CREATING AND DESIGNING SOUND EFFECTS FOR A MOBILE GAME

Pasi Pitkänen

Bachelor's thesis
May 2013
Degree Programme in Media
Digital Sound and Commercial Music

TAMPEREEN AMMATTIKORKEAKOULU
Tampere University of Applied Sciences

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Media
Digital Sound and Commercial Music

Pasi Pitkänen: Creating and Designing Sound Effects for a Mobile Game

Bachelor's thesis 44 pages, appendices 0 pages
May 2013

The aim of this thesis is to provide detailed theoretical and practical information on creating and designing sound effects for mobile games.

This thesis first gives a brief look into the history of game audio and its evolvement from simple bleeps and bloops to more realistic sounding soundscapes. Secondly, the thesis presents detailed knowledge of sound as a physical phenomenon and describes how the human ear perceives this aural information.

The middle part of the thesis gives a comprehensive look into the sound design production process of an Angry Birds mobile game. In the final part, the thesis brings up the challenges that are commonly encountered in mobile game audio and tries to find the ways to overcome them.

Key words: sound designer, sound design, sound effects, game audio, mobile game

TABLE OF CONTENTS

1 INTRODUCTION 6
2 A BRIEF LOOK AT THE HISTORY AND DEVELOPMENT OF GAME AUDIO 8
2.1 The bleeps, bloops and blips 8
2.2 Rise of home video gaming and chiptunes 9
2.3 Game audio comes of age 11
2.4 Squeezing sound into your pocket 13
3 WHAT IS SOUND AND HOW DOES IT WORK? 15
3.1 The Ear and the Perception of Sound 15
3.2 Sound Waves 16
3.3 Phase 18
3.4 The Speed of Sound 18
3.5 Frequency 18
3.6 Amplitude 19
3.7 Decibels 19
3.8 Acoustics 19
3.9 Reverberation 19
4 WHAT ARE SOUND EFFECTS? 20
4.1 Hard Effects 20
4.2 Foley Sound Effects 20
4.3 Background Effects 21
4.4 Production Elements 21
4.5 Sound Design Effects 21
5 PRE-PRODUCTION 23
5.1 Planning 23
5.2 Scheduling 24
6 PRODUCTION 26
6.1 Preparations 26
6.2 Recording 27
6.3 Editing 29
6.4 Noise reduction 30
7 PRODUCTION - PART II 32
7.1 Designing and mixing 32
7.2 Mastering 35
7.3 Batch processing / converting 35
7.4 File naming and formats 36

7.5 Delivery and Implementation of sound effects 36

8 WHAT ARE THE CHALLENGES IN MOBILE GAME AUDIO AND HOW CAN THEY BE OVERCOME? 38

8.1 Team size 38

8.2 File size 39

8.3 Hardware limitations 40

9 REFLECTION 41

REFERENCES 43

GLOSSARY

MIDI	Musical Instrument Digital Interface
DSP	Digital Signal Processing
DAC	Digital-to-Analog Converter
Hz	Hertz
dB	Decibel
FM	Frequency Modulation
SFD	Spectral Frequency Display

1 INTRODUCTION

The amount of people using mobile devices like smartphones and tablets has rapidly increased in recent years. People can easily and effortlessly access the internet and their favourite services through their mobile devices. The huge amount of mobile devices, combined with their easy access to the internet, has given a great boost to the mobile games industry, which is now flourishing and growing faster than ever before. Today, people spend more time playing games on mobile devices than watching television, so the demand for mobile games with unique and high quality audio content is high. (Wired 2013.)

The process of deciding the topic for my thesis was a short one as I instantly knew that I wanted to write my thesis about game audio, so I started by brainstorming the specific areas that I wanted to include. I wanted to focus more on mobile game audio since this part of the game industry is blooming at the moment and I also design audio content for mobile games for a living. After the brainstorming session I picked my topic and chose to write about how to create and design sound effects for a mobile game.

The first part of the thesis is more theoretical and is mainly based on written material (books, articles etc.) that has been published before by various authors working in the field of game audio. All the written material was carefully chosen to reflect the topic of the thesis and, more importantly, give the reader comprehensive information of the history and development of game audio, what sound is and how it works, and what are the five different categories for sound effects.

The second part of the thesis gives a good practical look inside my usual workflow and process for creating sound effects for a mobile game. I will use the Pig Dipper update to Angry Birds Space as a case study for this, since I created all the audio content for the update. I will show how I recorded, edited and mixed some of the audio content for the update. This part will also provide a glimpse to the audio implementation side of the process.

In the third and final part of the thesis I will identify some of the basic challenges that are present in the mobile game audio field, and I will suggest ways to overcome these challenges.

The primary goal of this thesis is to provide a good amount of useful theoretical and practical information to the reader who wants to know more about the field of mobile game audio. The secondary goal is to give the reader an inside look at the actual creation and design process of the audio content for a mobile game. By opening up my workflow I can easily present the reader the different challenges I faced during the production process of Angry Birds Space: Pig Dipper.

TABLE 1. Planned and fulfilled use of time

STAGE	**PLANNED**	**FULFILLED**
Planning	100	110
Pre-production	15	15
Production	135	135
Gathering references	5	5
Studying references	20	15
Writing	115	120
TOTAL	390	400

2 A BRIEF LOOK AT THE HISTORY AND DEVELOPMENT OF GAME AUDIO

2.1 The bleeps, bloops and blips

When comparing the game industry to the film industry, it is easy to see that the game industry is pretty young and still developing. The very first video games were developed in the late 1950s and early 1960s. William Higinbotham's unpublished tennis game *'Tennis for Two*' dates from 1958 and, '*Spacewar!'*, developed at the Massachusetts Institute of Technology, in 1962. Both of these games had no sound at all but everything changed when a pinball company called Nutting Associates released the first mass-produced video arcade game, *'Computer Space'*, in 1971. *'Computer Space'* had many different sounds that ranged from engine thrusters to firing missiles and even to explosions. The sound effects did not sound realistic but that did not stop the company from advertising the pinball machine with flyers where they emphasized its sound capabilities. (Collins 2008, 8.)

The first video arcade game that helped to give birth and create momentum for the game industry was '*Pong*', developed and published by Atari in 1972. '*Pong*' became a huge success and soon after its release, numerous companies entered the game industry and started to create their own versions. The sounds in '*Pong*' were just simple bleeps and bloops created with the on-board sync generator. In the end, the sounds acted as catalysts in making game sound more widely known. (Collins 2008, 8-9.)

In 1975, Taito released a game called '*Gunfight*'. One thing that made '*Gunfight*' different from the other arcade games was the use of a microprocessor instead of hardwired circuits. A one-channel amplifier provided the game with mono gunshot sounds. (Kent 2001, 64.)

The next important leap in game audio happened when Midway developed and published their hit arcade video game, *'Space Invaders'*, in 1978. *'Space Invaders*' was the first game that introduced continuous music to the players even though it was just a simple descending four-tone loop for the marching aliens in the game. The "music" would also play faster and faster as the game progressed. (Collins 2008, 12.)

In the early 80s, arcade companies upgraded their circuit boards with dedicated sound chips, which helped to create more complex background music and more detailed sound effects. Speech chips, that enabled the programmers to trigger voice samples or sound effects, were also gaining a foothold by the early 80s. By having separate sound chips for sound effects and music, the developers were able to create and implement more complex audio content into their games. (Collins 2008, 12-19.)

A good example of this complex audio content is one of the most popular video games of all time, '*Pacman*'. It was released by Namco in 1980 and it had several, now iconic, sound and music elements, which helped to make game sound a pop-culture phenomenon. (Wolf 2007, 73-74.)

'*Frogger*', an arcade game developed and released by Konami in 1981, introduced a dynamic approach to video game music. It used at least eleven different gameplay tracks, in addition to level-starting and game over themes, which changed according to the player's actions. This type of approach to game music was further enhanced by '*Dig Dug*', an arcade game released by Namco in 1982, where the music would react to player's movement. Whenever the player stopped moving the music stopped playing too. (Collins 2008, 19-20.)

2.2 Rise of home video gaming and chiptunes

It was inevitable that the success of the arcade games would soon move over to the living rooms of the consumers in the form of home video game consoles and personal computers. The first ever home video game console was released in 1972; the Magnavox Odyssey, invented by Ralph Baer. The Magnavox Odyssey did not have a sound module. (Collins 2008, 20.)

The Atari Video Computer System (VCS, later renamed to Atari 2600) was released in 1977 and it would eventually change the way people play video games at home. Atari had developed a chip called the Television Interface Adapter (TIA) specifically designed for sounds and graphics for their console. It brought the first (although minor) improvement to game sound with its two channels, which supported square and sine waveforms. (Collins 2008, 21.)

After the release of Atari 2600, the developers needed to keep players engaged with their games as much as possible, so they started to further develop the audio technology of the home video game consoles. First of the many was Mattel and Coleco, the biggest competitor to Atari on home video gaming, who released their Intellivision system in 1979. The sound generator of Intellivision was capable of producing three-part harmonies. (Marks 2009, 3.)

Atari acted fast and swooped in to the market with their 5200 system in 1982. Atari 5200 had a dedicated audio processor (Pokey), which had four channels and the ability to control the pitch, volume and distortion values of each channel (Marks 2009, 3).

The early 80s marked another milestone for game audio when Commodore released their hugely successful personal computer the Commodore 64 in 1982. When the C64 hit the market it had the most advanced graphic and sound capabilities, so it was no surprise that it appealed to consumers. The C64 had the now famous Sound Interface Device (SID) chip, which provided three separately programmable audio oscillators, four different waveforms per audio oscillator (sawtooth, triangle, pulse, noise) and lots of other options. (Wolf 2007, 78-79.)

Another game audio milestone was reached in 1983, when Musical Instrument Digital Interface (MIDI) was introduced to the market. MIDI revolutionized the way music was composed for video games by allowing the composer to focus solely on composing and not programming (Collins 2008, 50).

The General MIDI (GM) standard was determined in 1991. GM provided a standardized sound bank that allowed a standard MIDI file created on one device to sound exactly the same when played back on another device (Collins 2008, 50).

In 1985, Nintendo introduced their Famicom video game console to the North American markets with a new name, Nintendo Entertainment System (NES). NES was able to provide five channels for monophonic sounds and even offered the ability to make longer loopable pieces of music. One of the most successful games on NES was *'Super*

Mario Bros', which set a new standard for the quality of music and sound effects in video games. (Collins 2008, 25-27.)

Even with the hardware restrictions and limits, the arcade and 8-bit era of video gaming greatly helped game audio to evolve and laid down the cornerstone for future development. Also with the help of some hugely successful games, the sounds in video games were not unnoticed. Now, the game audio industry was more than ready to face the future challenges (Collins 2008, 34-36).

2.3 Game audio comes of age

The 16-bit era of video gaming began when the fourth generation of video game consoles started to appear to the market in the late 1980s. The first almost fully 16-bit home console was the TurboGrafx-16, developed and released by Hudson Soft and NEC in 1987. It had a full 16-bit graphics processor, six channels ready for stereo audio content but only dual 8-bit CPU processors. (Collins 2008, 39.)

The TurboGrafx-16 sold well for a brief moment but when Sega released their fully 16-bit Mega Drive home console in 1988 sales of the TurboGrafx-16 plummeted. The Mega Drive had great audio capabilities. It had a dedicated chip with four mono channels for sound effects and occasional music cues, one PCM 8-bit sample channel and the powerful Yamaha FM synthesis chip. The FM chip enabled developers to create more advanced sound effects and the chip was also flexible to use with its wide range of sounds. (Collins 2008, 38-40.)

The Sega Mega Drive kept selling really well, so Nintendo had to adapt and update their console offering with the Super Nintendo Entertainment System (SNES) in 1991. SNES had a separate three-part audio compartment, which consisted of Sony SPC-700 (an 8-bit processor), 16-bit Sony Digital Signal Processor (DSP) and a 16-bit stereo digital-to-analog converter (DAC). This DSP hardware enabled the developers to utilize eight 32KHz 16bit stereo channels with programmable volume and frequency parameters, a set of audio effects (reverb, filters, pan etc.) and a collection of MIDI instruments. (Collins 2008, 45-47.)

'Actraiser', a video game developed by Quintet, pushed this new audio technology to its limit with the musical score of the game, composed by Yuzo Koshiro. Yuzo Koshiro masterfully utilized his composing and technological skills to create one of the most realistic sounding orchestral game scores of that time (Racketboy 2011).

Various next-generation gaming consoles started to emerge in the early 1990s. One of them was the Sony Playstation, which came out in 1994. The Playstation was one of the first 32-bit video game consoles and it would eventually become the most popular console of the fifth generation, selling over 100 million systems. The sound chip on the Playstation provided 24 channels of CD-quality sound and had built-in support for digital effects such as reverb and looping. (Collins 2008, 68-69.)

This great advancement in audio hardware enabled developers to have high quality music and sound effects in their games. One of the very first games that had a live fully orchestrated score was *'The Lost World: Jurassic Park'* developed by Dreamworks Interactive (Square Enix Music 2007).

The Sony Playstation's main competitors were the Nintendo 64 and the Sega Saturn but they did not even come to close to the mass appeal that the Playstation had. Sega tried to get back into the game in 1998 with their Dreamcast game console but it only sold around 6 million units and was discontinued in 2002 (Collins 2008, 68-69).

Sony broke another record in 2000 when they released Playstation 2 (PS2) to the market. PS2 was the first game console to be able to play DVDs and fully support multichannel surround sound. The sound processor was able to reproduce 48KHz 16 bit audio and also had 48 MIDI channels (Collins 2008, 71).

In 2001, Microsoft entered the game console market with their Xbox console. The Xbox had its own Soundstorm audio processor which fully supported Microsoft's DirectX protocol and multichannel surround sound. It was also the first console to enable saving games to the integrated hard drive and offer online multiplayer gaming through the Xbox Live service. (Collins 2008, 73.)

With the releases of Microsoft's Xbox360 and Sony's Playstation 3 (PS3), the quality of game audio was no longer an issue. The developers could apply audio effects and filters in real-time to 512 channels and they could even stream 7.1 surround sound (Collins 2008, 71).

2.4 Squeezing sound into your pocket

The huge success of arcade and home video gaming gave birth to a new way of playing games, which was handheld gaming. It is not surprising that people who spent hours and hours playing their favourite games at the arcades, wanted to play those same games at home or on the go. Developers realized this and started to manufacture the very first handheld consoles on the market.

Mattel was the first company to enter the handheld console market with the release of '*Auto Race*' in 1976. The early handheld consoles were single game cartridge devices with battery-power and built-in screens. Milton-Bradley advanced the industry in 1979, when they released their Microvision console, which was the first handheld console to use interchangeable game cartridges. (Collins 2008, 75-76.)

None of the early handheld consoles had sound but this changed when Atari released their Lynx console in 1989. Lynx was the first handheld console to have four channel 8-bit sound. Lynx and the other consoles had a moderate success but they did not generate any mass appeal. (Collins 2008, 76.)

The console that made handheld gaming popular was Game Boy, manufactured and released by Nintendo in 1989. The sound chip on Game Boy had four channels, of which the first two were square waves. Third channel was a programmable channel containing 32 4-bit samples. The fourth channel was a 4-bit noise generator. (Collins 2008, 76.) The original Game Boy and its successors (Game Boy Color and Game Boy Advance) combined have sold over 200 million units worldwide (Nintendo 2013).

1997 marked a new leap in handheld gaming when the first game for a mobile phone was developed. The game was '*Snake*', programmed by Taneli Armanto for the Nokia 6110 phone (Collins 2008, 78).

In the early 2000s, countless developers started to either port their old hit games or create new ones for this rising gaming platform. The major phone manufacturers made sure that the mobile phone technology evolved at a steady pace by releasing new phones with advanced features like color screens, 3D graphics and better sound capabilities (Pocketgamer 2008).

The new era for mobile gaming began in 2007 when Apple released their first mobile phone called iPhone. The iPhone had a touch screen, a good set of features and the design was beautiful, so it was no surprise that the consumers wanted to buy it. The iPhone audio specifications stated that the phone had a 20Hz to 20kHz frequency response and it was able to play high quality audio content in various file formats (AAC, MP3, WAV etc.). Of course the alleged frequency response was only viable through headphones and not through the mono speaker. One thing that really advanced the mobile games industry was the launch of App Store in 2008. Now the consumer could easily download applications, games and other items to their phone and use them on the go. (Pocketgamer 2008.)

The App Store rapidly became the go-to source for buying and downloading the latest mobile games. Since Apple's App Store was only useable with their devices (like iPhone, iPad etc.) Google announced the Android Market (later named to Google Play) in October 2008. With the release of Android Market, Android phone users were also able to download applications and games for their phones (Android Developers Blog 2008).

Consumers quickly adapted to these new platforms and started to mass consume mobile games and applications. A good example of the scale of this mass consumption is a game called '*Angry Birds*', developed and released by Rovio Entertainment Ltd. in 2009. In March 2013, Rovio announced that '*Angry Birds*' and its spinoffs (Seasons, Rio, Space and Star Wars) had been downloaded over 1.7 billion times (The Next Web 2013).

3 WHAT IS SOUND AND HOW DOES IT WORK?

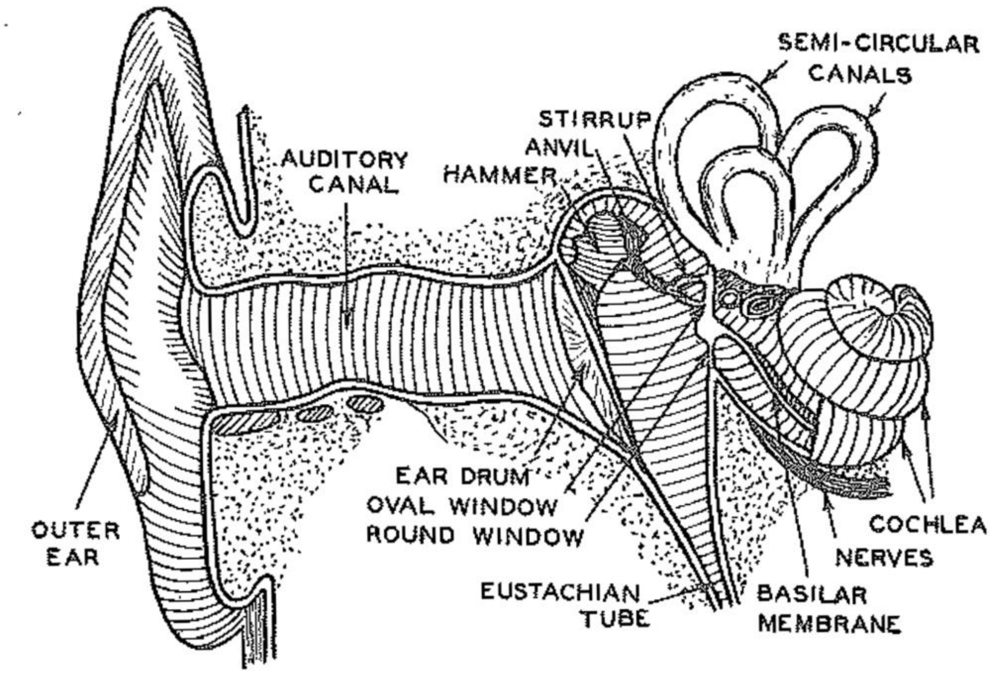
Sound

noun

The sensation produced by stimulation of the organs of hearing by vibrations transmitted through the air or other medium (Dictionary.com).

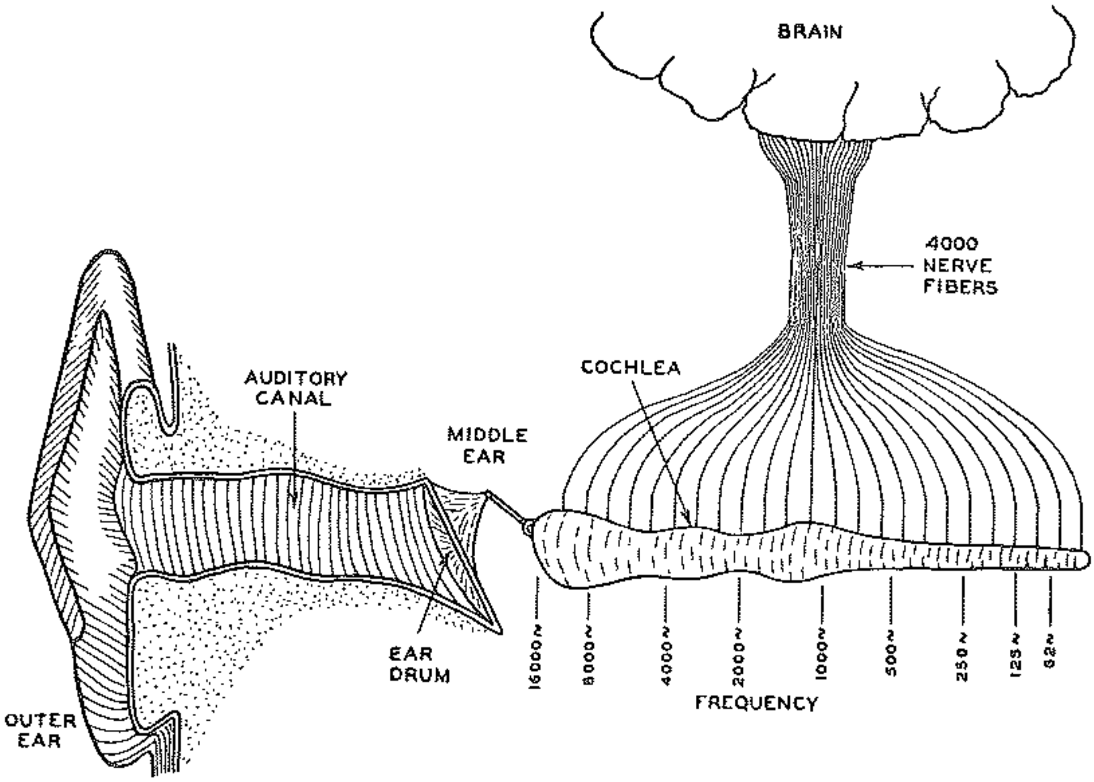
3.1 The Ear and the Perception of Sound

Our perception of sound depends on two things, our ears and our brain. Each ear is a very delicate organ that is able to recognize sound, help us to keep balance and even sense the direction of sound. The normal frequency response range for humans is from 20Hz to 20KHz. These frequencies are not definite, since the upper frequencies usually become inaudible when people get older. (Olson 2003, 248-249.)



PICTURE 1. Sectional view of the human hearing mechanism (Olson 2003, 243)

The human hearing mechanism consists of three sections: the outer ear, the middle ear and the inner ear. The outer ear includes the pinna (auricle), the ear canal and the ear drum (also called tympanic membrane). The middle ear begins after the ear drum and it includes three ear bones: the hammer, the anvil and the stirrup. The inner ear mainly consists of the cochlea and the vestibular system, which helps with the balance. (Olson 2003, 243.)



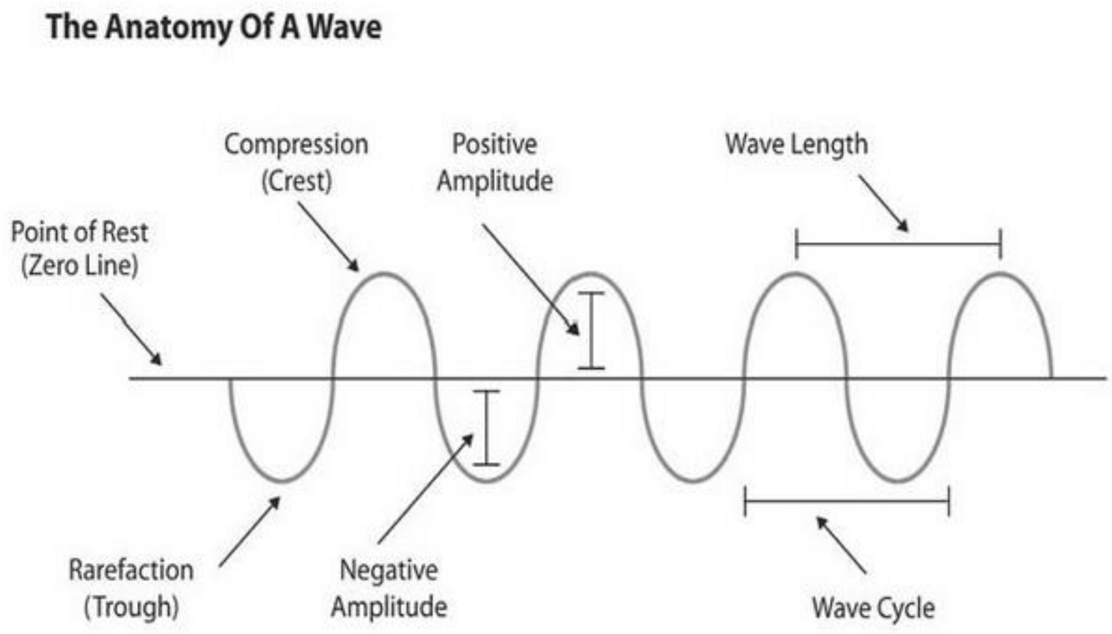
PICTURE 2. Schematic view of the human hearing mechanism (Olson 2003, 245)

The outer part of the ear collects sound. These mechanical vibrations are guided to the ear canal, which amplifies them before they hit the ear drum, causing it to vibrate at the same frequency. Then the mechanical vibrations continue their journey through the hammer, the anvil and the stirrup, and finally hit the oval window. The oval window guides the vibrations into the cochlea where they interact with the frequency sensitive hair cells, which are connected to the nerve fibers. When the vibrations hit the hair cells, they bend and send a nerve impulse to the brain. (Olson 2003, 244-246.)

3.2 Sound Waves

Sound waves are pressure waves caused by the vibrations of an object in conductive mediums such as air, gases, liquids and even some solid matter. For example, when you hit a snare drum, the skin of the drum starts to vibrate, and the vibrations travel through the air, where they can reach the ears of the player (Olson 2003, 3-4).

A sound wave consists of two things: a compression (crest) and a rarefaction (trough). When there is no disturbance in the air pressure, the particles are at point of rest. Whenever there is a disturbance in the air pressure, the particles start to move up and down. Once the pressure increases, the particles move up and this is called a compression. Also when the pressure decreases below the point of rest, the particles move down and this is called a rarefaction. (Olson 2003, 4-7.)



PICTURE 3. The Anatomy Of A Wave (Viers 2008, 8)

3.3 Phase

Phase is a word we use to describe the position of one sound wave relative to another. When the compression and rarefactions of two identical sound waves meet, they are 'in-phase' with each other. When the compression of one sound wave meets the rarefaction of the other sound wave, they are 'out-of-phase' and cancel each other out. The result is usually a thin, weak sound, or sometimes there is no sound at all. (Viers 2008, 9.)

3.4 The Speed of Sound

The speed of a sound wave depends mostly on the medium it is moving through. For example, a sound wave moves seven times faster through water than through air. In normal dry air at a temperature of 20 ºC, the speed of a sound wave is 344 metres per second. There are other factors that affect this speed, like temperature and the altitude relative to the sea level. (Everest 2001, 10.)

3.5 Frequency

Frequency is the number of whole wave cycles (one compression and one rarefaction) per second. For example, a sound that consists of 50 wave cycles per second has a frequency of 50Hz. A sound that consists of 5000 wave cycles per second has a frequency of 5KHz (5000Hz). Frequency is measured in Hertz (Hz), named after the German physicist Heinrich Rudolf Hertz. (Everest 2001, 10.)

Our ears perceive higher frequencies louder than lower frequencies even when they are played at the same volume. Low frequencies are a lot harder to control than high frequencies, because of the longer wavelength and bigger energy content (Viers 2008, 10-11).

3.6 Amplitude

Amplitude is the measurement of the change in the atmospheric pressure caused by the sound waves. The greater the change in atmospheric pressure, the louder the sound. Amplitude is measured in decibels (dB) (Laaksonen 2006, 6).

3.7 Decibels

Decibel is a logarithmic unit for measuring sound and it is named after Alexander Graham Bell. Basically, if you increase the audio source by 3dB, the result is double the power. If you increase the audio source by 6dB, the result is double the volume. When you decrease the audio source by 6dB, the result is half the original volume (Laaksonen 2006, 24-27).

3.8 Acoustics

Acoustics is the science of sound. It is also frequently used to characterize the sound of a room, a hall etc. One of the primary uses of acoustics is to reduce / minimize unwanted sounds and reflections in an environment by treating it with acoustic panels like diffusors, absorbers and bass traps (Viers 2008, 13).

3.9 Reverberation

Reverberation (commonly known as reverb), refers to the way sound waves reflect from various surfaces. As said before, sound waves are energy and they will keep travelling and bouncing from the surfaces until they lose their energy (Laaksonen 2006, 18).

4 WHAT ARE SOUND EFFECTS?

Sound Effect

noun

Any sound, other than music or speech, artificially reproduced to create an effect in a dramatic presentation, as the sound of a storm or a creaking door (Dictionary.com).

Sound effects are one of the main three elements that give a video game its own feel and emotion. They are an integral part of any game and their job is to completely immerse the player into the virtual world of the game. When they are combined with the other two main elements (music and dialogue) the whole gaming experience reaches a completely different level, which is more believable, engaging and fulfilling at the same time (Viers 2008, 2).

In the film industry, the sound effects are usually categorized into five different sections. This categorization method is easily applicable to video games as well since they are becoming more cinematic in style and have similar elements in common with feature films (Viers 2008, 4).

4.1 Hard Effects

Hard effects are the most common sound effects you can hear in a film or a video game. They are usually synced to events or triggers like punches, door slams, gunshots and other sounds that don't need a tightly synced performance (Viers 2008, 5).

4.2 Foley Sound Effects

Foley, named after Jack Donovan Foley, is an art form on its own. Foley's main purpose is to make a scene of a film or a video game more believable and realistic. There are people, called Foley Artists, who usually perform these events live in the Foley-room or studio and sync each cloth movement, footstep or sword movement to the picture (Marks 2009, 272).

4.3 Background Effects

Background effects are sounds that are not necessarily synchronized with the events which are happening in a scene. A more common name for a background effect is ambience and its main purposes are to indicate the setting and set the mood of the film or a video game to the audience / player (Viers 2008, 5).

4.4 Production Elements

Sound effects that fall under the production elements label are usually swishes, zips, clicks and effects created from synthesized sound sources. These sound effects are mainly used for user interface sounds and as source material for sound design effects (Viers 2008, 5).

4.5 Sound Design Effects

Some sound effects are almost impossible to record naturally, so that is why there are people who call themselves Sound Designers. Their job is to make the impossible possible by creating, layering and manipulating sound sources and running them through plug-ins and VST's in their Digital Audio Workstations. Sound design effects can vary from a really small effect to a massive one (Viers 2008, 6).

CASE STUDY

Angry Birds Space: Pig Dipper

PICTURE 4. Angry Birds Space (Google Play)

5 PRE-PRODUCTION

The most important phase of any project is the pre-production phase. It is in this phase where you get to see the project for the first time and start to paint the overall audio sound palette and workflow in your mind. One of the best ways to do this is to try to get as much material as you can from the developer team and go through it thoroughly. Game design documents, early concept art, rough animations and even playable builds can really help you to get inside a project and start the audio design process.

It is also important to point out that every asset the team creates and designs in the pre-production and production phases will still go through lots of iterations and changes.

5.1 Planning

The very first thing I did when I started working on the '*Angry Birds Space: Pig Dipper*' update, was to arrange a meeting with the team that was making it. We set up a Skype conference call and started to talk about the content of the update. The team already had a very rough idea of the theme and the sounds they wanted to have, so we went through them together, then we added more audio assets and eventually created the very first version of the audio design document.

The audio design document is supplementary to the game design document and it generally contains information about the audio design, pipeline, assets and implementation of the audio content. The audio design document helps you to keep things organized and on track throughout the project. It is also useful for the audio programmer (be it you or someone else) when the implementing process begins, so the person knows where everything is meant to go in the game. Here is a brief list of the information that is in the '*Angry Birds Space: Pig Dipper*' audio design document:

- Basic details of the project
- Key contacts in the team
- Dates for the deadlines (alpha, beta, gold etc.)
- Overview of the audio design (style, execution etc.)

- File format and naming guidelines
- Implementation guidelines
- All the audio assets clearly listed and categorized

5.2 Scheduling

Scheduling plays an important role in the development process, especially if the game requires a lot of new audio content. If you have not planned your workload or scheduled your time wisely, then it is likely that you will run into problems during the development process. In a worst case scenario, you will run out of time and you will not be able to deliver all the audio content to the development team. On top of this, the quality of the audio content that you managed to deliver varies greatly, since you did not balance the different work phases and you did not reserve enough time for mixing and mastering.

After creating the first version of the audio design document, I took a closer look at the deadlines and planned the overall schedule for audio. I could see that the update had a lot of new audio content, so I had to schedule my time accordingly.

I started to prioritize and color code all the audio content under three different tags (low, medium, high). Audio assets that had the high priority tag were the most important sounds in the update, because they would be tied closely to the gameplay elements and the players would hear these sounds almost constantly while playing. Audio assets that had the low priority tag were mostly sounds that would support the main gameplay sounds and at the same time maintain the atmosphere / mood of the update. This kind of prioritizing method helped me to plan my recording schedule for the sounds and it also enabled the development team to see what I was doing and when.

After the prioritization, I started to think about the upcoming production phase and the recording / editing process. I scheduled three days for the recording of the sounds, which enabled me to get enough variations for each sound and also do some weird experiments as well. I also reserved a couple of days for the editing of the recorded sounds and a week and a half for the sound design and mixing process.

Next, I focused on planning and scheduling the "post-production" phase. The audio post-production process in games usually revolves around tweaking and testing the audio content in the game. The line between audio production and post-production is somewhat blurry in game development, since it is common for sound designers to already have almost finalized audio assets in the end of the production phase. When the project hits the audio "post-production" phase, almost all of the focus is on testing the audio content in the game, while doing the necessary tweaks either to the audio files or directly to the code.

I wanted to have enough time to properly test the audio content in the game, so I reserved a week for that. During that week, I would repeatedly test all the audio assets in the game and make notes of the changes that needed to be made either to the implementation or to the audio files themselves.

Now that everything was planned and scheduled, it was time to start creating the sound effects, ambiences and other audio content for the update.

6 PRODUCTION

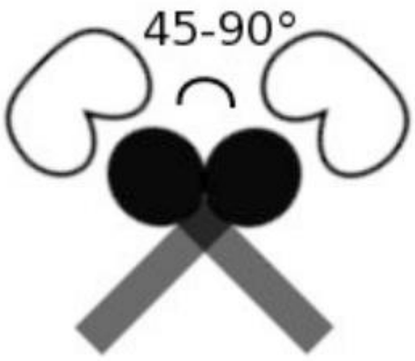
In the production phase, every member of the development team is working closely together to bring the game alive. Programmers begin to write new source code, artists create various art assets, like environments or character models. Sound designers and composers create the audio content for the game in the form of sound effects, ambiences and musical score. Level designers focus on building the levels and game designers continue to develop the overall design of the game.

6.1 Preparations

One thing that I always want to do before the recording process, is to go through the details of what I am recording, how I am going to record them and in what quality. I tend to pull aside sounds that need to be recorded separately at higher sample rates (e.g. audio content that I know will face a lot of pitch processing and time-stretching). By recording these sounds at higher sample rates, I can minimize the appearance of the grain artifacts, while maintaining the quality and accuracy of the recording (The Music of Sound 2010).

Since the update was going to revolve around the water element, I had to set up a water tank into my recording studio. Basically it was just a large plastic box filled up with water. I also needed to protect the microphones from the water splashes, so I applied some protection to them and then I was finally ready to start recording.

For this update, I wanted to record all the sounds both in stereo and mono, so I could experiment a little and have more options on my plate. For stereo recording, I used a pair of Sennheiser MKH8040 cardioid microphones in a 90-degree XY position. For mono recording, I used a regular Neumann TLM 103 large diaphragm microphone. All the microphones were hooked into the Roland Octa-Capture soundcard. Since the sounds would be mostly heard in mobile devices, I decided to record them into Cubase 6.5 in CD quality (44.1KHz, 16 bit). The sounds that would go through the pitch / stretch processing were recorded in 96KHz, 16bit format.



PICTURE 5. XY Technique (Laaksonen 2006, 280)

6.2 Recording

Now it was time to record, so I set up the microphone stands close to the water tank and attached the microphones. I did not want to break my microphones or drop them into the water, so I made sure that all the screws in the stands and microphone holders were properly tightened. As you can see from the picture below, the microphone stands were rather close to the water, so I could capture even the smallest details. I also played it safe and placed the microphones just above the border of the water tank.

PICTURE 6. Microphone positions

I began my recordings by doing some small experiments and testing that the microphones were not in the way or set up in the wrong place. After few test recordings and slight adjustments to the microphones, I was ready to do the real stuff. I will not be able to go through the recording process of all of the sound effects but I will give a couple of examples.

I mentioned earlier about the priority tagging that I did with all of the audio assets in the pre-production phase. Now it was time to revisit this tagging and pick some of the sounds with the high priority tag.

I started with the "bird enter water" and "bird exit water" sounds, since they would probably be the most played sounds in the update. I wanted the "bird enter water" sound effect to sound like a mixture of a bullet and a spinning object hitting the water at the same time. I also wanted it to have this slight ringing effect to give it some extra flavour.

After some trial and error, I found that this glass vase (confiscated from the office kitchen) gave just the right bullet like sound when I hit the bottom of it to the water in a slight angle. For the spinning layer, I just attacked the water with my hand like a cobra while making these spinning movements. Those actually turned out pretty well. With further experimentation I also managed to get some cool ringing sounds out the vase. I always record extra takes for each sound event in the project, so I can easily create variations of the sound effect that sound similar, yet different.

For the "bird exit water" sound, I wanted it to have this water bubble bursting feeling mixed with some water debris. I started to experiment with the objects that I had in the recording room and rather quickly found just the right bubbles. To get the water debris effect, I put my hand deep into the water and then pushed it up to the surface at various speeds.

Next in the recording queue were the "block enter water" and "block exit water" sounds. These sounds would be played when the different blocks fell into the water and bounced back to the surface. I had a feeling that recording these sounds in the small recording room could be messy, but I wanted to at least try. For this I had to place the microphone stands further away from the water tank and apply more protection to the microphones against the likely water splashes.

When everything was ready, I started dropping single oranges into the water tank (so far so good). Then I collected all the oranges and dropped them into the tank at the same time. This is where things got really wet and I had to stop. I could not control the water splashes well enough in this small environment, so I decided to layer the few recorded sounds with sounds from commercial libraries.

I spent the whole three days recording content for the various events in the game, while not forgetting to experiment as well. Most of the stuff that I recorded ended up in the game in one form or another. After three days of recording, I had gathered a lot of great sounding material that I could utilize during the design process of the final audio assets. The recording process was done but the editing process was just around the corner.

6.3 Editing

Sound editing is the first part of the audio polishing process. In the editing phase you usually cut out the unnecessary audio material, trim the useful audio content, add fade ins and outs to them and generally prepare / bounce the audio content for the designing process. When the editing phase is complete, every sound file should be clean of any unnecessary audio material and be ready to be manipulated in various ways (Viers 2008, 171).

When I am editing audio, I try to be as thorough as possible, so I can collect all the useable content and storage it on multiple hard drives. You never know when you will need the audio material again, so it is better to back it all up.

I started my editing process by making copies of all the recordings and backing them up to another hard drive and also to a secure server. It is always a good thing to work with a copy of the audio file instead of the original, so you won't accidentally lose it.

My usual workflow for editing is the following. I open up one of the recordings, separate the useable audio content, delete the unnecessary content and then organize the sounds within the Cubase session. I like to keep things very organized, so I make use of the folder function and group similar files under the same folder. After I have done this, I start the bouncing / naming process. It is important to be as descriptive as possible when naming your audio files. A few distinct keywords in the filename will help you to see or get a feeling of what sounds does the file contain.

When I bounce audio content, I usually make sure to peak normalize each of the audio files to -3 dB. Peak normalization is an automated process that changes the level of each sample in a digital audio signal by the same amount, such that the loudest sample reaches a specified level (Hometracked 2008). This will make the design process easier, so I don't have to spend extra time on adding gain to the audio files.

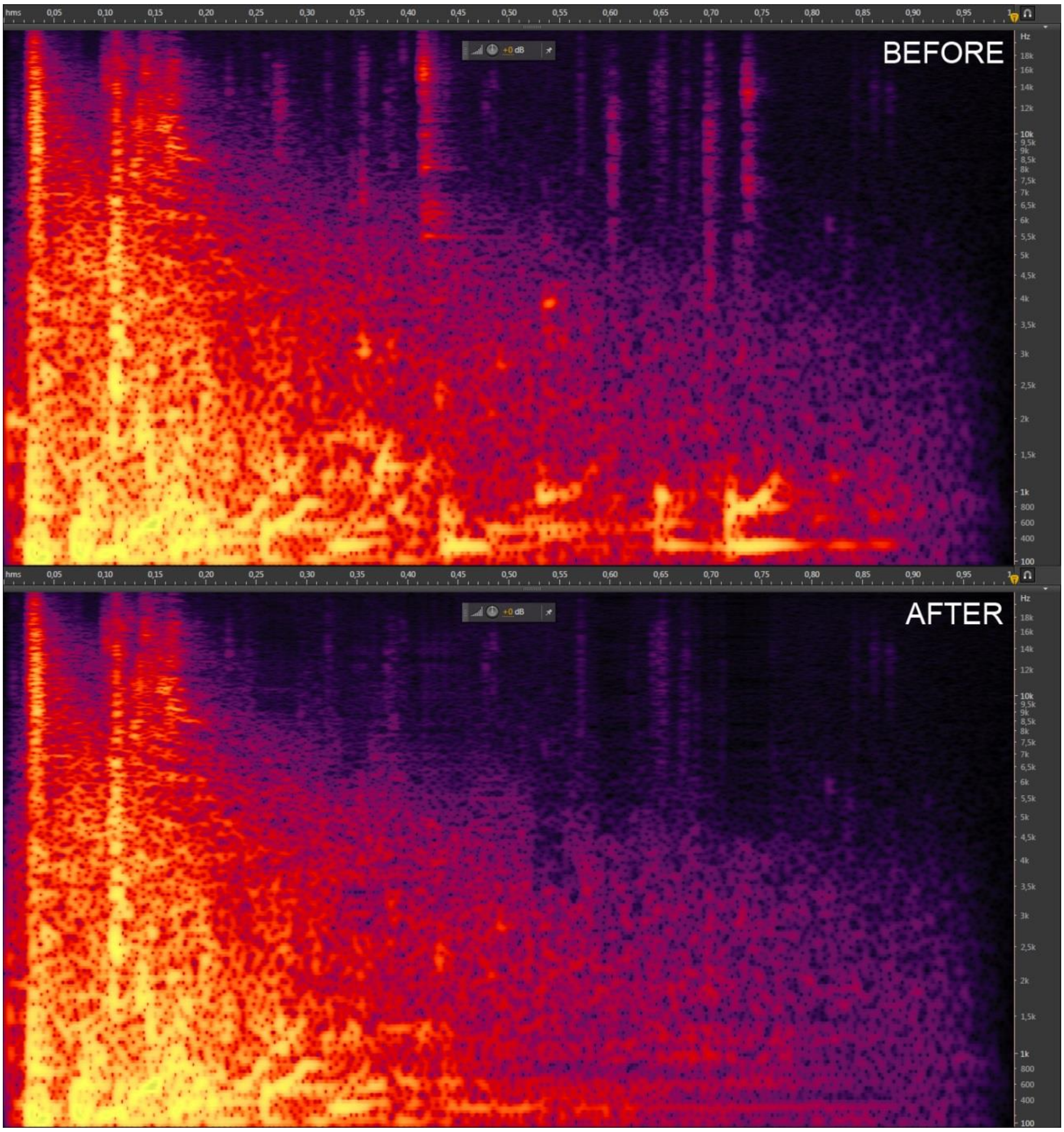
6.4 Noise reduction

Noise reduction is a process where you either tone down or remove unwanted noise, clicks and pops from an audio file. These unwanted artifacts can end up in the recording in various ways. Usually it is due to using a combination of a low quality microphone and a preamp that has a poor self-noise or signal-to-noise ratio (SNR). On the other hand, when you are using a high quality microphone and a preamp, you can still have a problem with small clicks and pops. This usually happens because the microphone and the preamp are so accurate that they pick up everything, even the smallest details (Sound On Sound 1999).

There were some unwanted clicks and pops in the recordings that I did. This was mainly because of the water element. The microphones picked up even the smallest details of the water, which sometimes resulted in unwanted clicks and pops. I also had to do some noise removal to the voice layers of the in-game power ups, because the recordings had some unwanted mouth noises in them.

I used the Spectral Frequency Display (SFD) in Adobe Audition CS6, to do all my noise reduction processes. The best thing in SFD is the fact that you can use it almost like Photoshop. You just select the problematic frequencies or areas with either the marquee or the lasso tool and then apply appropriate actions to it. The picture below shows the noise difference between the untreated "bird enter water" audio file and the treated one.

As you can see from the picture, I fixed some low frequency thumps and pops in the middle and end of the file. There were also a lot of high frequency clicks in the middle and end of the file that I did not want, so I got rid of them as well.



PICTURE 7. Noise reduction using Spectral Frequency Display

7 PRODUCTION - PART II

7.1 Designing and mixing

Sound design is the second part of the audio polishing process. In the designing phase you usually layer multiple sound sources together and manipulate the sounds by running them through various plug-ins and VST's.

Mixing is a process where you combine different audio elements together, while trying to find a good balance between their relative volume and frequency content. To achieve this you usually need tools like equalizers, compressors, limiters, effects (reverb, delay etc.) and of course an acoustically treated space with great speakers (Owsinski 2006, 7-10).

Now that I had all the audio files nicely edited, named and grouped, I could start the designing process. I will not be able to go through the designing and mixing process of all of the sound effects but I will give a couple of examples.

I tend to create separate sessions for each of the audio events, this way my sessions stay clean and organized. I created the session for the "bird enter water" sound and imported the three recorded sounds into the session. I began to design "the flow" of the "bird enter water" sound effect by experimenting with the different layers and slightly alternating the sequence of them. I quickly found the sequence I was after and then did some minor tweaks to the length of the layers by either simply shortening them or adjusting the fade curves.

I was happy with "the flow" of the sound effect, so I started to mix and blend the sounds together. Since the sounds were going to be mostly played on mobile devices, I decided to filter out most of the low frequency content. I made a group channel for the layers and linked the outputs of them to the group. Then I placed an equalizer (EQ) on the group channel and added a high-pass filter with a cutoff frequency around 80hz, so it would attenuate the frequencies below that cutoff value from all the layers. Then I started to treat each of the layers with their own EQ, so I could get rid of the muddy sounding frequencies and other unwanted artifacts.

After the equalizer treatment, it was time to tackle the dynamics of the layers. I placed a compressor plug-in on each of the layers and slightly compressed each of them. I compressed them so that the loud sounds would be reduced by -3 dB while the quieter sounds would not be affected. At this point, the sound effect was starting to sound great and fit the event in the game. One last thing I wanted to do was to tame the dynamics between the layers a bit, so I placed another compressor on the group channel and slightly compressed all of them. Now the "bird enter water" sound effect was ready to be implemented into the game.

The next sound that I was going to design was the background ambience for the water galaxy. The background ambience would be one of the most important sounds in the game, since it is meant to convey the mood and the atmosphere of the new galaxy to the player and help them to immerse themselves in the game. I looked and played the levels of the update and started to think about the elements that I wanted to include. After a short brainstorming session, I had all the elements I wanted, so it was time to start building the ambience.

The backbone for the ambience was going to be the water layer, which consisted of various water ripple recordings I did. I blended them together and pitched them down a bit. The other main sound for the ambience was the drone layer. The drone layer consisted of this bass like drone sound that I created with a program called Paulstretch and then added various effects like reverb, chorus etc. on it. The drone layer for the ambience was meant to loop seamlessly and at the moment it was not, so I had to fix that. The current length of the drone layer was around 2 minutes and I wanted the ambience to be around 1 minute in length, so I had to cut 1 minute out of it. The secret for making seamless loops is pretty easy. I selected the 2 minute long audio file and made my first cut to the file around 1 minute 5 seconds. Then I made another cut to the 1 minute mark. Now I selected the 5 second cut out part and dragged it to the start of the drone layer and dropped it there. Next, I would select both the cut out audio file and the drone layer and cross fade them together with a rather lengthy fade. I made additional tweaks to the fades and now the drone layer was 1 minute in length and it was looping seamlessly.

On top of these layers I added the boat creaking sound, which was actually a creaky wooden chair twisted back and forth. I had to pitch down that layer as well since I needed a heavy creaky sound for the boats of the pigs. Now I had created the basis of the ambience, so all I needed now was some fresh and unique layers to accompany them.

I experimented with pitched down whale calls and to my surprise they fit the ambience pretty well, so I decided to keep them in there. Since all the other background ambiences in AB Space had a subtle melodic element layer in them I incorporated one to this ambience as well. I wanted the instrument to have this wood timbre in it, so I searched through my Kontakt libraries and found what I was looking for. I really loved the smooth and expressive sound that the duduk had, so I started to play sparse notes with it to build the right melody for the ambience. Finally, when the melody was added to the ambience, the mood and the atmosphere of the water galaxy was complete. All I had to do now was to mix the layers together.

The first thing I did for most of the ambience layers was to add reverb to them. They sounded too dry to my liking and it was meant to be a water galaxy after all. I did not touch the drone layer since that was sounding great but I had to apply reverb to the rest of the layers, so they would blend better with the drone layer.

All the layers were now blending nicely together but the stereo image was not there yet. I started to pan some of the layers to both of the sides, while leaving the duduk layer in the middle. I did not want to pan the layers too much to the sides since I wanted to preserve some mono compability as well.

After the panning, I started to EQ each of the tracks. I added the usual group channel and linked all the layers to it. It had the same EQ with the high-pass filter on it but this time I lowered the cutoff frequency to 60hz, so the drone layer would not be attenuated too much. I took out some muddy frequencies from the duduk layer and I also applied a low pass filter with a really gentle cutoff frequency curve around 10KHz to the water layers. This way I could control the high frequency "bubbliness" of the water a bit.

The duduk and water layers were the only ones that needed some compression, so I applied gentle compression to them. This helped me to tame the occasional peaks that occurred especially in the water layers.

The examples above were just brief looks into the whole designing and mixing process of all of the audio content. There were some general treatments that applied to all of the audio files, like the high pass filtering of the low frequencies and the gentle downward compression to tame the occasional loud noises.

7.2 Mastering

Mastering is the third and the last part of the audio polishing process. In the mastering phase you usually look carefully through the audio content and check if there are any problems that show up and then try to fix them (Katz 2003, 11).

I did not have to do much mastering on the audio assets, since I spent a lot of time fine tuning and mixing them. There were of course some occasional frequencies that popped out in some of the sound effects but they were easily fixed with EQ treatment and gentle compression.

7.3 Batch processing / converting

Batch converter is a small program that can process a group of audio files instead of one audio file at a time. Batch converters are built-in to most of the wave editors that are available in the market today. I strongly suggest using a wave editor that has a batch converter built in it, because you will definitely save time when you are working on large game projects. For example, if you have a lot of audio files that need the same reverb treatment, then you can just batch process them as a group, instead of applying the reverb individually to every audio file. (Childs 2006, 109-110.)

I had a lot of audio content in this update that needed to be in two separate audio formats, since the game would be played in both low and high end devices. I used the batch process / converter in Adobe Audition CS6 for the wave file conversion and it was definitely a real time-saver, even though I was not in a hurry or close to a delivery deadline.

7.4 File naming and formats

Before I delivered the audio content to the developer team, I sorted out a couple of important things. First and foremost, I made sure that every audio file was clearly and correctly named and that they did not have any spaces in the filenames. The code usually does not understand spaces in the filenames, so it is better to use either an underscore or a dash instead. Here is one safe way of naming your files:

bird_enter_water_01.wav (where the number indicates the variation of the sound effect)
bird_enter_water_02.wav
bird_enter_water_03.wav

The second thing I did was to check that all the audio files were in the right format and quality. If the files are in another format than what the code tries to look for, then the game will probably have no sound or it will even crash. I used the winLAME software for the conversion and batch encoded the wave files to the mp3 format.

7.5 Delivery and Implementation of sound effects

Implementation is a process where a person (usually the programmer) starts to import the developed content (art, audio etc.) into the game. The creators of these assets usually work closely with the programmer, so that the assets go into their designated places in the correct way. The implementation process is really important for audio, since it controls when the audio files play, how they play and how they behave in the game (Game Sound Design).

For this update, I delivered all the audio files to the programmer and I also gave him a brief implementation guideline document. The guideline document briefly explained where each file should go and how it should be implemented. This was a good way to explain my implementation wishes to the programmer, since a face to face meeting was not possible at that moment.

When the audio files were implemented into the game I started my testing phase. I played the game and checked thoroughly that each of the audio files was playing back at the right pitch, speed and quality. Then I checked that the sounds were in their right place in the game. When I encountered some problems, I made a note of it and sent it to the programmer. When all the files were implemented correctly, I started to critically listen to the balance of the audio assets. At this point I opened up the sound manager file where I could easily tweak both individual and group channel volumes. I would make changes to the volumes of the audio assets, save the sound manager file, commit the file to the server and then start playing the game again, so I could hear the volume changes in action. I would then repeat this procedure again and again until everything sounded great.

8 WHAT ARE THE CHALLENGES IN MOBILE GAME AUDIO AND HOW CAN THEY BE OVERCOME?

The mobile gaming industry has grown very rapidly in the past couple of years. It is now the fastest-growing sector of the whole game industry. New games are being developed at breakneck speeds and they are often rushed to release to the markets. The downside for this is that integral parts of the game are overlooked and quite often it is the sound of the game. In this chapter, I am going to briefly introduce a couple of basic challenges that are often present when working with mobile game audio. I will then suggest ways to overcome these challenges.

8.1 Team size

The most omnipresent challenge facing new game developers in the mobile game industry is team size. It is very common to have just a handful of people working on the game, so often this also means that there is no dedicated audio person in the team. Because of the small team size the roles of the team members are usually generalized as well (Gamasutra 2012).

Small game developers usually have two options for outsourcing audio. Either get the audio from online sources or hire an audio person to do it. Most of the time they rely on the first option and get their audio content through various online sound library sites like FreeSound, SoundSnap and SoundDogs. The downside to this option is that there usually is no critical quality control for the audio content from the team side, so the sounds might not work well with the style and gameplay of the game.

The fact is that all mobile games need high quality audio if they hope to contend in the rapidly growing mobile gaming markets, so the best option is to hire a dedicated audio person for the team. A dedicated audio person works closely with the rest of the team, creates the unique audio content for the game and ensures that the quality bar for audio is being met. This is a more costly solution but if the developer wants their game to be of the highest quality, then they must shine in all aspects of the game, including audio.

If the dedicated audio person is experienced enough e.g. in programming he can enormously help the developer team by taking over some tasks like implementing the audio content into the game, which is normally done by the programmers. This way the programmers can solely focus on their own tasks.

8.2 File size

Mobile games development has always been challenged with disk space, processing power, memory usage and cellular bandwidth. All of these aspects affect how good the game will look and sound. Most of these things are not problems anymore but one still remains- the battle for the limited disk space. When Apple introduced the App Store, they put a 10MB download limit for the applications over the cellular network. This download limit has been raised a couple of times during the years and now it stands at around 50MB. If the application goes over the 50MB limit then you will only be able to download it via Wi-Fi. (Long 2012.)

All of the game development disciplines have had to adapt to these limits and for audio this meant that we started to use compressed audio file formats like mp3 and ogg vorbis. Mp3 (MPEG Audio Layer III) is the most widely used compressed audio file format. It was developed by Fraunhofer-Gesellshaft in 1987 and released to the public in 1995 (Fraunhofer IIS 2013). Ogg Vorbis (commonly used inside the ogg file container) is a compressed audio format developed by Xiph.org foundation and was publicly released in 2000. Ogg is a free, open-source and unpatented alternative to the mp3 file format, which means that people do not have to pay any license fees when utilizing ogg vorbis in their projects (Vorbis.com 2013).

By using compressed audio formats you can reduce the size of an audio file to one tenth of its normal size without losing too much of the sound quality. This will of course help to decrease the overall size of the audio content of the game as well but you still have to figure out the "sweet spot" for the acceptable balance between sound quality and file size. On top of this, it is also a good idea to optimize all of your audio files. Usually this means of cutting out the extra "fat" (unnecessary tails, reducing variations etc.) from the audio content.

Most of the mobile games today make good use of loopable music and ambiences. These files tend to be long so they are usually compressed to mp3 or ogg, so that the file sizes stay small. One of the small challenges with mp3 is the fact that it does not loop seamlessly. When an uncompressed file is converted to mp3, the compression algorithm adds samples to the audio file which make seamless loops impossible. Many engines already offer a way to avoid this issue but if you are not aware of this problem then it can become a time consuming challenge.

8.3 Hardware limitations

Most mobile phones and tablets today have more than enough processing power to playback high quality audio content. To fully hear this high quality audio content, the users have to use good quality headphones or attach a good set of speakers to the mobile device. Unfortunately this kind of behaviour does not happen that often when people are playing mobile games. Instead of using good headphones the user will most likely play the audio content through the small built-in mono speaker or worst case scenario, on mute. The frequency range of this mono speaker is usually between 100Hz and 4KHz, so a big part of the high quality audio content is not heard properly when listening through the small speaker (Long 2012).

These kind of hardware limitations create a new set of challenges for audio professionals. The main challenge is the fact that we have to take in account of the playback method of the audio content while we are creating the sounds. We have to create, treat and balance the audio content properly, so that it sounds great on the small mono speaker and also on the headphones. This usually means that the music and ambiences of the game are created in stereo and most of the sound effects are created in mono although sometimes it is best to use stereo sound effects, so you can really enhance the important parts and events of the game.

9 REFLECTION

The primary goal of this thesis was to provide useful theoretical and practical information for the reader who wants to know more about the field of mobile game audio. The secondary goal was to give the reader an inside look at the actual creation and design process of the audio content for a mobile game by opening up my workflow.

The thesis turned out to be as challenging as I predicted. The most challenging part was to write a brief but informative chapter of the history and development of game audio. To tackle this challenge I referenced at least four different published books and a handful of articles from the internet. Because I wanted to keep this chapter as compact as possible yet still give enough information, I had to carefully pick the most relevant examples and facts. In the end, I think I managed to write a good brief chapter of the history of game audio, its main turning points and technological advancements.

The other challenging aspect of this thesis was to research and write the 'What is Sound and how does it work' chapter. I already had a solid base of knowledge of the science of sound and how it works but this knowledge was mainly acquired from books written in Finnish. So I decided to spend some time to briefly read all the books that I had gathered. The books I had were written either in Finnish or English, so it was a little time consuming to bounce between them while writing the chapter. At first I was a bit hesitant to include this chapter in my thesis but after some thinking I accepted the challenge. While writing this chapter I learned a lot of new things about the science of sound and how our ears perceive the aural information. I also greatly expanded my English language terminology in the fields of sound and game audio.

After I had written the more theoretical parts of the thesis I could breathe a little easier and start writing the more practical part. My plan was to break down my workflow for the Angry Birds Space: Pig Dipper update and reveal all the right and wrong choices I made during the process. This part was a lot easier to write since the project was still fresh in my mind and I could use my own words instead of just referencing a pile of books. I tried to be as thorough as possible when explaining the different stages of my design process. At some point I wanted to include more examples in this chapter, so I could explain more of the audio content to the reader. I quickly realised that I would have to limit the examples to three different sounds because otherwise the length of the-sis would grow into unwanted proportions. I am quite happy of the results of this chap-ter. I managed to explain my workflow and present some of the techniques that I use daily to design my sounds.

The last main part of the thesis was to present some basic challenges that are commonly present in the field of mobile game audio and then try to suggest solutions for those challenges. I wanted to keep this chapter as compact as possible and focus on the most frequently met challenges. The information presented in this chapter was a mixture of my own experiences and articles written by other audio professionals. In the end, I man-aged to bring up the most basic challenges in mobile game audio and then present good practical solutions that will hopefully help the reader of this thesis to overcome those challenges.

In conclusion, I have to say that I managed to successfully reach the goals that I set for myself in the beginning. It took a lot of effort and time to reach this stage but I am hap-py with the result. I am also happy that I now have a deeper knowledge of the history of game audio and the scientific side of sound. Overall, I think this thesis has a good amount of theoretical and practical information that will hopefully encourage the reader to explore more of the exciting world of game audio, create high quality audio content for mobile games, evaluate their current workflow and evangelize the importance of audio in video games.

REFERENCES

Android Developers Blog. 2008. Android Market: Now available for users. Accessed 28.2.2013. http://android-developers.blogspot.fi/2008/10/android-market-now-available-for-users.html

Childs, G. 2006. Creating Music and Sound for Games. Course Technology Inc.

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

Dictionary.com. Sound Effect. Dictionary.com unabridged. Random House, Inc. Accessed 20.1.2013. http://dictionary.reference.com/browse/sound+effect

Everest, F. 2001. The Master Handbook of Acoustics. New York: McGraw-Hill Companies, Inc.

Fraunhofer IIS. 2013. The Story of mp3. Accessed 8.4.2013. http://www.mp3-history.com/en/the_story_of_mp3.html

Gamasutra. 2012. AAA-Lite Audio: Design Challenges and Methodologies to Bring Console-Quality Audio to Browser and Mobile Games. Accessed 1.4.2013. http://www.gamasutra.com/view/feature/134761

Game Sound Design. Getting Your Sound Into the Game: A Process for Implementing Sound Design. Accessed 26.3.2013. http://gamesounddesign.com/GettingYourSoundIntoTheGame.html

Google Play. Angry Birds Space. Accessed 20.1.2013. https://play.google.com/store/apps/details?id=com.rovio.angrybirdsspace.ads

Hometracked. 2008. 10 Myths About Normalization. Accessed 24.3.2013. http://www.hometracked.com/2008/04/20/10-myths-about-normalization/

Katz, B. 2003. Mastering Audio, the Art and the Science. Focal Press.

Kent, S. 2001. The Ultimate History of Video Games: From Pong to Pokemon and beyond. Three Rivers Press.

Laaksonen, J. 2006. Äänityön kivijalka. Helsinki: Idemco Oy.

Long, B. 2012. The Insider's Guide to Music & Sound for Mobile Games. Amazon Digital Services, Inc.

Marks, A. 2009. The Complete Guide to Game Audio, For Composers, Musicians, Sound Designers and Game Developers, 2nd Edition. Focal Press.

Nintendo. 2013. Consolidated Sales Transition by Region. Accessed 26.02.2013. http://www.nintendo.co.jp/ir/library/historical_data/pdf/consolidated_sales_e1212.pdf

Olson, H. 2003. Music, Physics and Engineering. Dover Publications Inc.

Owsinski, B. 2006. The Mixing Engineer's Handbook. Vallejo: Intertec Publishing.

Pocketgamer. 2008. A Brief History of Mobile Games: In the beginning, there was Snake. Accessed 24.2.2013. http://www.pocketgamer.biz/r/PG.Biz/A+Brief+History+of+Mobile+Games/feature.asp?c=10619

Racketboy. Games that pushed the limits of the Super Nintendo (SNES). Accessed 24.2.2013. http://www.racketboy.com/retro/super-nintendo-snes-games-that-pushed-the-limits-graphics-soun

Sound On Sound. 1999. 20 Tips On Reducing Noise. Accessed 8.4.2013. http://www.soundonsound.com/sos/mar99/articles/20tips.htm

Square Enix Music. The Lost World: Jurassic Park - Original soundtrack review. Accessed 24.2.2013. http://www.squareenixmusic.com/reviews/simonelchlepp/lostworld.shtml

The Music of Sound. 2010. Why use high sample rates? Accessed 17.3.2013. http://www.musicofsound.co.nz/blog/why-use-high-sample-rates

The Next Web. 2013. Hitting 1.7bn downloads, Rovio says Angry Birds Toons will be available on its games starting March 17. Accessed 19.3.2013. http://thenextweb.com/media/2013/03/11/hitting-1-7-billion-downloads-rovio-announces-that-angry-birds-toons-will-be-available-through-its-games-starting-march-17/

Viers, R. 2008. The Sound Effects Bible: How to Create and Record Hollywood Style Sound Effects. Michael Wiese Productions.

Vorbis.com. 2013. Frequently Asked Questions. Accessed 8.4.2013. http://www.vorbis.com/faq/

Wired. 2013. Mobile Kills the Console But Advances the Gaming Industry. Accessed 4.4.2013. http://www.wired.com/opinion/2013/01/how-mobile-kills-the-console-but-advances-the-gaming-industry/

Wolf, M. 2007. The Video Game Explosion: A History from Pong to Playstation and Beyond. Greenwood Press.