of immersion than in films:

> **In games with a protagonist, gameplay emotions may equal care for the protagonist, but this care is essentially different from empathetic emotion: From the perspective of gameplay, the protagonist is a means, a tool, for playing the game (and achieving the personal goal of completing the task).[1]**

If the protagonist fails, we fail, if he or she succeeds, we can feel proud of ourselves. Game designers (most of the time) want us to feel this sense of accomplishment, and, as this thesis is going to explore, sound is a very powerful tool to do this.
Games are evolving from their simple origins and their sound does so, too:

> **As game design becomes more complex in structure and quality, the use of sound as well as graphics to deliver information has grown in significance. No longer is sound simply a means to create mood and enhance to the look and feel of the game. Sound increasingly has a role to play providing the player with critical information.[2]**

To know how we should use sound, we first must know why we should use sound, and like most things there are pros and cons of using it. This becomes obvious when stripping away other senses, like in the audio game genre where most of the time there are no visuals, the focus lies on sound.

> **Sound has different strengths and weaknesses than graphics. It does something better, like conveying a sense of weather or surrounding activity. It does some things less well, like conveying the position of a specific object in a space. [...] Sound is great at conveying emotion, so it is both smart and natural to play on this strength.[3]**

1 Ekman 2008, p. 4
2 Ng/Nesbitt 2013, p. 1
3 Papworth 2010, p.4

Sound sometimes seems to take on the role of a support player, aiding to make visuals and gameplay shine without getting the full appreciation. Studies have shown how sound can influence the perceived visual quality, and vice versa:

> **These findings suggest that, when manipulating visual display pixel resolution and auditory display sampling frequency and:**
>
> **• when attending only to the visual modality, a high-quality visual display coupled with a medium-quality auditory display causes an increase in the perception of visual quality relative to established baseline conditions derived from visual-only quality perception evaluations;**
>
> **• when attending only to the visual modality or attending to both auditory and visual modalities, a high-quality visual display coupled with a high-quality auditory display causes an increase in the perception of visual quality relative to established baseline conditions derived from visual-only quality perception evaluations; and**
>
> **• when attending to both auditory and visual modalities, a medium-quality auditory display coupled with a low-quality visual display causes a decrease in the perception of auditory quality relative to established baseline conditions derived from auditory only quality perception evaluations.**[4]

> **More specifically, white noise consistently led to a reduction of visual fidelity perception while classical music and heavy metal music can sometimes lead to an increase in visual fidelity perception.**[5]

Sound has the power to subconsciously alter our perception, but it can also help to change our feelings and actions by using emotional feedback. Hug 2009 talks about *Super Mario Bros'* (1985, Nintendo) different sounds of „power bestowing mushrooms" and „frustrating empty stone blocks". By creating emotional sounds that we trigger in the game world we on one hand feel like we are really interacting with it and on the other hand we are influenced in our behaviour.[6] This seems simple, but there are various complex concepts in this short example that will be explained in this thesis.

4 Storms/Zyda 2000, p. 574

5 Rojas et al. 2013, p. 1581

6 Hug 2009, p. 161

1.2 Functions of Game Sound

The auditory layer has various functions in video games. Since film sound has been researched for a longer time, we can draw from that in some aspects too, although one must consider the media's differences. Ekman 2008 writes about how sound is helping in making things seem real by giving objects in pictures physicality, how sound creates a sense of immediacy by masking the medium (e.g. continuous sounds over cuts) and how it helps provoking sensory pleasure and displeasure. She also talks about game sound and how for example through conformation of actions there is a new function, that being supporting and facilitating gameplay.[7]

A huge difference between hearing and seeing is that we cannot shut our ears like our eyes, the human brain always listens. And not only that:

> **Our hearing is designed and "hard-wired" to constantly scan and analyze the soundscape surrounding us and react rationally to the sounds heard. Most of the time this is done subconsciously and our hearing can therefore be described as, to a large degree, intuitive, emotional, or pre-cognitive. The soundscape reaching our ears demands interpretation and disambiguation in other ways than the visual stimuli reaching our eyes. This need to interpret and disambiguate can be turned into a great asset in computer game design. A game with a well-designed, rich, and varied soundscape will play on the user's intuition and emotions: the game will be immersive and give fun and rewarding gaming experiences.[8]**

This is probably one of the most important functions of sound. It can provide information without us having to shift our primary/visual focus. This is great for complex tasks, be it while performing surgery[9] or just trying to beat a hard video game level.

In first-person shooters sound is essential to localize enemies using two or more sound channels. In this genre sound can also indicate parameters, like low ammunition in weapons,[10] something that is usually displayed in a corner of the screen, so without sounds you would need to shift your focus for those milliseconds that could get you killed by an enemy.

7 Ekman 2008, p. 4

8 Liljedahl 2011, p. 40

9 Sonification, the practise of conveying information through sound, is already widely used in the medical industry. An example can be found at Brazil/Fernström 2011, p. 518

10 Darth 2015, p. 27

Music can have various functions in games too, besides setting or reinforcing a specific mood. A common example is "providing a player with direct feedback in regards to their current status".[11] In many games music changes depending on the situation, so sometimes if an enemy is near, the fight music starts even before the player sees the enemy and will thereby be warned about its presence through listening. Even if the player character saw the enemy we now know through the musical style or through remembering the song that it is not a friendly character who is approaching. This is also applicable for voice over or sound effects. If a wolf-like creature runs towards the character in game, a happy panting or a deep growl informs the player on its presence as well as on the intention of the animal.

For the topic of video games one can generally draw from the research on interaction design. Serafin et al. 2011 lists emotions for interaction design which are helpful to be in mind when designing or analysing sound effects:

- **Valence results from the appraisal of intrinsic pleasantness (a feature of the stimulus) and goal conduciveness (the positive evaluation of a stimulus that helps reaching goals or satisfying needs).**

- **Arousal results from the appraisal of the stimulus' novelty and unexpectedness (when action is needed unexpectedly).**

- **Dominance results from the appraisal of the subject's coping potential.[12]**

11 Ng/Nesbitt 2013, p. 3
12 Serafin et al. 2011, p. 92

Ng & Nesbitt 2013 categorize sound effects as reactionary and preemptive sounds. Preemptive sounds warn the player, for example: "The sound of a grenade landing on the ground would indicate that it's about to explode and the player should retreat away from the blast zone". Reactionary sounds tell us what happened, the sound of a bullet flying next to our head tells us that the enemy shot and missed, while a flesh impact indicates damage taken. Warning signals like heartbeat sounds or the infamous beeping in *The Legend of Zelda* (1986, Nintendo) games further tell the player that his or her health is very low. Auditory cues should be designed specifically for each game because they may differ in the pool of various genres.[13]

In real-time strategy games (RTS), Ng & Nesbitt define warning signals and notifications a bit differently:

> **Notifications provide information about events, which occur in the environment but don't necessarily demand some response from the player. An example of this is when a worker has finished construction of a building; the message 'work complete' is played. The player is notified that the worker is ready to begin a new task. When a research upgrade has finished, the player is also notified that an upgrade for a certain unit or building has finished. In both cases, the player may choose to immediately assign new tasks for the work or new upgrades to enhance their armies.**
>
> **However in certain situations, such as when the player is attacking an enemy base or defending their own base from attacks, the player may choose to not ignore [sic, choose to ignore] the notifications as they tend to be low on their list of priorities. Notifications are also used to warn players if they perform an illegal move. For instance, when a play [sic, player] attempts to build on an unbuildable surface (for example water), train a unit or construct a building that the player can't afford, the player is notified by sound that the move is not possible.**
>
> **Warning signals provide information about threats to the player and usually demand an immediate evaluation and response. An example of this is when the enemy attacks the player's base or their forces, the message 'our base is under attack' is played, immediately letting the player know that they are under attack.[14]**

13 Ng/Nesbitt 2013, p. 4f

14 Ng/Nesbitt 2013, p. 6

Jørgensen 2006 distinguishes five different areas for game sound design: action-oriented, atmospheric, orienting, control-related and identifying functions. She uses the adjective transdiegetic to talk about many sounds in video games, because they often are on one hand diegetic (part of the fictional world) but are communicating at the same time with the player.[15]

• **Action-oriented function:**[16]
These include the above-mentioned status sounds/music, reactionary and preemptive. These sounds trigger actions or are caused by actions. Jørgensen also talks about how some sounds seem to make sense (clicking on a lumber mill and hearing the sound of a saw) but are actually very stylized and unnatural. The objective is to quickly let the player know what was clicked in a not too immersion breaking way.

• **Atmospheric function:**[17]
A variety of background sounds immerse us in the game world by influencing the player's mood as well as making the game feel more realistic. Sounds do not need discrete visual sources; they could just be in a fixed audio track playing in the background. Also, they do not influence actions directly, although contrary to Jørgensen they can make the player act differently in the authors opinion (in a horror atmosphere one might act more cautious than in a light-hearted cartoonish soundscape).

• **Orienting function:**[18]
To better identify where objects are in space, they need to have attached sound sources, especially if they are or could be offscreen. In today's standard stereo sound this enables localization in the horizontal plane, but not in the vertical plane. This would be possible using headphones and imitation of the head-related transfer function (HRTF) or by using 3D sound systems with speakers below and above the player.

• **Control-related function:**[19]
These give players more control over the game, in the same manner as the above-mentioned notifications and warnings in the RTS genre. But not only offscreen sounds are control-related, they are also used if there is too much visual information onscreen to help us make sense of what is happening, like giving information in a tense fight about who is still in the fight and who died.

15 Jørgensen 2006, p. 2
16 Jørgensen 2006, p. 2f
17 Jørgensen 2006, p. 3
18 Jørgensen 2006, p. 3
19 Jørgensen 2006, p. 4

• Identifying function:[20]

In video games it is often imperative to quickly identify objects. In shooters like *Counter-Strike: Global Offensive* weapons sound different, which enables players to evaluate what threat an enemy poses and which could even help players gain an advantage by setting up the engagement environment. This is also emphasized by making weaker weapons sound cheaper or by giving units more character (different voice-lines and sound effects like footsteps).

Identifying function sounds also tell the player how to use the tools the game developers provide us with, as Brown 2017 explains on the example of the Warthog vehicle in the *Halo* (2001, Bungie Studios) series:

> **[...] my favorite vehicle because it's small, it's light, it revs out at a little, you know, a slightly higher pitch. It lets you know: "Oh, this is more nimble, this is not heavy as a tank. This is gonna let me go up and on crags and on rocks." And it's all things that, again, without the player really thinking about it, it lets them know: these are the rules of the world; these are things that are at my disposal, that help me understand the mechanics of the game design itself [...].**[21]

By defining the above-mentioned functions, Jørgensen explains the duality of game sound design: the immersive and emotional aspects of film sound are complemented with usability functions. A dedicated game sound designer should strive to bring all of this into unity to make the gameplay experience sound and play as well as possible.

> **When sound in films breaks this common separation between diegesis and extradiegesis, it is understood as a stylistic, artistic and uncommon way of using sound, but games utilize this functionally to bind together usability and fictional space.**[22]

20 Jørgensen 2006, p. 4

21 Brown 2017, 4:45

22 Jørgensen 2006, p. 5

1.3 Diegetic & Transdiegetic

In film sound theory the phrases diegetic, non-diegetic and extradiegetic are often used to determine the border between just trying to be realistic and designing emotional sound. Film music is (most of the time) clearly non-diegetic, meaning there is not a source for it in the fictional world, it is completely artificially added. Through the participatory layer in video games this can get a bit more complicated:

> **Symbolic sounds have diegetic referents, but the actual sound signals are non-diegetic. These kinds of sounds are very common in computer games. One example is the use of music to accompany the player's actions in the game. These sounds relate to events within the game, while the signals remain non-diegetic. [...]**
>
> **Sometimes a sound signal is diegetic, but it signifies a non-diegetic event. In this kind of relationship, the sound is used to mask a non-diegetic message with a diegetic signal, hence the name. A common example of masking sound is when a player triggers a monster in the game and is notified of this by, for example, a growl or shout from the monster in question. The sound is, essentially, played because the player has entered a certain hot spot. In many games, the reason for the sound is not related to whether the monster actually can see the player, or vice versa, so the signified event is non-diegetic. However, the sound is masking this technicality and notifying the player of the event with a diegetic, in-game growl.[23]**

23 Ekman 2005, p. 3

As we see, the border between diegetic and non-diegetic is blurry in game sound design. But instead of just making things complicated, this is valuable because as shown in the chapter *Functions of Game Sound*, this is what elevates good sound to great game audio. One attempt to structure these terms regarding video games is the IEZA framework (Interface-Effect-Zone-Affect), which distinguishes between four categories by using the parameters activity and setting:

> **The Interface category expresses the activity in the non-diegetic part of the game environment. In many games of today this is sound that is synced with activity in the HUD, either as a response to player activity or as a response to game activity.**
>
> **The Effect category expresses the activity in the diegetic part of the game. Sound is often synced to events in the game world, either triggered by the player or by the game itself. However, activity in the diegetic part of the game can also include sound streams, such as the sound of a continuously burning fire.**
>
> **The Zone category expresses the setting (for example the geographical or topological setting) of the diegetic part of the game environment. In many games of today, Zone is often designed in such a way (using real time adaptation) that it reflects the consequences of game play on a game's world.**
>
> **The Affect category expresses the setting (for example the emotional, social and/or cultural setting) of the non-diegetic part of the game environment. Affect is often designed in such a way (using real time adaptation) that it reflects the emotional status of the game or that it anticipates upcoming events in the game.**[24]

24 Huiberts/Van Tool 2008, p. 6

This seems like a clean and relatively simple framework for defining sound and looking for what a designer should implement in games, but it still cuts between diegetic and non-diegetic. Blurring this line is an essential tool in game sound. As mentioned above, Jørgensen coined the term transdiegetic and defines this in more detail:

> **These three are versions of what I have defined as transdiegetic sounds. The transdiegetic should not be regarded as a clear-cut space that always is easy to identify in computer games, but rather as a property or a function of many diegetic and especially extradiegetic sounds found in computer games. External transdiegetic sounds are sounds that, strictly speaking, must be labelled extradiegetic, but seem to communicate to characters or address features internal to the diegesis. Internal transdiegetic sounds do the opposite: they have diegetic sources, but do not seem to address any other aspect of the game world. Instead, these sounds seem to communicate directly to the player who is situated in real world space. These sounds therefore seem to have some kind of self-reflexivity, where they seem to be conscious about their own fictional existence.**
>
> **Adaptive background music in computer games is typically external transdiegetic sounds. It has no perceived source within the diegesis, but work to inform the player about certain states which s/he may react to on the basis of this sound.[25]**

Four years later in 2011 she tries to distance game sound even more from the diegetic/non-diegetic border by defining new terms emphasizing on the information value instead of the diegetic aspect. These five terms bridge the gap between diegetic and non-diegetic, they are "a continuum that integrates user interface elements into the game-world to a lesser or greater degree"[26]: metaphorical, overlay, integrated, emphasized and iconic interface.

There is more to learn about diegetic, transdiegetic and interface sounds, but this is enough for the purpose of this thesis. Even though, the author would recommend all aspiring game sound designers to read Jørgensen's and Ekman's research on their own.

25 Jørgensen 2007, p. 9

26 Jørgensen 2011, p. 92

1.4 Research on Difficulty directly through Game Sound

There is very little research about how sound designers influence the difficulty of video games. A 2016 study found that:

> **A game with intense music can feel up to 40% more difficult compared to the same game with more relaxed music. [...] The amount of mistakes made increases about 50% between the relaxed music and intense music version.[27]**

So, music can have an influence on difficulty perception as well as difficulty itself. But what about sound effects and soundscapes? In another study the effect of sound design in slot machines was researched, to be specific how often people think they won and how often they actually won. Spins where players would only win a fraction of the bet money were also counted as a loss, but celebrated with light and sounds. On average, players overestimated their wins 9% more on slot machines with sounds on (33 estimated wins at sound off, 36 estimated wins at sound on, 28 actual wins.[28]

Voiceover (VO) & player-triggered sounds have an influence on difficulty perception too:

> **[...] such as friendly AI waring [sic, warning] the player that a grenade is being used somewhere in the players [sic] proximity. [..] loud weapons might influence the perceived difficulty, possibly by making it harder to hear the various voice sounds coming from the players [sic] allies and enemies that are suggested to be more of value to the player.[29]**

27 Wiechers 2016

28 Dixon et al., p. 916

29 Darth 2015, p. 27

One practice to make games easier, that is also used in the visual domain, is *telegraphing* attacks. In a similar fashion like the above-mentioned masking signal, enemies can tell the player that they are going to attack and which attack they are going to use if the player analysed previous patterns of the enemy:

> **But one thing you'll come to realize is that these attacks are ultimately predictable. [...] And this means that a big part of a boss battle is just learning about the boss, figuring and then memorizing its patterns. And these mini self-taught epiphanies stack up to the point where you can out-predict and outsmart the boss at every turn, which feels pretty good, after having taken quit the beating. [...] Luckily, the enemy will often announce what attack it's going to do next, either through animation [...] or through sound, like these walking flames [in Cuphead's Grim Matchstick stage] who yell out before they jump at you. [...] This is called "telegraphing", and the duration of an enemy's telegraphing will affect the difficulty of the fight.[30]**

As we see there is not a lot research dedicated to difficulty change itself, so this information must be found in other research fields.

30 Brown 2017

1.5 Multimodality
1.5.1 Combining auditory & visual Stimuli

Multimodality is the connection of two or more senses to create one phenomenon.[31] For example, the clicking sounds of the old iPod when turning the selection wheel enforce a haptic feedback. The multimodality here is that the tactile and auditory senses are connected to create a nicer feeling and better usability.[32]

"The concept of synchresis – that merging of image and sound [...]"[33] is a main focus for film sound designers, but there are game specific aspects, too. One reason for multimodality to be a topic in this thesis is because it can influence task performance. A 2015 study showed that "white noise led to a decrease in performance while the sounds that were related to the visual scene improved performance (when the sound was not distracting) across all levels of visual realism [...]".[34] The addition of an ambience that suits the visual environment could have an effect because of better immersion and thereby focus on the simulation instead of the real world surrounding the user.

The effectiveness of learning can also be increased by multimodal cues, as shown by testing motion-detection tasks. They found that even when the test is repeated without sound, a person that learned it with sound is more likely to perform better at the task:

> **Although it is perhaps to be expected that practice on an audio-visual task should increase audio-visual interactions, it is intriguing that audio-visual practice contributes to enhanced performance on unimodal trials.[35]**

An experiment on a rhythm task for infants using multimodal cues also proved that synchronous multimodal cues help learning. In the same paper it was proposed that:

> **[...] synchrony between audible stimulation and visible stimulation elicits greater attention and processing than does their simple co-occurrence. [...]**

31 "Where media are concerned, multimodality is the use of several modes (media) to create a single artefact." https://en.wikipedia.org/wiki/Multimodality (last consulted March 13, 2019)

32 Serafin et al. 2011, p. 95f

33 Collins 2013, p. 26

34 Cowan et al. 2015, p. 1213

35 Seitz et al. 2006, p. 1425

> **Specifically, intersensory redundancy recruits infant attention, causing the redundant information to become "foreground" and other information to become "background." This fosters perceptual differentiation, learning, and memory for redundant, amodal properties before other stimulus properties. As a result, intersensory redundancy ensures coordinated perception of unitary events. Further, when the same amodal property is presented unimodally, it will not recruit comparable levels of attention and thus will not be perceived, learned, or remembered as well. Rather, unimodal stimulation may at first foster more attention to properties that are modality specific, such as color and form for the visual modality and pitch and timbre for the auditory modality.[36]**

Results from Kristal-Ern 2017 are consistent with the study above. He also ties this to games and difficulty:

> **[...] for example multimodal cues could be used to decrease the difficulty for players that are stuck on a level. To keep the pace of the game narrative, the game designer could choose to implement cues as clues that makes the target or goal in the game more salient so that the players that become stuck find it easier. Another area of usage could be to remove amodal information from events that are distracting the players, indirectly making the target more salient.**
>
> **The multimodal cue in the study was not directly tied to visual search target as in the visual search target did not convey any properties redundantly conveyed through another modality. Despite this, the multimodal cue seems to have helped subjects in the synchronous group. This is interpreted as evidence for that multimodal cues can be used to guide players, even if the multimodal cue is not what the players has been explicitly asked to look for in a game task.[37]**

36 Bahrick/Lickliter 2000, p. 13f

37 Kristal-Ern 2017, p. 8 & 28

In multimodality the information flow for each sense can be called a display (e.g. auditory display, visual display, etc.). The terms to describe these displays in depending on their relation to another and their relation to game design can be found in Ng/Nesbitt 2013:

> **Complementary displays attempt to provide useful, but different, information on each sensory channel. Compared to a single-sensory display a complementary display should allow the user to perform better. If the user actually performs worse with the multi-sensory display then the display is described as conflicting. This is presumably due to some conflicting information that the user receives on the different sensory channels. With redundant displays the same information is displayed to each sense. This may serve the purpose of increasing the user's confidence or reducing the perceived workload. Although users may report a reduction in workload or an increase in confidence with redundant displays, the performance of the user with the multi-sensory display is the same as with single-sensory display.**
>
> **All three types of display may be relevant to the design of computer games. For example, complementary displays may improve user performance and they could also act as a reward for players who reach higher levels. In most domains, it would be abnormal for the designer to provide the user with a conflicting display, but a conflicting display could be useful if game designers wished to increase the level of difficulty for the player, or to punish the game player, or perhaps make a task more taxing by increasing the player's stress level.**
>
> **On the other hand, redundant displays may give players greater confidence in some situations and this may act as a reward or assist the player to learn skills required in higher levels of the game.[38]**

We have seen that multimodality enforces learning and its effects are stronger the more synchronous it is. Asynchronous cues on the other hand can be very confusing,[39] especially as a delay with the modalities action and sound, which bring us to our next topic.

38 Ng/Nesbitt 2013, p. 2

39 Collins 2013, p. 33

1.5.2 Multimodality of Sound and Action

As this thesis is about game sound design the subject of interaction is essential. The connection of sound and action is called *kinaesonic synchresis.*[40] When a player pushes a button or a key in a game to perform an action, sound helps to immerse the player in that action. As Collins describes:

> **Even in cases where we do not have kinaesonic congruence with sound, however, we still receive some of that sonic response to our action. Although our physical body did not kinaesonically create that sound, the mimetic hypothesis (and mirror neuronal research) suggests that we may still feel that it did through our own subsequent mental/corporeal imitation. If we hear sound in terms of our own embodied experience of that sound, then when we hear those action sounds in games, even though we did not kinaesonically create them, we hear them as if we did. In other words, we have a direct, embodied interaction with the sounds that we evoke and hear in games, and coupled with our physical and/or kinaesonic-congruent action, these sounds (and thus the game character) can become an extension of the self. [...] The embodied cognitive connection to sound is vital in our extension of our self, as an extension of our body-intechnology. In this sense, it is not the controller through which our body is extended, it is through the game character. The character is the tool through which we experience the virtual world - through which we bump into walls, get shot, dig holes and talk to other characters.**[41]

The combination of interaction and sound can also invoke a substitution for other senses like touch/haptic, as seen in the iPod example[42] from above and in the lockpicking game mechanic in *Thief 3: Deadly Shadows* (2004, Ion Storm Austin).[43] The kinaesonic synchresis works best if we recognize the sounds, because one can better immerse oneself if one has generated the sounds before, which is called kinaesthetic sympathy (Collins 2013, p. 60f). Also, the repetition of the same sound is beneficial for fast learning and identification, but it can feel unnatural and even annoying. Therefore, there should be a pool of different sounds, alteration of parameters (for example pitch) or even segmentation of sounds with dependence of the characters speed in step sounds.[44]

---

40 Collins 2013, p. 32

41 Collins 2011, p. 4f

42 Serafin et al. 2011, p. 95f

43 “Through sound the player „feels“ how the lock is behaving, if it resists or slowly succumbs” [translated by the author]: in Hug 2009, p.162f

44 (Collins 2013, p. 33f).

So which multimodality is more important, sound and picture or sound and action? Of course, it would be best to have them all synchronous, but in games the synchrony of sound and action is a bit more important than sound and vision.[45] In a 2015 Game Developer's Conference Talk Jonaas Turner elaborates on this in a hands-on way:

> **[11:30] Make the beginnings shorter of your sounds. Think about the player pushing a button [...] It's instantanish for the button press [...] You want the player to press a button and be like: "Hey, I did that, I press the button and I jump in the game, instead of I press the button and that creature jumps in the game."**
>
> **[16:00] If you have a small punch sound, even if it's like barely, you can't even hear it proper, it's just like a feeling like a small base *pft* over there just like that and you have that in front [of your action sounds, could be the same at all player's action-sounds for consistency and feel] [...] that small thing makes it feel like I did something.**
>
> **[18:30] What if I put ducking on the weapons [to lower background noises when a weapon is fired]? [...] The shooting became more prominent, it was more clear, it feels more, there's more power. I'm doing something that affects the world. [...] And what if you would apply filtering to that as well? [...] Right now it's like overly accented, but if you do that slightly it will bring so much power and feelings for the actions. You could apply that to anything, like UI [User Interface, author's note] or jumping.[46]**

These are great examples of how to adapt the whole theory to actual craftsmanship of game sound design, which is what will be examined in the next chapter.

45 Collins 2013, p. 60
46 Turner 2015

1.6 Execution of Game Sound Design

A general necessity for informative game sound is that "the meaning should be clear instantly (for example the distinction friendly vs. hostile and safe vs. dangerous) or at least easily learnable".[47] In the case of warning sounds Fricke 2009 further elaborates that they should

> **[…] be quick, have a high signal-to-noise ratio, should not be too loud to not be annoying, be clear and distinguishable from other sounds, have a low false-alarm-rate and allow adjustments in the detection sensitivity.[48]**

She talks about this in the context of warning and alarm sounds in automobiles, but it is as true in game sound design.

One might think: "Why not just use speech as an information carrier?" This is of course a valid option, but on one hand speech samples are often not as quick, elegant and diegetic as sound effects and on the other hand their repetition gets subjectively more annoying. Liljedahl notes:

> **A danger with speech is the risk of wearing out often-repeated phrases. It is therefore useful to give the players the option to skip, for example, instructions when they are no longer needed. […] One of the advantages with sound effects and music is that they are not limited by language, but are more universal. This can of course be used in many ways. […] By carefully selecting the metaphor aspect of a game's design, tremendous opportunities to create sound effects for feedback and information can be opened up. By placing the game in an environment (metaphor) that the players are likely to have some kind of relation to, the designer can choose sounds for feedback and information that are natural in that environment. Using natural sounds that the players can immediately relate to can greatly enhance the gameplay aspect of the same game as well as create the sought-after sense of presence and immersion.[49]**

47 Hug 2009, p. 164, translated by the author.

48 Fricke 2009, p. 53, translated by the author

49 Liljedahl 2011, p. 20

Realism is an interesting topic in the execution of game sound design and sound design in general. Sound does not need to be realistic, but it must fit the fictional world to be believable. Also, one must balance between the informative part, emotional alteration and authenticity.

> **The narrative realism of a sound is thus not in faithful reproduction of sound sources, nor of their environments. The apparent realism of a sound in the context of narrative is defined by how representative a sound is of a certain event. Sounds that are highly representative have good narrative fit. High narrative fit supports empathetic emotion. [...]**
>
> **In games, the task of many sounds is primarily to provide feedback about actions. Hence, narrative fit is often sacrificed for utility. To the extent that auditory cues are used to guide actions, they are treated with utmost respect for legibility. For example, even in the case of instructions with diegetic source (a non-player character, voice mail, etc.) it is common that auditory instructions remain heard even if the character runs away from their diegetic source.**
>
> **An interesting avenue for sound design in games is to shift focus from music to the emotional impacts of Foley and sound effects. A possible alternative for emotionality in games is in environmental sounds, which is already used in many games, where ambient sounds are beautifully merged with musically suggestive elements and event sounds into a sonic landscape in the spirit of musique concrète. However, for this approach to be systematically explored, there is need for better understanding of how everyday sounds influence emotions. In these investigations, theories of unconscious emotion may prove especially informative.[50]**

50 Ekman 2008, p. 7

In an essay Zack Quarles (audio director and supervising sound designer) talks about how they made their game *Killer Instinct* (2013, Microsoft Studios) more accessible to the visually impaired. This is accomplished through the above-mentioned identifiable (or even iconic) audio content, dynamic music and dynamic mixing (referring to *Wwise*'s HDR audio system). He states:

> **Ambient loops and sweetener layers that didn't provide strong player feedback were backed off a bit. This still allowed for a very layered soundscape for the game but we really wanted to focus on the individual characters to make sure that the player could tell where they were and what they were doing at any given time. [...]**
>
> **We split the screen into three major sections during gameplay: Left, Center, and Right. When a player passes through one of these sections we do an aggressive hard pan on their location to make sure that the player can track where they are in the battle arena at any given time from audio alone.**
>
> **This might seem drastic when you first start playing the game, but once you sit down and hear it in action, it starts to feel natural. Particularly over headphones. Likewise, when we are in the character selection screen, we hard pan Player 1 UI sounds in the left channel and hard pan Player 2 UI sounds in the right. It gives a very clean and distinct method of letting the player know who is selecting what.**

> **Clean Up Your Audio Mix – It's always tempting to make sure that everything is a delicious treat for the ears, but sometimes having too many elements working at the same time can cause confusion and will muddy up the waters. What's the most important information that you need the player to hear? Find the core pillar of your game and mix around that. Don't try to compete with it.[51]**

These insights are especially interesting because on one hand they are from an actual game and not just theory and on the other hand thy validate some of the above-mentioned theory.

51 Quarlse 2019

1.7 Ideas for Game Sound Design

We have looked at many ideas that are implemented in many games, some maybe unconsciously. But in the thesis research some unique ones came up too. One example that has been talked about quite a bit in the game audio community is *Overwatch* (2016, Blizzard Entertainment). Using the above mentioned HDR audio system the sound effects priority for dynamic mixing is calculated through a:

> **Threat Level. In Overwatch each enemy's volume is based on how much threat they pose to you at any given moment. This means that the footsteps of the guy aiming a shotgun at the back of your head are louder than the gunfire pointed in another direction.[52]**

Giving players auditory foreknowledge as the mentioned *telegraphing* could also be used in time-based puzzle games:

> **[About Tetris] but how would the gameplay change if each unique puzzle piece had a unique sound cue, making the next puzzle piece identifiable without visual attention? The game would likely be less challenging as players would be able to combine sensory inputs to complete the task, something which comes very naturally to humans. In other words, we probably won't have to look at which pieces that comes next after a while as this information is already being acquired through our hearing. As Tetris is designed to be a challenging game the reduced effort for the user might not be desired.[53]**

52 McGee 2017

53 Kristal-Ern 2017, p. 4f

Most of this thesis is more focused on sound effects than music, but there are very interesting ideas in the field of game music. Adaptive soundtracks have been talked about in the context of enemies approaching, but there are other implementations as well:

> **A more recent example is Portal 2. Not only does it apply nifty effects to the music when you jump on the blue gel and run on the orange gel, but in some test chambers the different puzzle bits sing out as you activate them. Watch how redirecting these lasers produces a new soundtrack. It's not just a nice auditory gimmick but it tells you that you're on the right path, and if you mess up your current progress you'll know because you hear it.**[54]

In this example music gives feedback on puzzle progression. On a related topic one study indicates that musical intervals could be used to guide players:

> **The results from the test have shown that, when given the choice between consonant and dissonant intervals, a majority of the participants chose to move towards the consonant interval, with one room being an exception. [...]**
>
> **The individual intervals which indicated to be the most effective in guiding players were the fourth, fifth, octave, and minor second. Apart from the minor second, which is dissonant, the effective intervals were consonant. [...]**
>
> **Utilizing musical consonance and dissonance in sound design is just a small part of the many aspects that could potentially be of use when trying to guide a player with sound. Aspects of sound such as, for example, volume intensity, frequency range, directionality and spatialization could also be of use when designing sound for aiding the player in a game environment. sual presentation and all activity occurring in the environment.**[55]

54 Brown 2014

55 Östlund 2015, p. 26 & 28

Another psychological way of guiding players is to shift their general focus through music and mood:

> **Music can have a profound effect on the mood of the listener, and a change in mood can affect not only what is visually noticed, but also how much visual detail can be perceived. At the Affect and Cognition Laboratory at the University of Toronto, researchers conducted an experiment to determine how mood influences visual perception (Schmitz, De Rosa, and Anderson 2009). [...]**
>
> **The researchers concluded that a positive mood increases a person's field of view, allowing them a wider range of vision than those experiencing a negative mood, whose field of view would be comparably narrower.**
>
> **The results of this study have complex implications for the use of music within the video game framework.**
>
> **[...] sad music could have the power to help direct the player's attention to singular event or objects to exclusion of all else that may be happening onscreen. Conversely, happy music may have the power to expand the player's field of vision to encompass more of the full picture, allowing for a greater awareness of the entire visual presentation and all activity occurring in the environment.[56]**

56 Phillips 2014, p. 46

1.8 Audio Implementation

The nonlinearity of video games demands for a system that plays the sounds when needed. So, for sound design of games, a middle step between sound creation and audio directing/mastering is necessary:

> **Audio implementation, or integration is the process of integrating audio assets and/or systems into the game engine, establishing sonic behaviours, audio mixing, and revising the content according to the emergent demands.[57]**

Dwight Okahara (Audio Lead at Insomniac Games) notes that 60% of the game audio work is implementation.[58] The audio implementation process consists of many steps. One big example is the randomization of sounds. If the audio clip of a shot always was the same, it could get annoying very fast. Therefore, one could pick a random clip from a pool of clips (plus making sure that it is not the same clip as last time), maybe even random different layers that make up one sound and/or change the pitch of it.

In game music it is possible to make the soundtrack nearly never loop itself through randomization of the parts that can be played and, like written before, adapt seamlessly to new game states (for example from exploration to fight music). Other examples of audio implementation include:

> **Middleware lets designers link sounds to game objects, including animations, scripted events and areas. It also includes features such as parameter controlled DSP effects, dedicated prototyping environments,, sound prioritization and real-time parameter controls. Real time mixing consists of techniques like mixer snapshots (group of parameter values that can be applied instantaneaously in a single command), which take advantage of the tool's own bus hierarchy.[59]**

57 Uzer 2016, p. 15

58 Waves 2017

59 Pires et al. 2014, p. 3

Many game engines have their own audio integration. *Unity*, the game engine that was used for the workpiece of this thesis, also has some implementation tools, even though many things like the randomization of clips must be coded. Another possibility is using audio middleware that has many game audio utilities ready for sound designers:

> **Its last instalment, FMOD Studio, is an authoring tool and runtime audio engine that allows audio content creation for games, with an interface that resembles more a professional Digital Audio Workstation than existing game audio tools. Similarly, Wwise's approach tries to ease the work of both sound designers and audio programmers by redefining the production workflow for audio an improving pipeline efficiency.[60]**

To summarize, the game sound cycle at Insomniac Games looks like this:

> **To create a sound effect, we'll start with the source sounds we want to work with – they can come from personal SFX libraries, commercial SFX libraries, or field recordings – and we'll design the sound using multiple plugins. Once we've created a sound effect, we'll bounce that element out as a .WAV file and implement that into our sound engine. We use Wwise in our games, and once a sound is implemented, the real-timeness of the game takes over. When that sound is played back in the game, it hits any appropriate reverb, EQ or occlusion/obstruction filters that we have created, so that the sound plays back as expected in the environment whenever the player is triggering it.[61]**

60 Pires et al. 2014, p. 3

61 Waves 2017

1.9 Research Summary

Through the research on this topic several guidelines and ideas for the game sound design and the actual game design were collected for the workpiece of this thesis. Parallel (and unconscious) listening is a key factor for informational sounds. Panning works for enemy localization (offscreen and onscreen).

Sounds should be instantly recognizable (meaning and explicit sound) but should have alteration of audio parameters to not sound too artificial or get annoying. Reactionary and pre-emptive sounds as well as notifications and warning signal help to inform the player while not being too artificial when implemented as masking signals.

Game states (e.g. health or danger) can be made clear by sounds or music.

Adaptive music can also accompany actions or help with puzzles. An ambience that supports the setting can boost performance, but it should not blur the intake of informative sound. To ensure this, dynamic mixing can be implemented, maybe even in a more complex form like threat level mixing.

Redundant displays and multimodal synchronous cues make games easier while unimodal cues, conflicting displays and asynchronicity make them harder. The synchronicity of action and sound is more important than the synchronicity of picture and sound. To strengthen the feeling of influence action sounds should start fast, could have a short impact at the beginning and duck background noises.

1.10 History of Arcade Shoot 'em Ups

The game designed for this master thesis has its roots in classic arcade shooter games like *Galaga* (1981, Namco). Video games and more directly arcade games developed from "the amusement industry with pinball machines and board games on the one hand, and the rapid development of computer technology on the other."[62]

The earliest real video games are *Tennis for Two* (1958, William Higinbotham) and *Spacewar!* (1962, developed at the Massachusetts Institute of Technology). They both had no sound yet.[63] The first commercial hit video game did have sound, the beeps from *Pong* (1972, Atari). Atari's co-founder Nolan Bushnell "can be considered the founding father of the arcade video game".[64]
The role of a forefather of the shoot 'em up genre can be credited to *Tank* (1974, Atari), with many games of that genre to be produced in the years to come.[65] Just four years later the next big hit after *Pong* appeared and showed that not only Americans could make video games:

> **On June 5, 1978, Taito, a Japanese company established by Belorussian immigrant Michael Kogan, released Space Invaders. [...]**
>
> **Already in 1981, the game had swallowed more than four billion coins, an average of one game per inhabitant of the earth. The success of Space Invaders can, for a large part be explained by the nature of the game itself. [...]**
>
> **Referring to the body of science fiction ideas that existed, a simple story was told of a swarm of aliens that came to conquer the earth and it was up to the player to stop them from getting a stronghold on this planet. [...]**
>
> **There was no upper limit to the number of points that a player could score, and as a result players could keep on playing indefinitely, always finding a new challenge in having to do better than the time before. Third, Space Invaders used sound in a functional way: the rhythmical bass-based soundtrack, which sped up with the rhythm of the game, was an integral part of the game experience.[66]**

62 Raessens/Goldstein 2005, p. 23
63 Collins 2008, p. 8
64 Raessens/Goldstein 2005, p. 25
65 Raessens/Goldstein 2005, p. 27
66 Raessens/Goldstein 2005, p. 28f

> **In terms of nondiegetic sound Space Invaders (Midway, 1978) set an important precedent for continuous music, with a descending four-tone loop of marching alien feet that sped up as the game progressed. Arguably, Space Invaders and Asteroids (Atari, 1979, with a two-note "melody") represent the first examples of continuous music in games, depending on how one defines music.[67]**

The controls for commercial hits were relatively simple in the beginning because people needed to get used to the whole concept of games. As time passed, players were ready for physics-related controls as seen in *Asteroids* (1979, Cinematronics)[68] and more complex enemy design:

> **Galaxian [Namco, 1979] was more difficult than Space Invaders. Rather than marching in straight lines across the screen, the alien ships in Galaxian swooped down in changing formations. Though its profits were only a fraction of the money brought in by Space Invaders, Galaxian was one of the most successful games of its time."[69]**

Opening new design possibilities through a new mindset on the shooting genre was brought by the game *Defender* (1981, Williams).

> **In contrast with Space Invaders and its many relatives, the action did not take place vertically, but horizontally. […] Where Asteroids already suggested that offscreen action was possible (by having the spaceship disappear and reappear on opposite ends of the screen), Defender was the first game to actually use it. […] games moved away from the principle of "the faster you move and shoot, the more points you score." Strategic thinking became more and more important.[70]**

There are not many classic shoot 'em up games anymore, but there are still new entries that built on their success. To name just a few derivatives there is the "roguelike" shoot 'em up *Enter the Gungeon* (2016, Dodge Roll), the "bullet hell" music game *Soundodger+* (2013, Studio Bean) and the 1930s cartoon-inspired shoot 'em up *Cuphead* (2017, Studio MDHR).

---

67 Collins 2008, p. 12

68 Kent 2001, p. 132

69 Kent 2001, p. 137

70 Raessens/Goldstein 2005, p. 31

2. Game Presentation

Fig. 1: Game Cover Graphic

2.1 SHMUP

For this thesis the game "Space Hangar M: Ultimate Protocol" was created. The name is a reference to the abbreviation of the genre (shoot 'em up). It was developed in *Unity* because of its wide usage among independent developers and its free availability for non-profit video games. An audio middleware was not used because the game is not that complex and a more in-depth look into audio programming is beneficial for the thesis question.

To best demonstrate the possibilities to change difficulty through sound design the genre of classic 2D space shooters was chosen. The setting allows for diegetic HUD sounds and the genre lets the game designer easily throw in a variety of game mechanics that can be enhanced through sound effects.

The base layer for programming and game design was a YouTube tutorial,[71] but new and different mechanics as well as enemies were added. The pixel graphics were made by Christina Cossee (except for the crosshair) and audio samples and music from the internet were used,[72] everything else was made by the author.

As in most science fiction (sci-fi) entertainment products the physics in this game are not bound to realism, most importantly, the player can hear sounds outside of the spaceship, although it would normally be quiet due to the vacuum.

---

71 PUREHEART: Creating a Classic 2D Shoot 'Em Up https://www.youtube.com/watch?v=lwXwtvDBqbo&list=PLsgaKssSvST5hgN_71aQnyDxMY3wAlPLz (last consulted August 25, 2019)
72 See Sound Attributions

2.2 Game Mechanics
2.2.1 UI

UI elements are kept very simple with just using text and a few buttons in both menus. Buttons have three different background opacities, low when just there, middle when the player hovers over it and highlighted when pressed. The cursor, which is used for menu inputs as well as the shooting in-game, is replaced with a pixel-style crosshair:

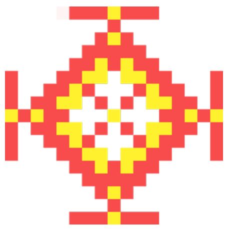


Fig. 2: Pixel Crosshair

a) Main Menu

The main menu greets the player with the games title, creator and the in-game space background. In this game sound has an important role in aiding the player. If the player chooses to play this game on bad laptop speakers, the audio quality will be insufficient and information about the horizontal position of enemies would mostly be lost. This would be a constraint, like playing with a mousepad instead of a normal mouse. For this reason, even before the controls the player is advised to use headphones and a mouse for best performance. Controls are explained through simple text. The player has the option to start or quit the game. When pressing "Sound Credits" the attributions of used sounds will appear (the same as in Sound Credits).

Fig. 3: Main Menu Screenshot

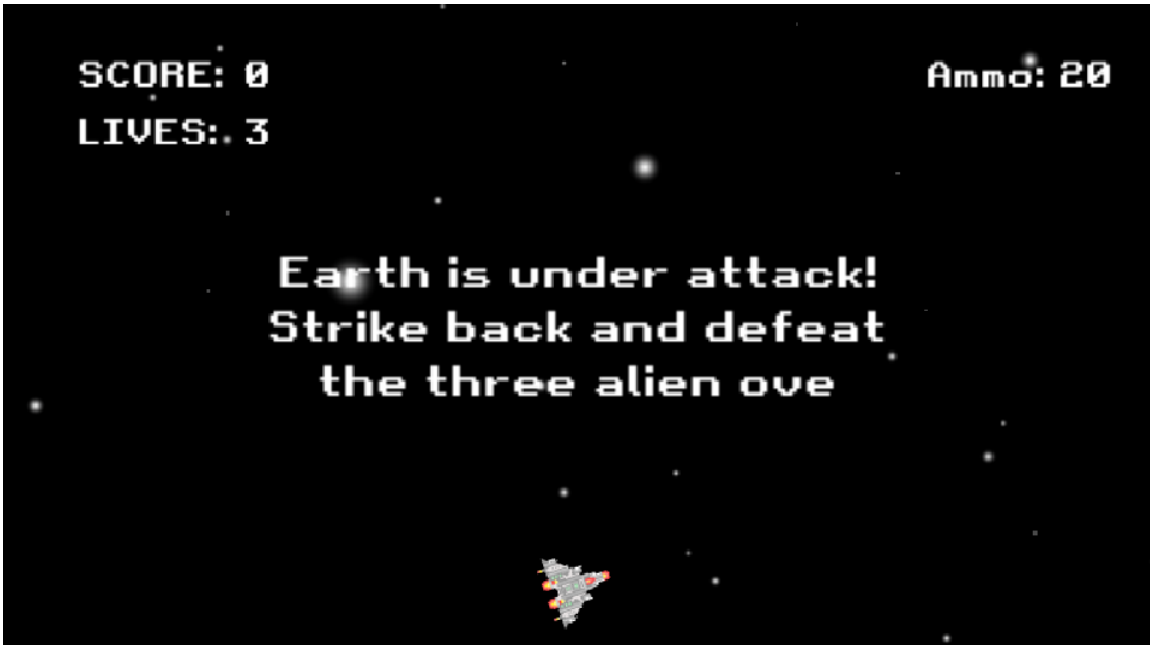


Fig. 4: In-Game Screenshot

b) In-Game

The in-game UI features the score, the player's lives and ammunition. It is just shown with text since the focus lies on sound design. The current health is intentionally not shown to leave this to sound design. When starting the game, the player gets a short incoming transmission ordering him or her to defeat the aliens that are attacking earth. The text appears in a typewriter-style with the time between characters appearing being randomized.

c) Pause Menu

In the menu the paused gameplay scene and its UI is still shown, but a few things are added. It shows the "Pause" title, explains the controls again and displays the current high-score. There are also three buttons for resuming, restarting and quitting the game.

d) Game Over

The game over screen reads "Game Over" or "You Won!", depending on if the player lost all lives or finished the game. It also shows the score and high-score. If a new highscore is reached, the number will blink. The player can start a new game or quit the application through two buttons.

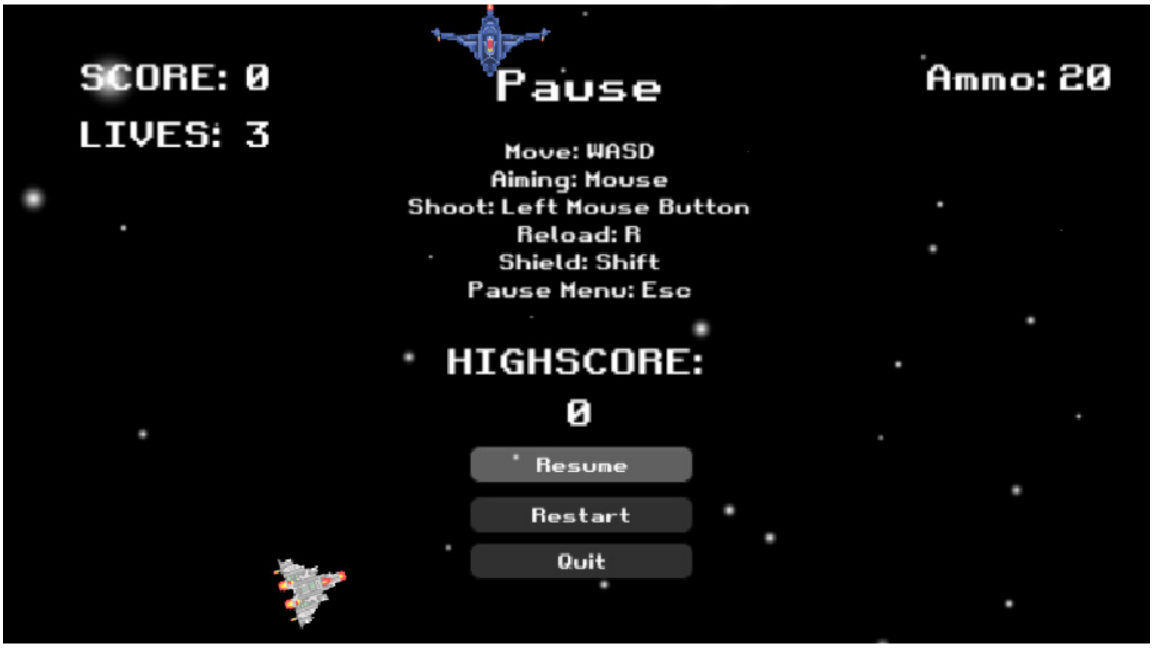


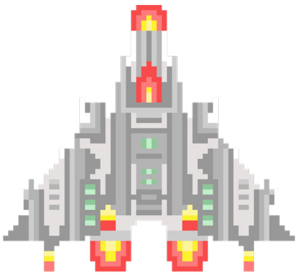
Fig. 5: Pause Menu Screenshot

Fig. 6: Game Over Screenshot

2.2.2 Player Controls

The orientation of the ship is controlled with the mouse so the player can make precise adjustments when aiming at enemies. Movement is possible on the visible screen in 2D from a bird's eye perspective. Using background effects in form of two particle systems, it looks as if the player is travelling upwards through space even if the ship is not moving. The use of acceleration and drag when manoeuvring is supposed to make movement feel more realistic and be more challenging. The ships movement is controlled by classic WASD keyboard controls.

The player shoots by clicking the left mouse button. Holding it down results in continuous shots with short delays in between, but faster shooting is possible through repeatedly pressing the mouse button. The player has only twenty shots at a time which are reloaded through clicking the R key. This takes a short amount of time and shooting is not possible while reloading, but the more ammunition the player has left the shorter the reload time. With the shift key the player can create a shield around the ship, which negates damage from normal shots, but neither shooting nor reloading is allowed in this state and ammunition is consumed through using it.



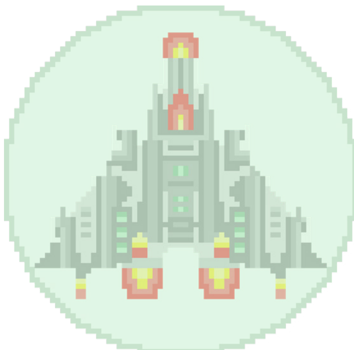


Fig. 7: Player normal (left) and shielded (right)

2.2.3 Enemies

There are five enemy types in the game. The "Kamikaze" ship just goes straight down and does damage to the player if hit but also gets destroyed. There are two enemies that are slower than the Kamikaze ship but shoot straight down or up.

While the "Down" enemy is not that big of a threat, the "Up" ship should be focused because it will keep shooting for a short time even offscreen, which makes it seem silly at first but is revealed as dangerous if not dealt with. Up and down enemies have the same appearance, which means the player can only distinguish them by sound or by their shooting direction.

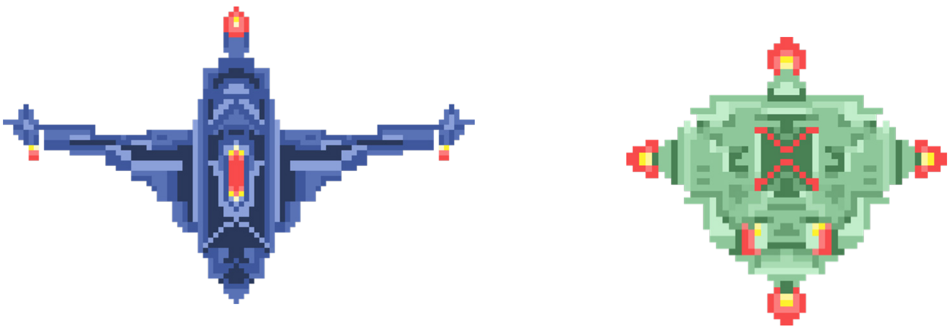


Fig. 8: Enemies: Kamikaze (left) and Up/Down (right)

The most difficult small enemy is the "Charger". This enemy will stop moving downwards when arriving at a randomized set altitude and instead starts sliding left and right. If it stops moving it will shoot in the direction of the player, who must either dodge the shot through fast movement or by using the shield. This enemy is only vulnerable when charging up its shoot, which means the player must time actions between shooting and dodging/shielding.

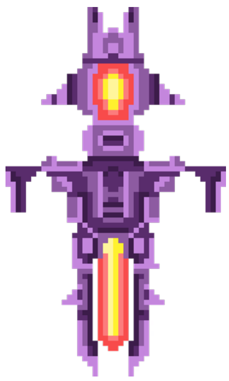


Fig. 9: Charger Enemy

The last enemy is the "Boss", a giant ship with three phases. In the first phase it volleys three times three aimed and angled shots from its left and right turret. The player must shoot at the cannons to destroy them. When one is destroyed, the other one's bullets will be faster. If both are destroyed the ship will channel a giant laser from its tip at a delayed player position. Before doing this, it will show a short aiming phase and, like the previously shown Charger enemy, the turret is only vulnerable while charging and shooting.

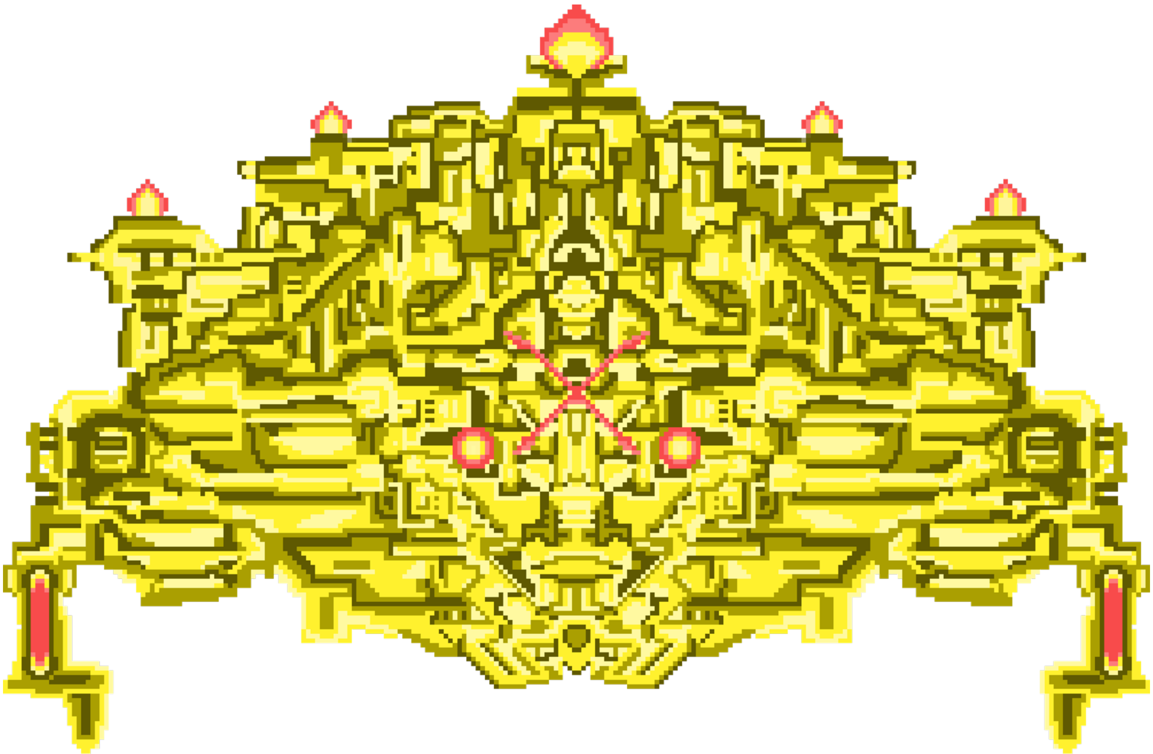


Fig. 10: Boss Enemy

2.2.4 Difficulty Curve

The enemy types are introduced in an order from easy to hard and the Boss is alone at its first appearance. After the player beats the Boss the spawn rate for enemies gets faster and easier enemies are gradually not spawning anymore, leaving the player with more and harder enemies. When the Boss appears the second time it is accompanied by Kamikaze enemies, its turrets have doubled health and the position delay of the giant laser is lowered.

On meeting it for the third time the player also has to deal with "Chargers", turret health is tripled and the position delay is nearly gone. Since the Charger enemy stays on-screen until it is beaten, its current instances are limited, but rising in numbers with the difficulty curve. The game ends after this third time beating the Boss.

Fig. 11: Shots from top to bottom: Boss Triplet, Up/Down Shot, Player/Charger Shot, Boss Laser

2.2.5 Health

The player starts with three lives and three health points. A normal shot does one damage which can be negated by using the shield. Crashing into an enemy as well as the Boss laser conflict two points of damage which can be reduced to one by using the shield. After being hit the player has a short time of invincibility. When killing an enemy there is a chance of a health item appearing, with the probability depending on the difficulty of the enemy. The Boss enemy will always drop the item. It will be accelerated downwards, so the player must act quickly to touch it. If it is collected the reward is an extra health point as well as an extra life. If the player has lives left when dying, he or she will be respawned at the same point of the difficulty curve. If the player runs out of lives he or she must start from the beginning.

If an enemy or the player ship is damaged it will blink using (most of the time) inverted colours to show that the shot landed. If the player shield is active the shield's opacity will be changed from 100 to 255, like it is completely guarding the ship. If an enemy has an active shield the player does not see it, it just blinks up when damage is negated.

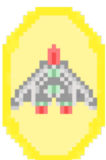


Fig. 12: Health Item

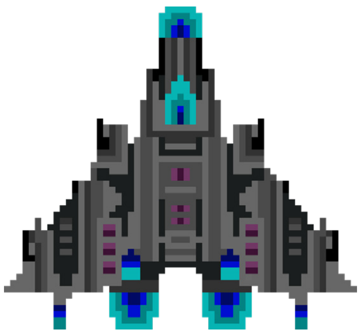


Fig. 13: Player Ship blinking for received Damage

3. Sound Design & Implementation
3.1 Style

Classic arcade game sound design was restricted by what hardware was available and affordable at the time. Some iconic sounds like in Pong were just kept simple out of lack of electronic parts and sound design knowledge.[73] It is possible that technical or aesthetic boundaries evoke great design. Nevertheless, the sound design for this game is a modern approach, like consumers today are used to in other science fiction works. This was chosen so the sound is informative as well as recognizable and enjoyable, while not being just beeps for the average user.

The player is supposed to feel like the underdog, which is enforced by the player ship's sounds, which are most of the time purposely made more low-fi[74] than enemy sounds. The practise of Used Future from Star Wars is also used, where on one hand it sounds like futuristic technology, but also a bit rickety. Enemies on the other hand should sound more advanced and cleaner.

73 Kent 2001, p. 42

74 Low fidelity, in this thesis most of the time accomplished by applying a bit-crusher.

3.2 General Procedure

a) Sound Design

A list for needed game sounds was made and for each of them ideas for audio layers were collected. For example, the lasers could be composed of Ben Burtt (sound designer for Star Wars) style wire recordings, thunderclaps, explosions, electronic engines, synthesizers and pulling tape from different surfaces. Samples for the sound design were acquired from the author's personal sound library, recording new samples with a Zoom H5 and freesound.org.[75] Since the Sci-Fi sound creation would include heavy use of editing and plugins the original sound quality form the H5 and freesound is not as important as in creating naturalistic soundscapes or Foley. Sound effects were designed with *Steinberg's Cubase 8* and always with necessities for audio implementation in mind, for example the timing of different layers of one sound effect and fading between looping/continuous sounds.

b) Implementation

For randomization most of the sounds were picked randomly from a sound pool, while picking the one last played was restricted. Also, the pitch changes most of the time for the sound effects.
Randomization, layering and continuous sounds means dealing with many audio sources when just using *Unity*. The Boss even needs 14 because of different positions of guns, while player and Charger each have seven audio sources. Since multiple enemies of one type can be active on screen a general enemy audio manager was needed that distributes the sound to the enemies. This also deals with saving the last used Boss sounds when the Boss game object gets reset due to the player dying.

```
public void ChargerShoot(AudioSource audioSource)
{
    audioSource.pitch = Random.Range(.9f, 1.1f);
    audioSource.PlayOneShot(chargerShooting[nowChargerShoot]);
    lastChargerShoot = nowChargerShoot;
    int i = 0;
    while (nowChargerShoot == lastChargerShoot)
    {
        nowChargerShoot = Mathf.FloorToInt(Random.Range(0, chargerShooting.Length - .01f));
        i++;
        if (i > 1000) { break; }
    }
}
```

Fig. 14 Example for Randomization: Charger Shots Screenshot (InvokeRepeating)

75 See Sound Attributions

3.3 Shields

For the soundscape of the player using the shield a low-pass effect in *Unity* is applied on everything except the player ship sounds. Also, a shield ambience is played, mixed out of four different sounds: an Alesis airSynth (performed by the author), heavy reverbed recording of jamming on a soundbox, a hover engine (just the frequencies around 1 kHz) and a fan room sound (Lo-Cut at 3 kHz). The group signal of those four was altered by applying a chorus (more techy), a signal crusher (lo-fi player sounds) and a vintage compressor to glue everything together. The sound lasts 40 seconds, long enough so the player can not get to the end of the audiofile in-game because ammunition would run out first.

Since the duration of the shield depends on the players actions the ending of the sound has to be controlled by *Unity*. Apart from simply fading out and adding another airSynth snippet, there is also a low-pass and high-pass filter applied which both quickly sweep down from 2 kHz to 20 Hz for a „turning off“ audio effect, which is also added in reverse when turning the shield on.

75 See Sound Attributions

```
if (shieldLPnow != shieldLPTarget)
{
    if (shieldLPnow * shieldLPSpeed < shieldLPTarget)
    {
        shieldLPnow *= shieldLPSpeed;
    }
    else
    {
        if (shieldLPnow / shieldLPSpeed > shieldLPTarget)
        {
            shieldLPnow /= shieldLPSpeed;
        }
        else
        {
            shieldLPnow = shieldLPTarget;
        }
    }
    masterMixer.SetFloat("UILPFC", shieldLPnow);
    masterMixer.SetFloat("EnemyLPFC", shieldLPnow);
}
playerMove1.pitch = playerScript.activeSpeed * 10/3 + .5f;
playerMove2.pitch = playerMove1.pitch;
```

Fig. 15: Applying Low-Pass to out of Ship Sounds Screenshot (FixedUpdate)

```
if(cutOffShieldChange != 0)
{
    cutOffShield = cutOffShield * (1 + cutOffShieldChange/4);
    highPassShield.cutoffFrequency = cutOffShield;
    lowPassShield.cutoffFrequency = cutOffShield;
    audioShield.volume = Mathf.Pow(cutOffShield / highestCutOffShield, 2)*.5f;
    if (cutOffShield > highestCutOffShield && cutOffShieldChange > 0 )
    {
        cutOffShieldChange = 0;
        audioShield.volume = .5f;
    }
    if(cutOffShield < lowestCutOffShield && cutOffShieldChange < 0)
    {
        cutOffShieldChange = 0;
        audioShield.volume = 0;
        audioShield.Stop();
    }
}
```

Fig. 16: Turning the Shield on/off Effect Screenshot (FixedUpdate)

The sounds of being hit while having the shield active were composed of two different airSynth sounds in three variations. There are three ships that can be shielded (Player, Charger, Boss), which is why the samples are altered through three different insert chains, which leaves nine sounds in total.

The Player's sound is compressed and bit-crushed (lo-fi player), the Charger is loudmaxed and detuned, so the player notices, that enemies can be shielded too, and the Boss is reverbed and compressed, with much more reverb than direct signal to make the attack feel insignificant.

Fig. 17: *Cubase* Screenshot of Shield Sounds

3.4 Shots

a) Boss Laser

In order to get the impact difference of distinct shot types right, the weapons with the highest impact were designed first. Different shot types should on one hand sound unique, but on the other hand some layers were used in different shots for consistency in the diegetic universe.

This sound consists of 3 phases: two seconds of loading, three seconds of shooting and three seconds of reverb. A coil recording of a hair dryer with an Octaver VST loads in a low frequency band and shoots in a high frequency one. A different airSynth than above plays with a rising pitch as loading, an added burst at the shot start and transient shaping during shooting. A recording of a ZubeTube adds a classic laser sound charm to the mix, while a coil recording of a toothbrush is so heavily altered to sound something like raw energy bubbling.

A low and very noise-like rocket thrust effect adds energy that reminds one more of a burner. For more impact a low frequency bomb sound was added at the shot start and another one in combination with thunder rumbling at the end for the reverb.

The ship UI also warns the player of the laser in form of beeps rising in pitch and rate while charging and then staying the same while shooting. For added effect there are also some flickering glitches as if the player's diegetic UI is nearly failing because of the enormous power that is channelled through the laser.

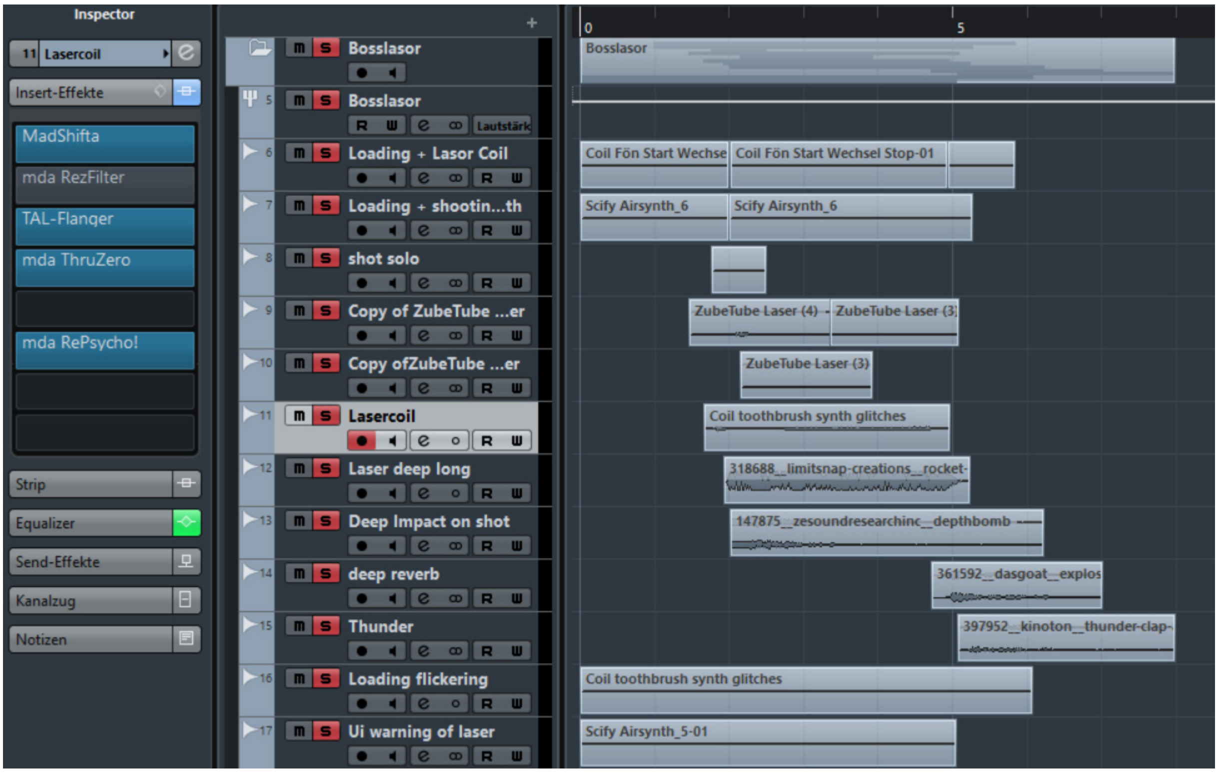


Fig. 18: *Cubase* Screenshot of Boss Laser Sound

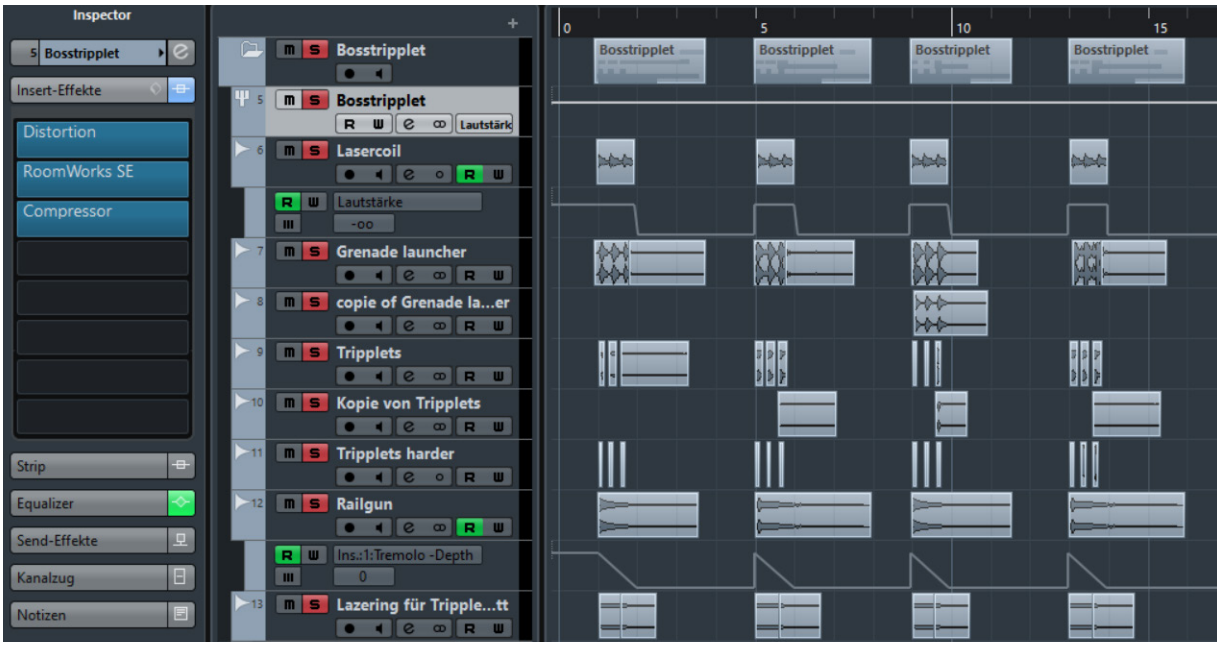


Fig. 19: *Cubase* Screenshot of Boss Triplet Shot Sounds

b) Boss Triplets

The base for this sound is a recording of an electric toothbrush with a coil microphone similar to the one of the big laser, but this time as short bursts with the deep basses cut. The triple shot prominently features a grenade launcher sound which was time-stretched to fit the 300ms delay between shots and high-cut at 4.5 kHz. Another layer consists of snippets from rubbing metal impacts with an octaver and extra distortion. The plucking of piano string was also distorted, time-stretched and reverbed. A tremolo railgun sound effect and an edited laser sound combine the triple shot into one sound cue. The sound group finally was distorted, reverbed, compressed and slightly filtered for details. There are four variations, two for the first Boss phase and two heavier ones for the second phase.

c) Charger

The loading and shooting sounds are separated for more variance. Basis for loading are weird pulses with always four of them being cut together, heavily filtered, chopped and overdriven. The overdrive effect starts muffled and gets clearer over time as the shot energy is collected. A metalized spaceship whoosh conveys bubbling energy in the high frequency range. A diddley bow sound resampled, reversed, three VSTs were used (see figure 20 highlighted track) and a filter was applied to provide strong low frequency energy. The whole sound was compressed and reverbed.
The Charger shot also contains two wire sounds: tapping a wire for background energy plus reverb as well as hitting it to sound like a shot (plus octaving and pitch-shifting). The more realistic sounding layers use the explosion at the start of a modern space shuttle sound effect and an explosion sound effect that sounds like a big cannon. Reverb was applied to finish the sound.

Fig. 20: *Cubase* Screenshot of Charger Loading and Shot Sounds

d) Up/Down

The sounds of these common enemies were kept quite simple. A pitch-shifted firework explosion is the "realistic" base layer, an octaved and pitch-shifted ZubeTube laser sound makes it feel more sci-fi and the bending of thick wires provide a deep reverb. Everything is compressed, limited and reverbed.

To help the player distinguish which shots are dangerous the volume of shot sounds is calculated by how far the enemy is away from the player on the x-axis (restricted between the values 0.25 and 0.5).

$$volume = 1 - \left(\frac{|x_{player,position} - x_{enemy,position}|}{width_{window}}\right)^{1.5}$$

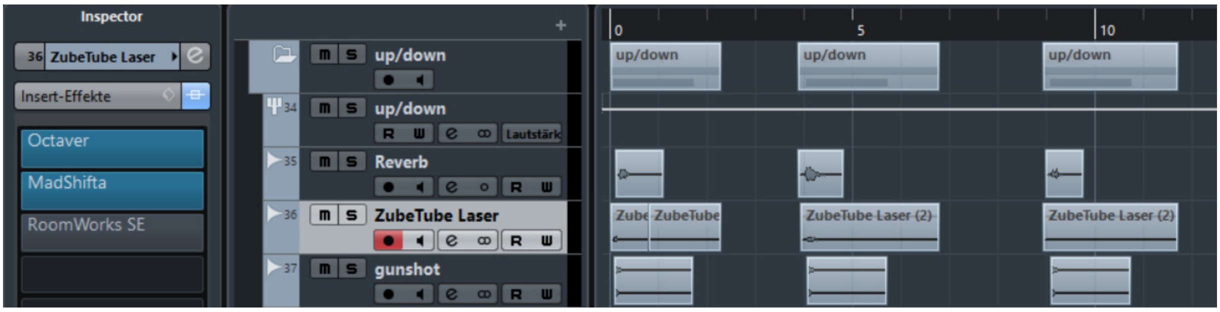


Fig. 21: *Cubase* Screenshot of Up/Down Shot Sounds

e) Player Fire

The player shot sound is one of the most important sounds. Since the fire rate is high the sound designer must make sure that it does not get annoying by being too extraordinary while also being interesting and identifiable for the player.

A gunshot and a short noise sound from a missile launch sound effect on one hand lay out a realistic base layer and on the other hand start very quickly, which makes the player feel more like he or she created the sound through pressing the left mouse button. The mechanical spring sound and the metal wire grinder are related to the sci-fi wire sounds but are very primitive sounding, like intended for the Used Future aesthetic. For an organic touch two different birds are used as sound sources. One is a modular synthesized bird sound. The other one comes from a lyrebird:

> To persuade females to come close, the male lyrebird sings the most complex song he can manage. And he does that by copying the sounds of all the other birds he hears around him – including the sounds of chainsaws and camera shutters![76]

The samples used here are the imitation of sci-fi lasers shots of those lyrebirds. Finally, building on the low-fi retro sound aesthetic an 8-bit shaper, bit-crusher as well as limiter and reverb were used to shape the sound.

There are four variations of the base sound and eight different bird samples each, so there are 256 variations of the sound which should be enough for the player not to get annoyed even though the very high fire rate.

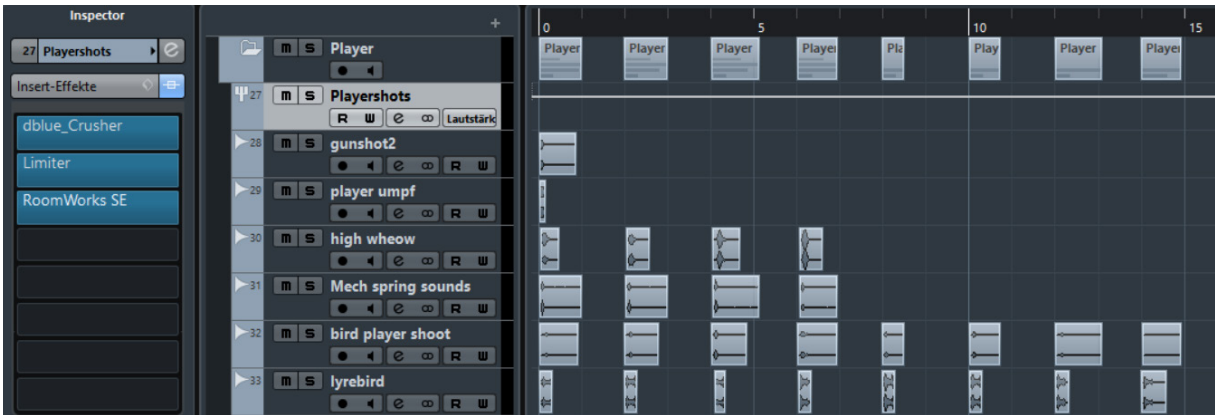


Fig. 22: *Cubase* Screenshot of Player Shot Sounds

76 BBC 2014

f) Player Ammunition

The player can hear how much ammunition is left through a low-pass filter attached to the shot sound playing audio source. Every shot the cut-off frequency of this low-pass filter is altered through a formula that is adjusted to logarithmic hearing and the wanted urgency level:

$$f_{LP,now} = 2 * (ammo_{in\,\%} * 100)^2 + f_{LP,lowest}$$

For the change of the sound to feel more natural the low-pass frequency is then altered in small steps (Fig. 23).

The shot volume is also adjusted to the percentage of ammunition left as a linear curve between 0.5 to 0.25, so the effect gets more prominent the lower the ammunition is to urge the player to reload. If the player shoots without having any ammunition, the clicking soundbites from reloading will play, to inform the player that input was recognized but no bullets are left. When the player reloads, the low-pass frequency is set back to 20 kHz and the volume back to 0.5.

g) Player Reloading

The time it takes to reload depends on how much ammunition is left, the maximum ammunition being 20 and the time it takes to reload is 0.1 seconds per shot. Initially every refill was meant to play a click sound, but half the repletion speed sounded better which is why there are ten different base layer reload sounds (0-1 left, 2-3 left, …, 18-19 left). These consist of a camera flash sweeping up two octaves and a reversed bass note, both time-stretched for every base sound so that they fit the duration between 0.2 seconds and 2 seconds. The sound was also filtered and bit-crushed. There are 17 click sounds from a toy gun trigger sound that was octaved, compressed and filtered. If the reload is finished there are four different soundbites to play. These consist of the synthesized bird sound from player fire high-cut at 16 kHz for a higher sounding ping and a sci-fi curser click sound low-cut at 9 kHz forming a lower sounding ping. The whole “reloading finished” sound is mixed with a copy of itself that was run through a Chebyshev polynomial of the 4th order and another equalizer.

```
if (LPShootChanged)
{
    if (Mathf.Abs(lowPassShoot.cutoffFrequency - cutOffShoot) > cutOffShoot * .05f)
    {
        if (cutOffShoot > lowPassShoot.cutoffFrequency)
        {
            lowPassShoot.cutoffFrequency *= 1.3f;
        }
        else
        {
            lowPassShoot.cutoffFrequency *= .95f;
        }
    }
    else
    {
        LPShootChanged = false;
    }
}
```

Fig. 23: Implementation of changing the Low-Pass depending on Ammunition (FixedUpdate)

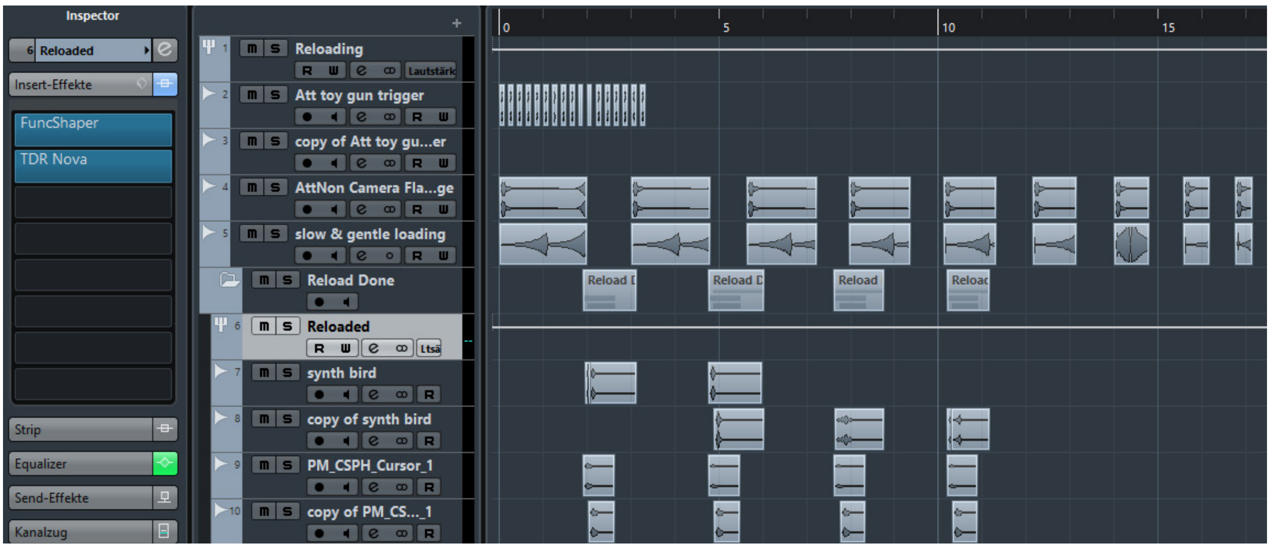


Fig. 24: *Cubase* Screenshot of Player Reloading Sounds

3.5 Damage

a) Enemies

Instead of giving enemies health bars another way of showing their status was chosen. The sound of hitting them with shots gives the player a feeling for their remaining energy. If an enemy has more than 70 percent of its health, the shot does not sound like it has a big impact, but if its health is below 30 percent it sounds like it is already falling apart. The Boss ship on the other hand always sounds like the shots are not really penetrating it, which makes it more frightening in the emotional aspect and more difficult in the gameplay aspect.

There are four samples each for the hits on high and mid health enemies, and 4 plus 3 samples for enemies on low health (using two layers in the game engine). Every finished sound was compressed and reverbed.

On high health the sound consists of a bullet ricochet explosion, the low frequency part of hitting a metallic cage and of a sniper shot. Everything was muffled to lower the perceived impact.

The shots on mid-life enemies consist of a short and muffled explosion, a down-sampled and function-shaped (tube-amp simulation) dropping of a key and the mid frequencies of a metal hit with much reverb.

On low health the base sound in high and low frequencies is hitting ice including a metallizer plugin. In the mid frequency spectrum there is already a notable explosion (overdriven & muffled) as well as concrete and glass flying with stutter reverb. For the low frequencies there is also the sound of pitched down metal rubble flying around.

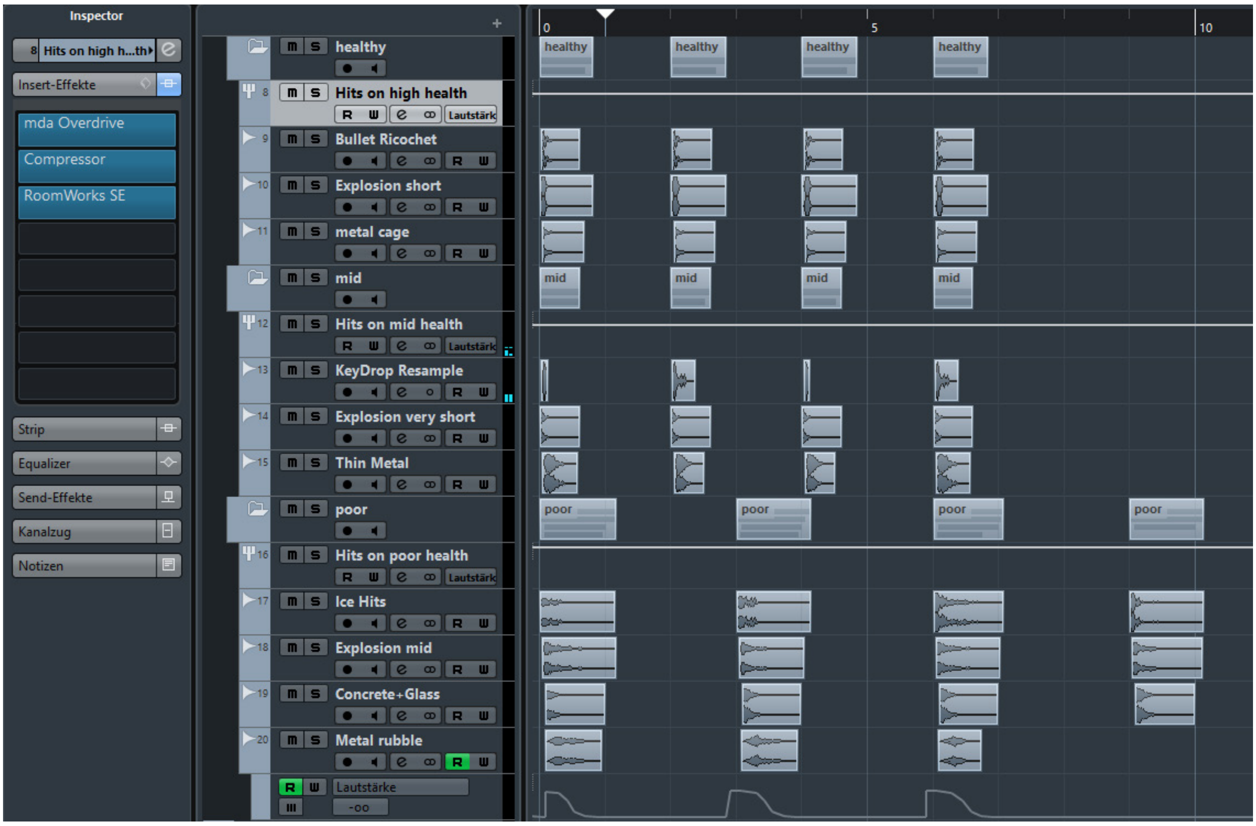


Fig. 25: *Cubase* Screenshot of damaging Enemies Sounds

b) Player

The sound of the player being hit consists of a variety of sounds, including metal & glass being hit, a car crash, a ship being hit by a cannonball (with a metallizer VST), a muffled explosion and a glitch sound for including a diegetic UI sound. Everything was compressed, reverbed and, as standard for player sound, slightly bit-crushed.

The player ship does not have much health, so taking damage should have a big impact but there is no need for many different damage sounds.

However, the player can take either one or two damage depending on the source, so there are two samples and two variations of it with an extra layer and with the room-size of the reverb boosted by 33 percent.

If the player would receive two damage but uses the shield, only one damage is dealt instead. In this case a normal damage sound and a "shielded" sound play, but the latter one's volume is halved to convey that the player should focus on avoiding this kind of damage instead of just shielding it. Also, the pixel shields opacity does not change here like it would with completely negated damage.

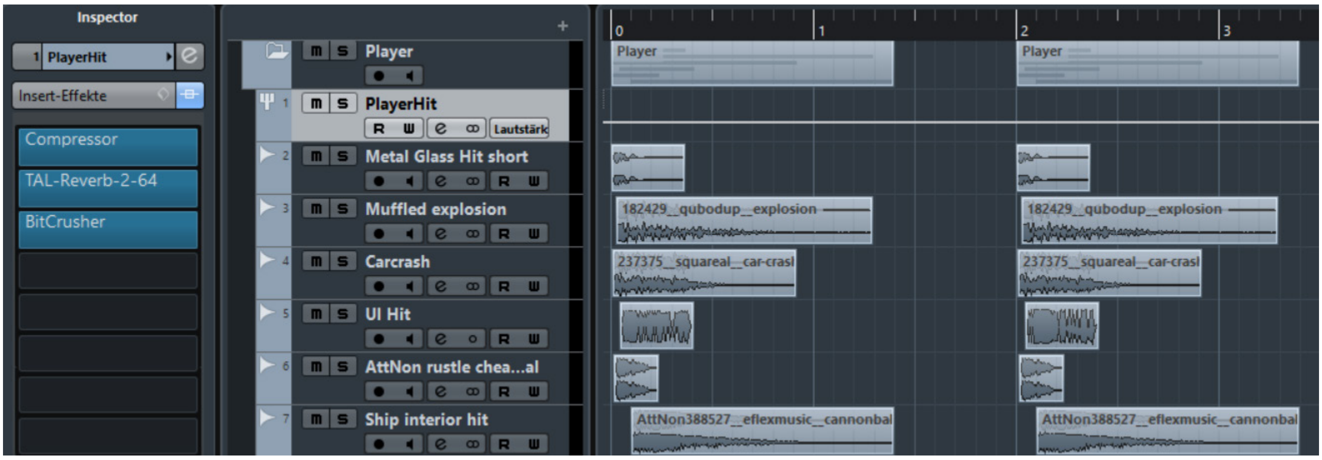


Fig. 26: *Cubase* Screenshot of damaging Player Sounds

```
public void Damage(bool shield, float damage, AudioSource exploAudio, float health)
{
    if (exploAudio != null)
    {
        exploAudio.pitch = Random.Range(.9f, 1.1f);
        exploAudio.clip = dying;
        exploAudio.Play();
    }
    else
    {
        if (shield)
        {
            gotHit.PlayOneShot(damageShielded[nowShield],.1f-damage*.5f);
            int i = 0;
            while (lastShield == nowShield)
            {
                nowShield = Mathf.FloorToInt(Random.Range(0, damageShielded.Length - .01f));
                i++;
                if (i > 1000) { break; }
            }
            lastShield = nowShield;

        }
        if (damage > 0)
        {
            if (damage < 2)
            {
                if (nowDamage1) { gotHit.PlayOneShot(damaging[0]); }
                else { gotHit.PlayOneShot(damaging[1]); }
                nowDamage1 = !nowDamage1;
            }
            else
            {
                if (nowDamage2) { gotHit.PlayOneShot(damaging[2]); }
                else { gotHit.PlayOneShot(damaging[3]); }
                nowDamage2 = !nowDamage2;
            }
        }
    }
```

Fig. 27: Implementation of damaging Player Sounds (OnTriggerEnter/OnCollisionEnter)

3.6 Movement

The general idea behind the movement was using musical intervals to determine the danger of the different enemies through their movement sound.

The movement sound for the player has a small peak at 262 Hz, which corresponds to a C3 in the twelve-tone equal-temperament scale.

The Kamikaze is boosted a minor third above at 311 Hz (Eb3), neither known as an extremely pleasant or unpleasant interval.

The Up/Down enemies have a peak at the tritone 370 Hz (Gb3), known as a very unpleasant interval.

The Charger's idle sound is boosted at 247 Hz (B2), creating a strong dissonance of the minor second pulling upwards, and one at 494 Hz (B3), creating a major seventh.

Using musical clusters was planned for the Boss movement, but its appearance and introduction is clue enough for the player that it is the most dangerous enemy.

Of course, players are not expected to determine the enemy by identifying the musical interval. The interval theory is used as a tool for shaping the player's perception of an enemy (if the player is used to western music).

a) Boss

For the Boss enemy there are three final layers to be used in *Unity*. The first is the base layer, consisting of a reverbed airSynth recording which is simulating a helicopter and a jet engine recording with a low-shelf filter. This sound triggers every 30 seconds the Boss is active (fading between repetitions). The second continuous sound is a recording of a fan phasing with itself, reverbed, filtered and send into a delay line. There are two different samples resulting in a two-minute loop.

There are also four additional sounds for special movements. Two are used for the Boss' entrance and re-entrance after losing a life, one for the Boss moving up again to reset itself when the player loses a life and a last one where the Boss descends because the game is over. These consist of another airSynth recording and a rocket thruster sound effect, both filtered, compressed and reverbed.

Fig. 28: *Cubase* Screenshot of Boss Movement Sounds

b) Charger

The Charger has 5 different audio layers mixed in *Unity*. When the enemy spawns and moves downwards into the frame, there is one base sound of a jet start-up mixed with one of eight different synthesizer bits containing a delay line. The two idle sounds that play the entire time are a filtered fan room sound and one of five different soundbites from a pitch-shifted and time-expanded recording of rubbing a pan with a wire whisk, which gives the Charger a creepy vibe.

When the Charger moves sideways before stopping to shoot at the player, one of seven sounds from another jet start-up sample is used.

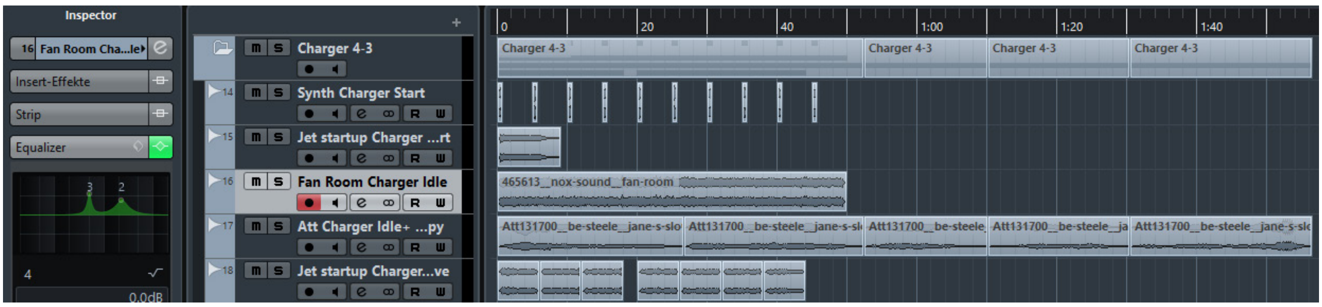


Fig. 29: *Cubase* Screenshot of Charger Movement Sounds

c) Up/Down

The base sound layer for these enemies consists of a rocket thruster effect, filtered jet noise and an alien drone sound (8 samples). The second layer is a giant fan recording, filtered to focus the lower frequencies to make it sound more intimidating. Everything is compressed, reverbed and filtered. There is a parametric EQ implemented in *Unity* to boost frequencies around 370 Hz even more. The enemies are intentionally kept interconvertible to make distinction a challenging game mechanic for the player.

Apart from just watching where the enemy shoots, the player can also listen to the movement sounds, because the Up enemy gets an additional third layer. These are eight samples of creepy humming wind. In an earlier version the fan layer was the extra layer for the Up enemy, but this was changed because the higher frequencies of the humming sound are easier to localize, and therefore they are clearer to distinguish when multiple enemies are on screen.

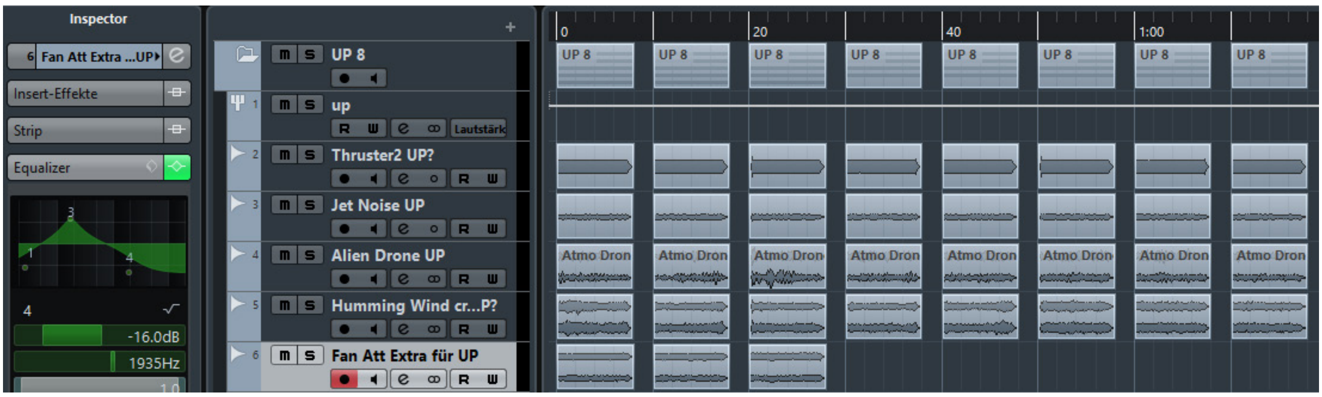


Fig. 30: *Cubase* Screenshot of Up/Down Movement Sounds

d) Kamikaze

The Kamikaze enemy is not that dangerous, which is why its sound should not be very dominant in a hectic scene. One sample was made that is altered by randomised pitch-shifting. The most prominent layer is a loud afterburner effect at the beginning. A resampled, pitch-shifted, octaved and filtered acceleration sound empowers the feeling of speed later in the sample.

A low drone noise and heavy reverbed recording of traffic give greater variance and base energy to the sound. Everything is compressed and filtered. Since the movement tone-boost is altered by the randomised pitch-shift in *Unity*, a parametric EQ was added to the mixer of the Kamikaze movement sounds in *Unity* itself.

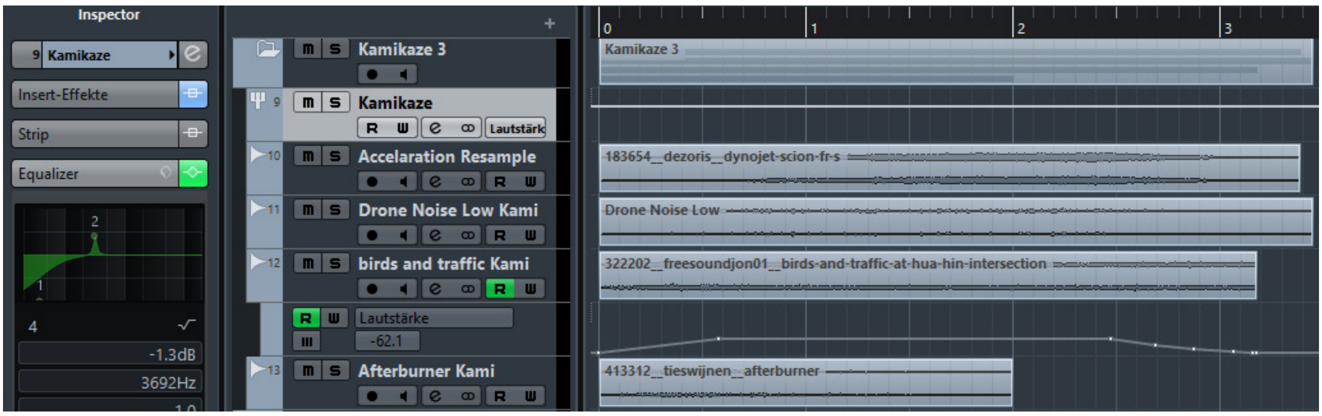


Fig. 31: *Cubase* Screenshot of Kamikaze Movement Sounds

e) Player

The player movement sound is a simple recording of a plane inside the passenger cabin. An enhancer adds low frequencies while an equalizer cuts the parts above 7 kHz as well as below 30 Hz and gives the boost at 262 Hz. This sound is looped in *Unity* and the pitch of the audio sources is changed depending on the active velocity of the player to give the player more feedback on his current speed.

The conversion from velocity to pitch is made so the pitch value is always between 0.5 and 1.5 while normal movement results in a barely altered pitch from the original version.
These values where chosen aesthetically and in mind with the earlier idea of having the player pitch be C3/262 Hz. The idea does not really work as musical intervals anymore, but it helps distinguish enemies.

```
activeSpeed = Mathf.Abs((transform.position - lastPosition).magnitude);
lastPosition = transform.position;

playerMove1.pitch = playerScript.activeSpeed * 10/3 + .5f;
playerMove2.pitch = playerMove1.pitch;

if (playerMove1.isPlaying && !playerMove2.isPlaying)
{
    if(playerMove1.time > 115.1f) { playerMove2.Play(); }
}
if (playerMove2.isPlaying && !playerMove1.isPlaying)
{
    if (playerMove2.time > 115.8f) { playerMove1.Play(); }
}
```

Fig. 32: Implementation of Player Movement Sounds (FixedUpdate)

3.7 UI

a) Select

The select sound is played on every UI interaction, for example clicking the "Resume" button. It consists of the very fast starting short *thump* of a basketball drop to immediately confirm interaction. For a sci-fi component there is also a beep added. Both are bit-crushed to add more character to the sound.

b) Wrong Input

If the player presses an UI element while another one is still processing, if he or she tries to reload at full ammunition or use the shield with no ammunition, a "wrong input" sound will be played. It consists of two similar beeps, pitch-shifted a bit differently to get a "no"-sounding downwards interval, and the low frequencies of a rougher beep. Everything is compressed, bit-crushed, reverbed and filtered.

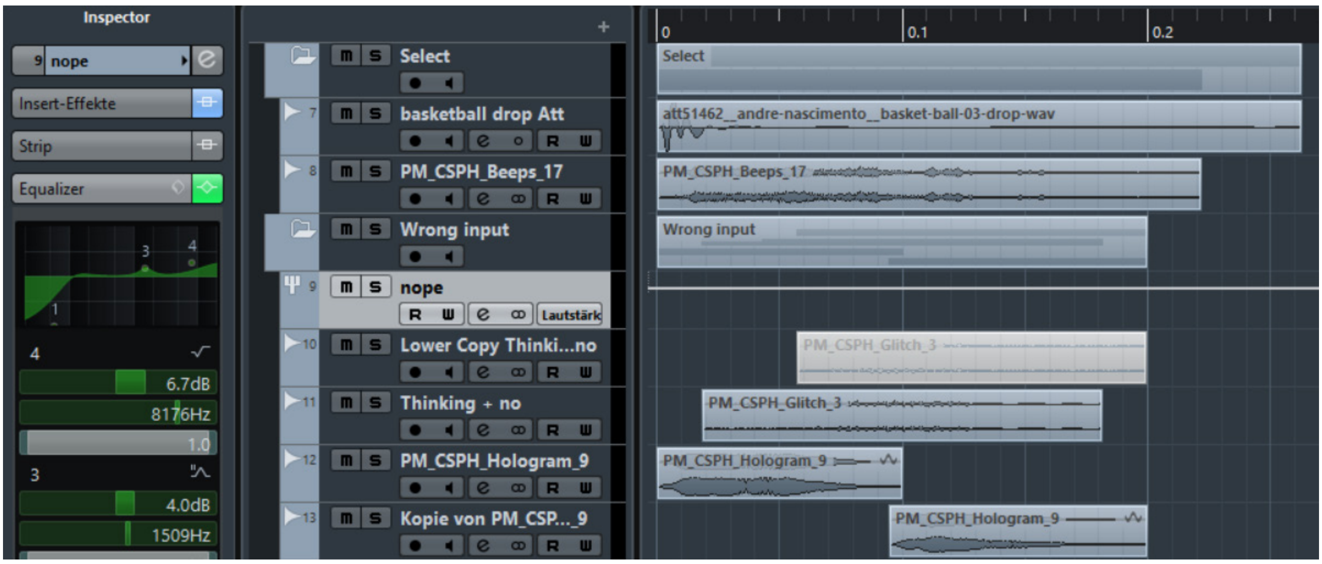


Fig. 33: *Cubase* Screenshot of Select and Wrong Input Sounds

c) Hover

The possibility of clicking a button is not only done visually, but also audible by using hover sounds. They consist of short and noisy bursts that are wave-shaped using soft clipping and bit-crushing. Low frequencies are filtered out and the 500 Hz area is a bit boosted for fine-tuning.

d) Quit

For quitting the game a combination of a confirmation beep, a closing UI sound sequence and a glitch sound play. They are compressed, reverbed and bit-crushed.

e) Start

For the typewriter-style text beeps were made to accompany every character as well as three beeps as the "incoming" notification and one for turning the transmission off. The beeps were recorded from the airSynth, cut apart, bit-crushed, reverbed and filtered. There is randomization in the timing of the characters and the six sounds are picked randomly too.

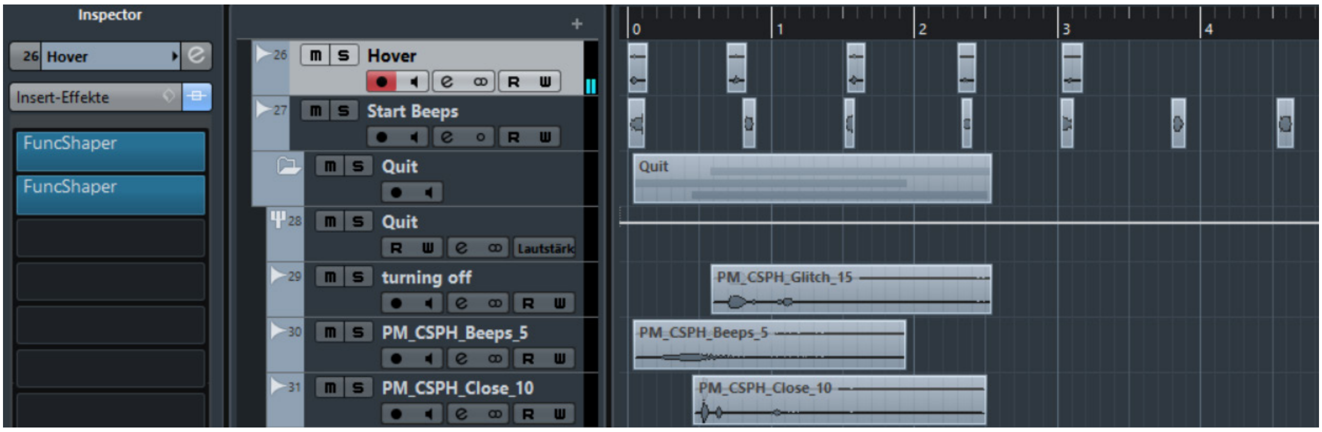


Fig. 34 *Cubase* Screenshot of Hover, Start and Quit Sounds

f) Respawn

Early in development it was determined that the best way to get rid of the on-screen enemies when the player dies was to fast forward the game through manipulation of the game's time scale. This was also the idea for the sound design: time passing until a new spaceship gets warped in. The volume, pitch and speed of a simple rhythm was automated to get the fast-forward feeling. This is accompanied by a pulsating noise.

Before Respawning a short "loading" sound is played, while two portal sounds ring out together on respawning (one in low and one in high frequencies). The whole sound is compressed, heavily reverbed and bit-crushed.

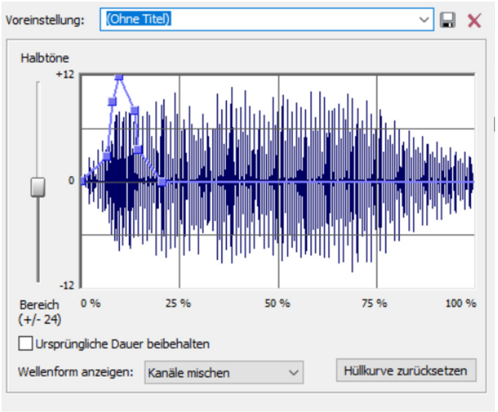


Fig. 35 *Magix SoundForge 10* Screenshot of the Rhythm Pitch Bend

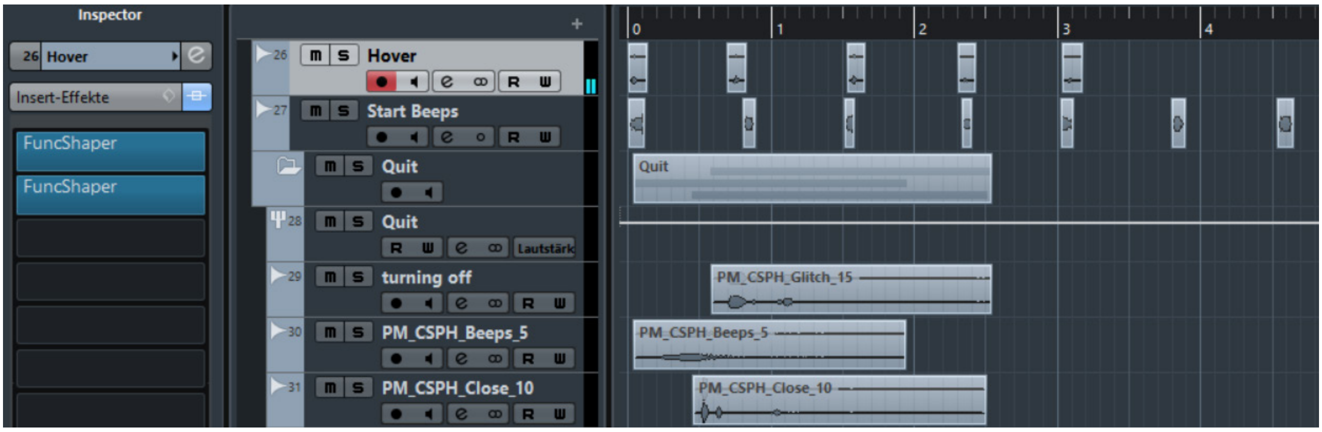


Fig. 36: *Cubase* Screenshot of the Respawn Sound

g) Restart

Since respawning equals fast-forwarding the idea for the restart sound was to rewind the diegetic space. The two most prominent sounds are beeps sounding like a computer calculating something for the sci-fi feeling and an airSynth synthesizer. The latter as well as tremolo frequency with some low frequency effects are rising and then falling in pitch.

The starting part of a "loading" sound is repeated six times before continuing to its full length. The sound of shooting a rocket is used as an added "whoosh" sound effect, then everything is compressed, heavily reverbed and bit-crushed.

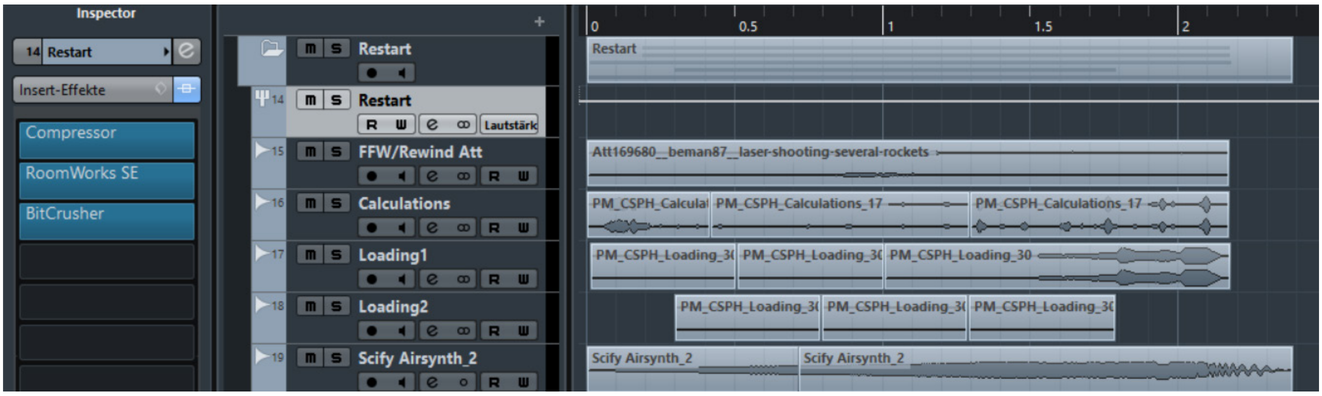


Fig. 37: *Cubase* Screenshot of the Restarting Sound

h) Health Item

The audio design of the health item was built on sweeps and coin sounds as incentives to picking them up. The graphic design of the coin with the player ship imprinted on it actually followed the sound design instead of the other way around.

When spawning the first sound recognizable consists the high parts of a coin flip recording. The other two sounds are "teleporting" sound effects, the first one being a low frequency effect sweeping up and the second one a high frequency effect sweeping a bit down. Everything was bit-crushed and compressed.

The sound of the player picking the health item up also consists also of a sweeping teleport sound effect and a coin flip, but both being high in frequency and more focused on the pickup impact. The sweep is reverbed and bit-crushed, the coin flip has automated reverb to extend it.

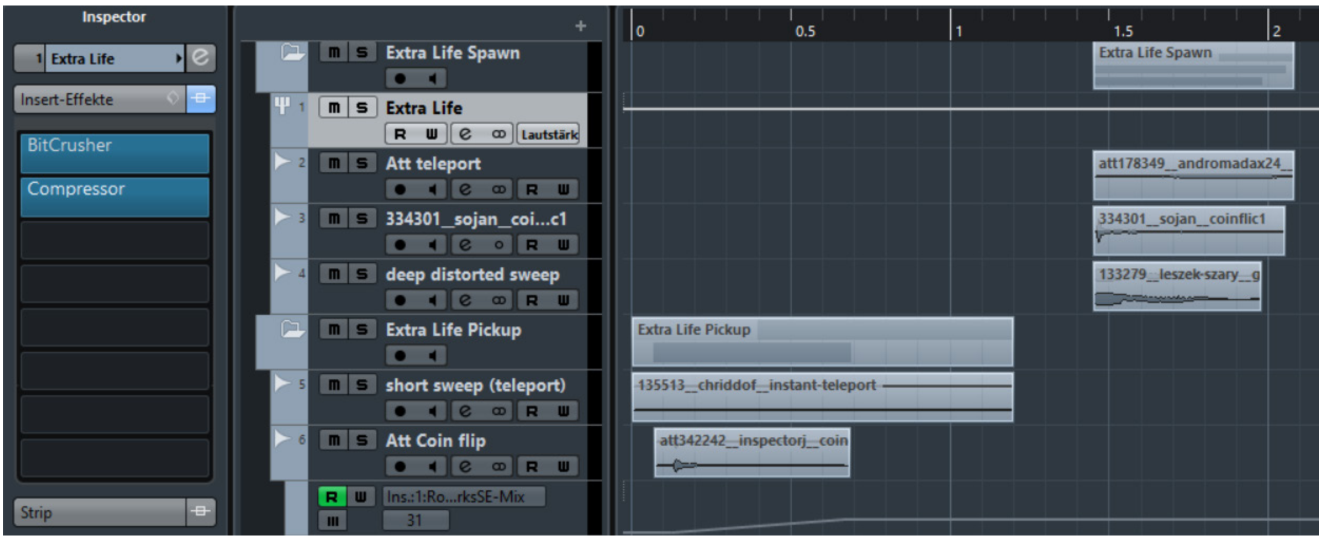


Fig. 38: *Cubase* Screenshot of the Health Item Sounds

i) Warning Low Hp

If the player has only one health point left, a continuous warning signal will play, urging the player to be careful, like in the *The Legend of Zelda* series. In order to not be too annoying, the beeps in this game are lower in frequency and instead more rough sounding to get the attention. The signal is made by duplicating a recording of the air-Synth and pitch-shifting both instances down in different amounts, to make use of the acoustic harshness effect. Finally, a bit-crusher was added. There are also various additional crackle, noise and distortion sounds to simulate that the controls of the spaceship are starting to fail.

j) Warning Shot from Below

When an Up enemy gets past the player it continues to shoot, even doing this from off-screen so a while. In order to make this not too unfair a singular alarm is played 0.3 seconds before this happens, consisting of a short alarm beep with heavy reverb and bit-crushing. The timing was chosen to warn the player as early as possible while also keeping the causal connection between warning and shot. The sound then will be panned in *Unity* according to where the enemy is on the x-axis in relation to the player.

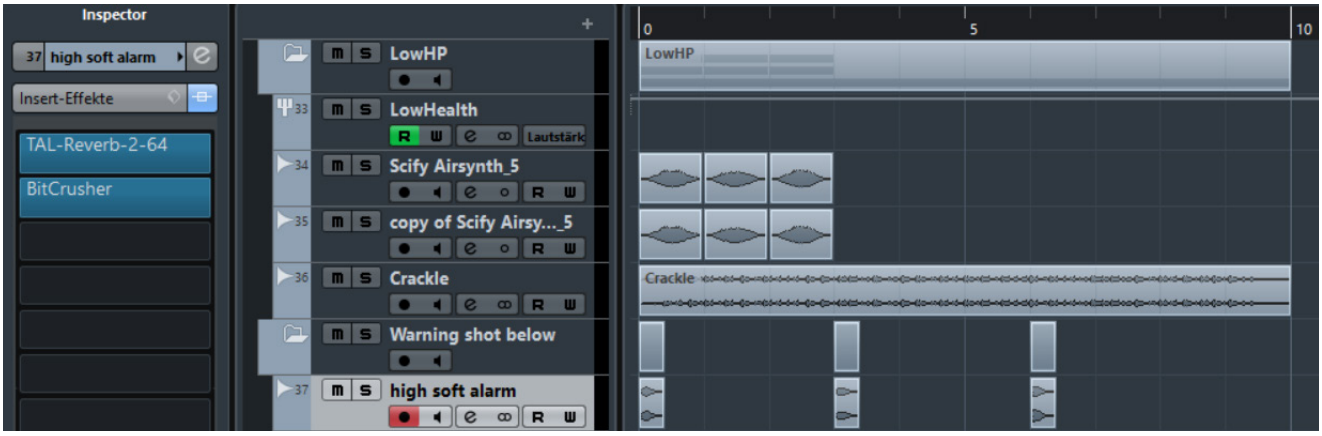


Fig. 39 *Cubase* Screenshot Low Health and Shot from below Sounds

3.8 Ambience

The ambience consists of different layers that fade in and out depending on the game stage the player is in. It starts relaxing and gets more abstract and disturbing to make the player feel like stepping from the home world into the alien world. This is also supposed to help the player act more carefully in later/harder levels. Even though the stage structure is mostly based on time, ambiences must be loopable because if the player dies the time it takes to get to the next level will be increased.

There is one general chapter ambience that plays the whole time between boss fights. The first chapter has a really relaxed and not too complex ambience. The second chapter is a more otherworldly "dreamscape" ambience with a metallizer plugin for it not to be too relaxing, creating the transmission to the final one, which is just weird and unsettling.

The first chapter consists of 4 stage ambiences that evolve from simple flying noise to a full-on horror drone, but all the ambiences are smooth in texture. The second chapter does the same development with three ambiences, but them being grainier. The last one is a crackling thruster-like noise followed by a spooky wind ambience, creating the calm before the storm in form of the last boss fight. Through this same setup the player can feel when if the boss stage is near because of the scary ambience instead of noisier ones. Also, the overall style gets more abstract and otherworldly over the whole progress of the game.

The Boss fight has its own ambience: A short and simple action rhythm loop giving it more tension, an intro and a creepy tech sound ambience. The intro differs between boss fights, the first one being a sci-fi alarm, the second one a rustling machine ambience and the third one something like creepy bee buzzing.

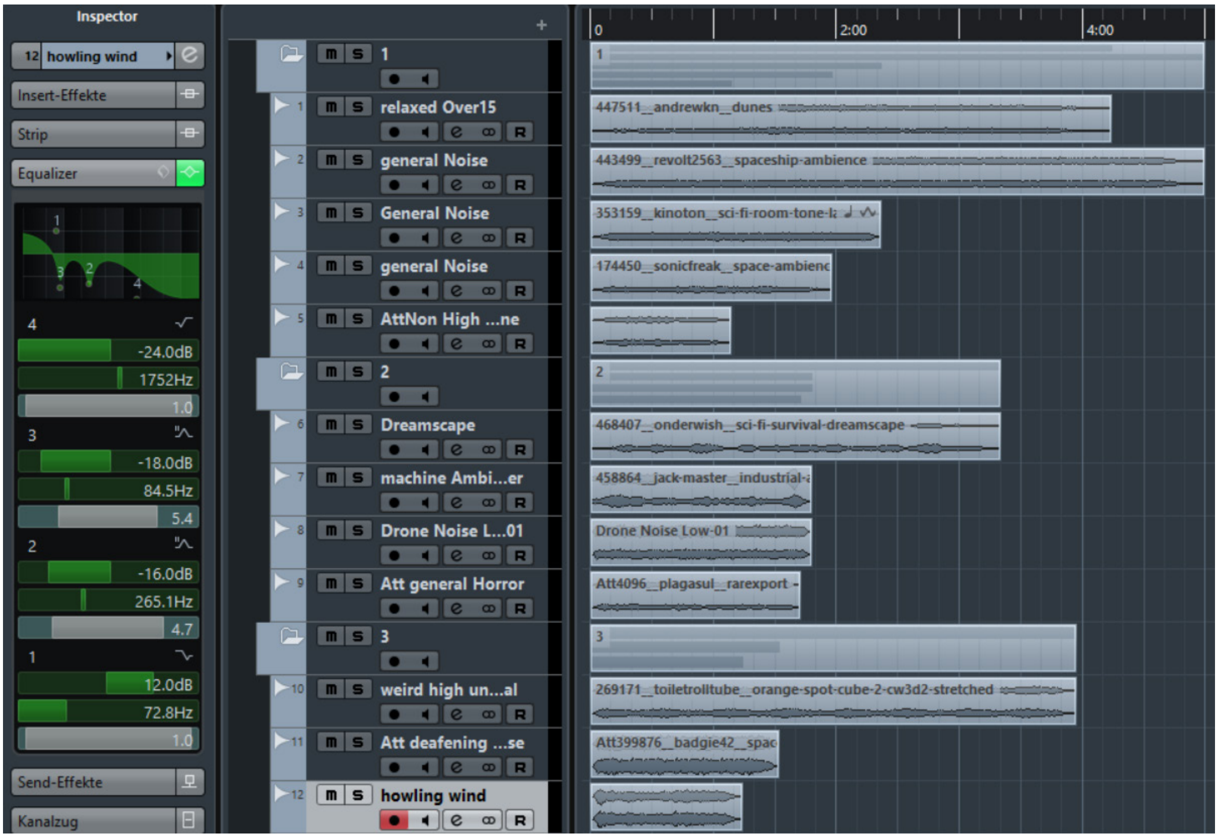


Fig. 40 *Cubase* Screenshot of non-Boss Ambiences

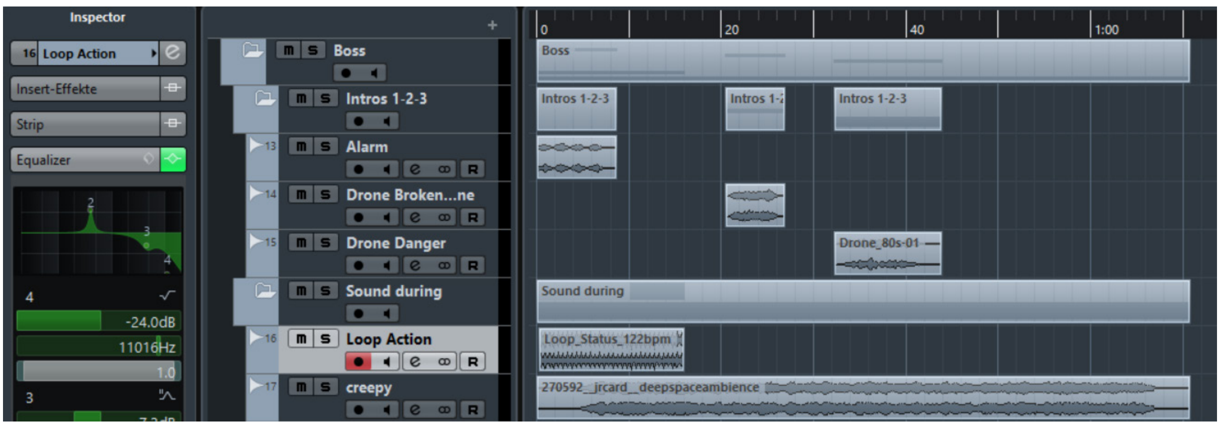


Fig. 41: *Cubase* Screenshot of Boss Ambiences

3.9 Music

Music was taken from the internet and just cut and faded to be loopable. During the actual gameplay there is no music to focus on the enemy sounds.

a) Main Menu

The player is greeted by a very simple synthesizer music. It conveys the intended sci-fi feeling and is not too distracting, because the main menu is just there for reading the controls before starting the game.

b) Intro

For the time of the intro transmission a short sci-fi music plays that is slowly faded in and out, and while fading in getting slowly boosted at 200 Hz for added effect.

c) Pause Menu

When entering the menu most audio sources are stopped, while the music and UI sound audio sources are not affected. The music is very relaxed to give the player an actual break from the gameplay. If the player exits the pause menu, the music will be paused instead of stopped, so when the game is paused the next time the music continues instead of always starting from the beginning. All other audios sources will continue playing.

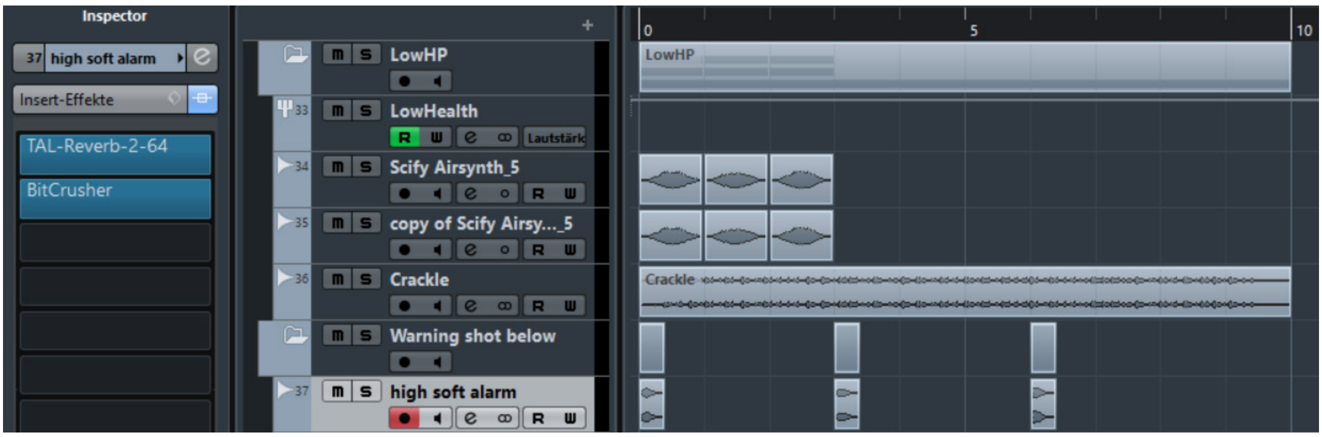


Fig. 42: *Cubase* Screenshot of Main Menu, Intro and Pause Menu Music

d) Game Over

If the player dies and did not get a new high-score, the music is supposed to resemble the feeling of drifting aimlessly through space in a shipwreck. To add to this there were also some computer malfunction and warning beeps mixed together.

If the player got a new high-score, a short upbeat loop plays to celebrate. When the player reaches the end of the game the synthesizer solo part of an upbeat music is playing to celebrate beating the game.

If the player completes the game without killing any enemies (in video games known as a pacifist run), the saved high-score is reset to zero and a short quirky melody is playing. This is just a little Easter Egg (hidden jokes or achievement in video games or other media) to reward exploration.

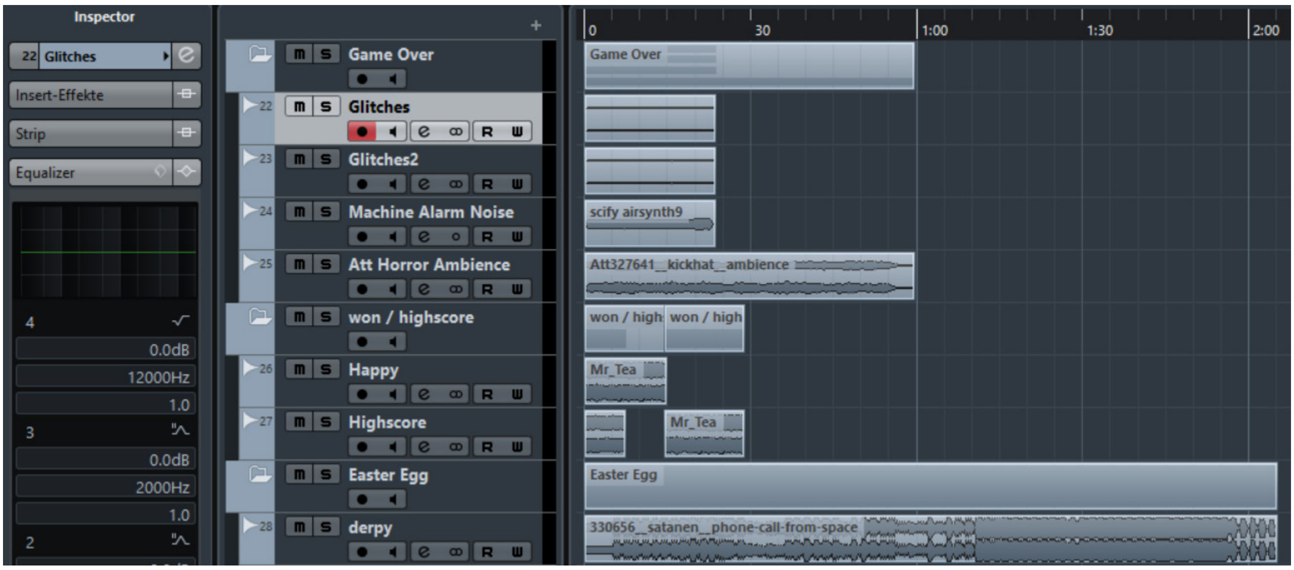


Fig. 43: *Cubase* Screenshot of Game Over, High-Score, Winning and Easter Egg Music

3.10 Adaptive Mixing

For every object and sound type a mixer in *Unity* was added, e.g. *Charger Shots*, *Player Shield* or *UI Effects*. There are also three groups called *Shots*, *Damage* and *UI* that duck the master signal to increase the impact of those actions. Additionally, the ambience is ducked by every other sound source to get more room for important sound effects. For each audio source in *Unity* the priority was roughly changed, so if more sounds than the computer can handle play at once *Unity* takes out less important ones.

For the player to still hear his or her own feedback sounds the *Player* mixer gets turned up the more enemies are on screen. This is done by keeping tabs of all the active enemies, adding up their numbers and using the power function to shape the curve. In a similar way enemy audio is altered. For every active enemy that is more dangerous the mixer is lowered by a small amount, to not get too high volumes at once and have the bigger threat stand out. As an example, the formula for the *Up* mixer volume is:

$$vol_{Up} = vol_{Up,start} - 2 * \left(amount_{active\ Chargers} + amount_{active\ Bosses}\right)^{0.8}$$

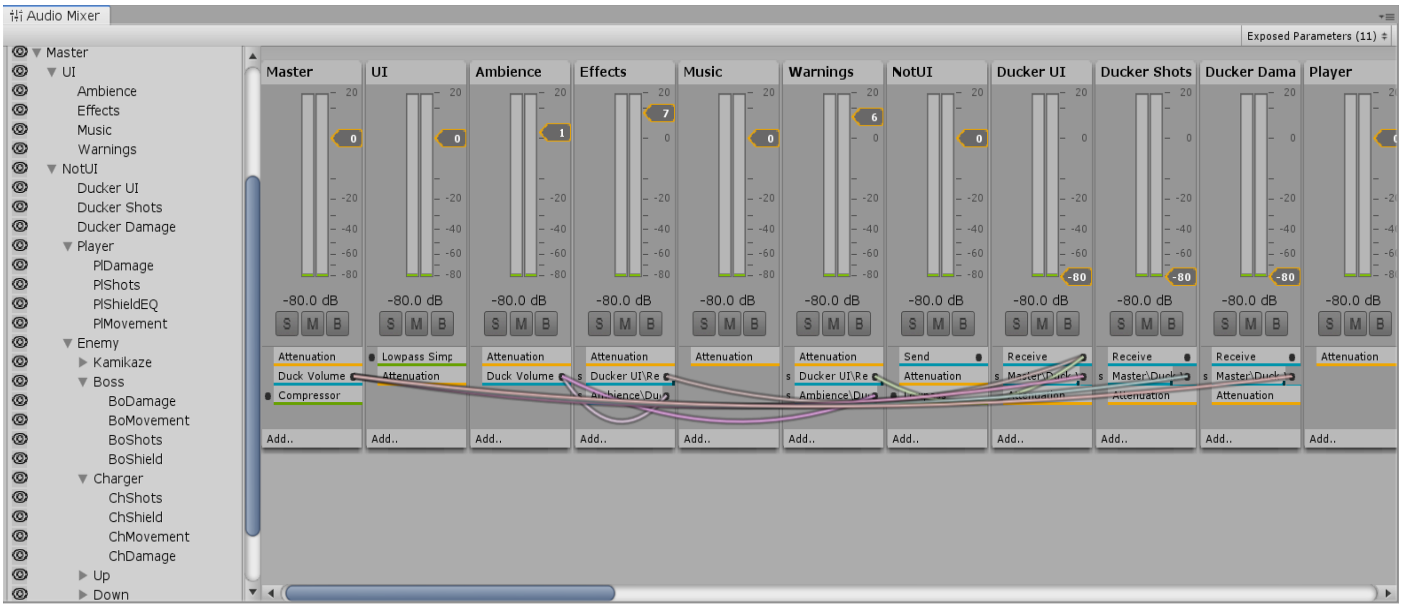


Fig. 44: *Unity* Screenshot of the Audio Mixer

Adaptive mixing was implemented to help players who need it. If the player has the starting 3 lives or more, nothing changes, but if fewer lives are remaining, the effects written about below get stronger to help the player not to get a "Game Over".

With games that feature many enemies at once it is essential for the player to shift focus between them. Each enemy has a specific timespan in which they are most dangerous, and some can be completely ignored at times. Kamikaze and Down are dangerous when they are starting and no threat when they passed the player. The Up enemy should be dealt with when it is above the player but really is just dangerous when it passed the player. The Charger is no threat in the beginning and cannot be damaged, but both this enemy and the Boss are most dangerous when charging up their shots. To help the player with this, these sections are also boosted or lowered depending on the player's lives left.

For the player to notice if the ammunition is running out while using the shield in a similar fashion to the shot sounds the lower the ammunition is the lower the shield volume gets.

```
if (helpStage > 0) {
    int i = 0;
    foreach (AudioSource audioSource in audioSources)
    {
        if (audioSource != null)
        {
            if (name.Contains("Kamikaze") || (name.Contains("Down")))
            {
                if (playRelPos > -1)
                {
                    audioSource.volume = Mathf.Max(audioSource.volume * (1 - Time.deltaTime), startValues[i] / (1 + helpStage / 3));
                }
            }
            if (name.Contains("Up"))
            {
                if (playRelPos > -1)
                {
                    audioSource.volume = Mathf.Min(audioSource.volume * (1 + Time.deltaTime), startValues[i] * (1 + helpStage / 3));
                }
            }
            if (name.Contains("Charger") || name.Contains("Boss"))
            {
                switch (state)
                {
                    case 0:
                        audioSource.volume *= (1 - helpStage / 6);
                        break;
                    case 1:
                        audioSource.volume = Mathf.Min(audioSource.volume * (1 + 2 * Time.deltaTime), startValues[i] * (1 + helpStage / 3));
                        break;
                    case 2:
                        audioSource.volume = Mathf.Max(audioSource.volume * (1 - Time.deltaTime), startValues[i]);
                        break;
                }
            }
            i++;
        }
    }
}
```

Fig. 45: Implementation of Adaptive Mixing for Enemy Sounds (FixedUpdate)

4. Game Résumé
4.1 General Discussion

The objective in creating this game was to demonstrate many techniques that alter the difficulty through sound. Most of the game design followed this premise, which is why the game design was kept relatively simple from a gameplay and story perspective.

When designing sound for video games, many difficulty-changing aspects are approached without thinking about it. For example, an audio designer instantly considers what one enemy type should sound like, so the aspects of localisation and differentiation are already involved. But when difficulty is already in mind in an early stage of the sound design and these tools are used consciously, they will be better executed resulting in stronger support for the player.

The research on which terms to use when talking about game audio functions, be it (extra-)diegetic, transdiegetic or the various degrees of interface sounds, clarifies the need to always think about the two core game functions at the same time: "When sound in films breaks this common separation between diegesis and extra-diegesis, it is understood as a stylistic, artistic and uncommon way of using sound, but games utilize this functionally to bind together usability and fictional space."[77]

Speech can be a great asset, especially in story driven games, but was passed on in this game due to the sparse story. Also, sound effects are more universal considering different languages, wear not out as fast as voice-lines and, if already familiar, are quicker in delivering information to the player.

77 Jørgensen 2006, p. 5

4.2 Aesthetics & Function

In SHMUP the audio layer plays a critical role in providing the player with information. This is done by creating sound effects that could be real in the diegetic world, which makes them more enjoyable for the player, instead of using very functional and raw sonification sounds. The designed audio clips access the players' intuition and emotions, to feel natural and fun instead of a chore. For example, by using musical intervals for the movement of the ships they become easier identifiable and clearer in their threat level. This example is an advanced technique that probably does not work well with most players, but generally the game's functions are made clearer through the right sound design.

When the player uses the shield, all outside noises get muffled by a low-pass, which makes the player feel protected. On the other hand, if he or she shoots at the shielded Boss, the hit feels insignificant because it is 99% reverb and nearly no impact. Sound effects tell about how something affected the game world in form of reactionary sounds. If an enemy is hit, the player immediately hears not only if it was shielded or not, but also the enemy's health. This is made audible by using three different sound effect pools for enemies with over 70 percent of their health, under 30 percent and in-between.

When designing UI audio, intervals that remind of speech patterns are used to convey emotion, see Wall-E (2008, Pixar) for reference, or acoustic phenomena like the harshness effect.

4.3 Supporting the Player

The player gets warned in different ways by pre-emptive sounds, for example enemies are audible and distinguishable before appearing on-screen. Some enemies, like the Boss, even telegraph their attacks visually and audibly through diegetic dedicated interface warning sounds (weapon loading and UI beeps rising in pitch). When the player gets shot at from below, another alarm is played, which is even panned depending on the player and bullet position on the x-axis. There are also warning signals in form of a low health beep plus crackling as well as notifications like the "Wrong input" sound or that reloading has been finished.

Orientation in form of the game progress is audible through the background ambience. Each of the three chapters between boss fights evolves from a relaxed atmosphere to a scary and weird soundscape, while the whole game progress is shown through the timbre of ambiences (smooth to grainy texture).

4.4 Multimodal Practices

Many events in the game are multimodal, for example a ship being damaged results in a sound effect as well as blinking. This helps players learn the game system faster and overall perform better. One implementation was to make it audible when shooting or using the shield how much ammunition the player has left before reloading. By doing this he or she does not have to look in the right corner of the screen and get killed while looking away from the action. It is still important that this information is also made available in the visual domain for the player to recognize the connections and learn them easier. Another way of conveying this information would be through explaining it in-game, but this would be rather inelegant.

The Up and Down enemies may look the same, but are very different, especially considering when they are a threat to the player. To help players distinguish them a movement sound layer was added for the Up enemy to make it more intimidating. Here, amodality is used as a gameplay mechanic.

The game is focussed on the unity of action and sound because, regarding player performance, this is more important than picture and audio. Kinaesthetic sympathy was utilized by developing a style that is not too far from known sci-fi sounds. It would have been even better to use more noises that players would have created themselves before (like the coin flip or basketball drop), but the science fiction theme generally demands other aesthetics. Most sound effects have enough variation in using randomly chosen iterations or layers and pitch shifting to make them feel more realistic instead of unnatural or annoying, but for fast learning and identification they are not too different from one another.

Amodality, conflicting displays or asynchronicity could have been used to make the game harder for players who might feel it is too easy, but for this thesis only positive reinforcement techniques through sound where implemented.

4.5 Mixing and Implementation

Dynamic mixing was implemented to help struggling players, by shifting their focus to the currently important information. As an example, the Down enemy is dangerous in the beginning and can nearly be ignored later, which is why it gets mixed up or down depending on if it is above the player or below the player on the y-axis.

Classic mixing also helps the player to not get overwhelmed and to give more impact to important actions. Depending on how many and which enemies are on screen the player and enemy volume is changed, so that he or she can hear the input feedback and the most dangerous enemies. Also, most player actions and other important sounds, like shots and warnings, duck the master signal, while the generally unimportant ambience is ducked by everything else.

The necessary audio implementation and testing effort should not be underestimated when planning a project (see 1.8 Audio implementation), for example evaluation workarounds for special game conditions, testing fades of music/ambience tracks and generally the mixture of different audio assets. With larger projects it is essential to map out as much as possible how things should be done beforehand, because new features or tools could generate new bugs or preventable problems and force revisions of virtually finished work. In case of this game's audio implementation the convenient mixer snapshots of *Unity* could not be used because of the way the adaptive mixing was implemented. Also, building many features from the ground up and having to alter copied versions in different scripts is tedious, which is why possible collaboration between audio programmers and sound designers or the use of middleware like *Wwise* should be considered when calculating how long the audio work on a game will take.

Conclusion

In this thesis the effect of sound design in relation to the difficulty of video games was examined. This was done by finding, compressing and presenting relevant research in the theoretical part and apply found practices in form of a shoot 'em up video game created for this thesis.

It was claimed that sound design is already widely used to form the difficulty level of games. This thesis is supported by the fact that there was not only scientific research about this, but also promotion in interviews and talks from working game sound designers. It was also found that even when designing the audio layer without thinking directly of difficulty, a certain amount of shaping is done by default. Sound design affects the difficulty of video games, and a good sound designer can form it in various degrees. The audio layer might not be the simplest form of doing this, but it is a very elegant way and worthwhile to use in combination with techniques using other senses. Various possible tools were implemented in the game to demonstrate the scientific studies on an actual game.

The research on which language to use when talking about game sounds shows the need to always think about usability and the fictional world at the same time, supporting both story and player. A shield should sound like it is shielding and a health bar can be realised by making different impact audio clips, forwarding critical information to the player through sound. UI noises give feedback on input reception and on conditions like "finished reloading" or "low health". Sounds can let the player know what and where something will happen, in form of warnings before attacks, through diegetic or UI sounds. Even the ambience can give the player orientation of space or something more abstract like game progress.

Essential effects, like movement sound, can play on the user's intuition and emotions to change the behaviour. They can also be a game mechanic, as demonstrated by differentiation of similar looking enemies through sound. Displaying information like "left ammunition" both visually and sonically helps the player to quickly learn sounds and, when learned, focus his or her vision on enemies instead of UI.

Since video games are focused on input, the timing of audio clips should be influenced by action instead of animation. Action sounds work better when including noises that the player created in real life before. Diversifying sounds, be it through randomization of pitch or layers, is essential for them not to get annoying or feel unrealistic.

Dynamic mixing helps players who need it by automatically mixing enemies in their dangerous/undangerous phase up/down. Player and enemy volume are changed depending on how many and which enemies are active, while ducking also helps important sounds to stand out.

Most of the above-mentioned tools could have been used contrariwise to make the game more difficult if the player performs too well and might get bored, but this video game is focused on helping instead of disrupting.

Audio implementation and testing can be underestimated, which is why enough time for planning, the use of both sound designers and designated audio programmers or usage of middleware like *Wwise* should be considered.

It was fascinating how the game instantly sounded significantly better when the cheap placeholder sprites were changed to the finished pixel art (compare 1.5.1 Combining auditory and visual stimuli). It is important to listen to the sounds in context and see if the aesthetics fit.

Another important aspect of game audio is real-time processing of reverb, occlusion, etc. Since the game plays in space these topics were disregarded, but in most other games it is important to keep them in mind. For example, sound effects should have as less reverb as possible, so reverb can be applied in the audio engine to get a consistent and good feeling of space.

Unfortunately, there was no time for testing, which is why the impact of each difficulty-changing tool is scientifically unclear. Nonetheless, they worked for the author and can be adapted and tested in further research.

Since the production of video games is an industry, it would be interesting which techniques are most elegant and efficient for their purpose. More insight on scientific parameters like frequency and rhythm could be useful, too. For this purpose, further research on sonification should also be studied, because they can often be applied to game sound design.

Of course, not all possible techniques to form a games difficulty through sound design were explored. For each specific project it is essential to stay creative and come up with strategies that fit the project the best.

Designing sound for games is a fascinating and manifold work. There are seemingly endless possibilities for aesthetic creativity and player influence, be it emotionally, functionally or, as shown in this thesis, both.

List of References

Bahrick, Lorraine E./Lickliter, Robert: *Intersensory Redundancy Guides Attentional Selectivity and Perceptual Learning in Infancy.* Developmental Psychology, 36, 2 (2000), pp. 190-201

BBC (April 21, 2014): *An Australian bird that mimics the sound of a chainsaw.* In: http://www.bbc.com/travel/story/20140416-an-australian-bird-that-mimics-the-sound-of-a-chainsaw (last consulted August 26, 2019)

Brazil, Eoin/Fernström, Mikael: *Navigation of Data.* In: Hermann, Thomas/Hunt, Andy/Neuhoff, John G. (ed.): The Sonification Handbook. 1st Edition. Berlin: Logos Publishing House 2011, pp. 509-523

Brown, Khris (August 16, 2019): *Fighting Legos: Audio Choreography for Battle Design.* In: https://www.youtube.com/watch?v=q6Vs3d6CrUc (last consulted August 17, 2019)

Brown, Mark (November 11 2014): *Adaptive Soundtracks | Game Maker's Toolkit.* In: https://www.youtube.com/watch?v=b0gvM4q2hdI (last consulted February 23, 2019)

Brown, Mark (October 19 2017): *How Cuphead's Bosses (Try to) Kill You | Game Maker's Toolkit.* In: https://www.youtube.com/watch?v=F8T6Ul4aHTI (last consulted February 23, 2019)

Collins, Karen: *Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design.* Cambridge: The MIT Press 2008

Collins, Karen: *Making gamers cry: mirror neurons and embodied interaction with game sound.* In: Proceedings for the 6th Audio Mostly Conference (2011), pp. 39-46

Collins, Karem: *Playing with Sound. Cambridge*: The MIT Press 2013

Cowan, Brent et al.: *Effects of sound on visual realism perception and task performance.* In: The Visual Computer, 31, 9 (2015), pp. 1207-1216

Darth, Sebastian: *How does Sound Design Impact the Players Perception of the Level of Difficulty in a First Person Shooter Game?* Dissertation, 2015.

Dixon, Mike et al.: *The Impact of Sound in Modern Multiline Video Slot Machine Play.* In: Journal of Gambling Studies, 30, 4 (2014), pp. 913-929

Ekman, Inger: *Meaningful Noise: Understanding Sound Effects in Computer Games.* In: Proceedings of the conference on Digital Arts & Culture (2005)

Ekman, Inger: *Psychologically motivated techniques for emotional sound in computer games.* In: Proceedings for the 3rd Audio Mostly Conference (2008), pp. 20-26

Fricke, Nicola: *Warn- und Alarmsounds im Automobil.* In: Spehr, Georg (ed.): Funktionale Klänge: hörbare Daten, klingende Geräte und gestaltete Hörerfahrungen. Bielefeld: Transcript, 2009, pp. 45-64

Hug, Daniel: *Ton ab, und Action! Narrative Klanggestaltung interaktiver Objekte.* In: Spehr, Georg (ed.): Funktionale Klänge: hörbare Daten, klingende Geräte und gestaltete Hörerfahrungen. Bielefeld: Transcript, 2009, pp. 143-170

Huiberts, Sander/Van Tol, Richard (January 23, 2008): *IEZA: A Framework For Game Audio.* In: http://www.gamasutra.com/view/feature/131915/ieza_a_framework_for_game_audio.php (last consulted February 22, 2019)

Jørgensen, Kristine: *On the Functional Aspects of Computer Game Audio.* In: Proceedings of the Audio Mostly Conference (2006), pp. 48-52

Jørgensen, Kristine: *On transdiegetic sounds in computer games.* In: Northern Lights: Film and Media Studies Yearbook, 5, 1 (2007), pp. 105-117

Jørgensen, Kristine: *Time for New Terminology? Diegetic and Non-Diegetic Sounds in Computer Games Revisited.* In: Grimshaw, Mark (ed.): Game Sound Technology and Player Interaction: Concepts and Development. IGI Global, 2011, pp. 78-97

Kent, Steven L.: *The Ultimate History of Video Games: from Pong to Pokemon and beyond...the story behind the craze that touched our lives and changed the world.* Roseville: Prima Publishing 2001

Kristal-Ern Alfred: *Can sound be used to effectively direct players' attention in a visual gameplay oriented task?* Dissertation, 2017

Liljedahl, Mats: *Sound for Fantasy and Freedom.* In: Grimshaw, Mark (ed.): Game Sound Technology and Player Interaction: Concepts and Developments. IGI Global, 2011, pp. 22-43

McGee, Marshall (February 28 2017): *Is Overwatch the best sounding shooter ever?* In: https://www.youtube.com/watch?v=MbV_wKScrHA (last consulted February 23, 2019)

Ng, Patrick/Nesbitt, Keith: *Informative Sound Design in Video Games.* In: Proceedings of the 9th Australasian Conference on interactive entertainment (2013), pp. 1-9

Östlund, Daniel: *Examining Sound Design's Effect On A Player's Wayfininding By Utilizing Musical Intervals.* Dissertation, 2015.

Papworth, Nigel: *iSpooks: an Audio focused Game design.* In: Proceedings of the 5th Audio Mostly Conference (2010), pp. 1-8

Phillips, Winifred: *A Composer's Guide to Game Music.* Cambridge: The MIT Press 2014

Pires, Durval/Alves, Valter/Roque, Licinio: *A Software Architecture for Dynamic Enhancement of Soundscapes in Games.* In: Proceedings of the 9th Audio Mostly Conference (2014), pp. 1-8

Quarles, Zachary (September 25, 2017): *Postmortem: Greater accessibility through audio in Killer Instinct.* In: Postmortem_Greater_accessibility_through_audio_in_Killer_Instinct.php (last consulted February 22, 2019)

Raessens, Joost/Goldstein, Jeffrey (ed.): *Handbook of Computer Game Studies.* Cambridge: The MIT Press 2005

Rojas, David et al.: T*he Effect of Sound on Visual Fidelity Perception in Stereoscopic 3-D.* In: IEEE Transactions on Cybernetics, 43, 6 (2013), pp. 1572-1583

Seitz, Aaron R./Kim, Robyn/Shams, Ladan: *Sound Facilitates Visual Learning.* In: Current Biology, 16, 14 (2006), pp. 1422-1427

Serafin, Stefania et al.: *Sonic Interaction Design.* In: Hermann, Thomas/Hunt, Andy/Neuhoff, John G. (ed.): The Sonification Handbook. 1st Edition. Berlin: Logos Publishing House 2011, pp. 87-110

Storms/Zyda, Russell L./Zyda, Michael J.: *Interactions in Perceived Quality of Auditory-Visual Displays.* In: Presence, 9, 6 (2000), pp. 557-580

Turner, Jonaas (October 13, 2015): *Oh My! That Sound Made the Game Feel Better!* In: https://www.youtube.com/watch?v=1Z5WqC97uZQ (last consulted February 28, 2019)

Uzer, Can: *Conceptual Design Methods for Game Sound.* Dissertation, 2016.

Waves (October 19, 2017): *Game Sound Design Tips from Dwight Okahara.* In: https://www.waves.com/game-sound-design-tips-dwight-okahara (last consulted July 15, 2019)

Wiechers, Harm-Jan (January 22, 2016): *The influence of music on the perception of difficulty.* In: http://www.gamasutra.com/blogs/HarmJanWiechers/20160122/264144/The_influence_of_music_on_the_perception_of_difficulty.php (last consulted February 22, 2019)

Table of Figures

Sound Attributions

This game uses sounds from freesound:
Basket_ball_03_drop.wav by andre.nascimen (https://freesound.org/people/andre.nascimento/)
S_Teleport_01 by Andromadax24 (https://freesound.org/people/Andromadax24/)
Space Hulk Reactor Tunnel by Badgie42 (https://freesound.org/people/Badgie42/)
Laser shooting several rockets by beman87 (https://freesound.org/people/beman87/)
Jane's Slow Pan by be-steele (https://freesound.org/people/be-steele/)
Train destruction by CGEffex (https://freesound.org/people/CGEffex/)
Space ship atmos by CuddleNucks (https://freesound.org/people/CuddleNucks/)
Teleport-01a by DWOBoyle (https://freesound.org/people/DWOBoyle/)
Cannonball Hitting Ship (Interior) (Mixed) by EFlexMusic (https://freesound.org/people/EFlexMusic/)
Fan by geodylabs (https://freesound.org/people/geodylabs/)
Jet Engine 1 by InSintesi (https://freesound.org/people/InSintesi/)
Coin Flipping, A by InspectorJ (https://freesound.org/people/InspectorJ/)
ROCKET-EXPLOSED by Julien Nicolas (https://freesound.org/people/Julien%20Nicolas/)
Soundbox by justkiddink (https://freesound.org/people/juskiddink/)
Ambience by kickhat (https://freesound.org/people/kickhat/)
Wire-tapping by klangstrand (https://freesound.org/people/klangstrand/)
Rarexport-05 by plagasul (https://freesound.org/people/plagasul/)
Corrugated Iron Crash by ReyDros (https://freesound.org/people/ReyDros/)
Big space Drone 3 by Speedenza (https://freesound.org/people/Speedenza/)
Short space theme by TheFlakesMaster (https://freesound.org/people/TheFlakesMaster/)
less-artifacts-remix-#2-of-121864-timbre-before-after-remix-3-of-97151-franq-flash-charging by Timbre (https://freesound.org/people/Timbre/), using *Flash charging* by Franq (https://freesound.org/people/Franq/)
Toy gun trigger lighter by TRNGLE (https://freesound.org/people/TRNGLE/)
Unfa's Menu Sounds by unfa (https://freesound.org/people/unfa/)
Zatdude_SoundsMix by ZatDudeOnline (https://freesound.org/people/ZatDudeOnline/)