St. Pölten University of Applied Sciences

/medien & digitale technologien

/fh/// st.pölten

Music in Games

How non-linear music composing differs from linear music composing

Research Paper

Bachelor course on Media Technology
at St. Pölten University of Applied Sciences

by:

Thomas Maislinger

MT-161125

Supervising tutor: Dipl. Ing. FH – Prof. Andreas Büchele

[Vienna, 31.08.2018]

Declaration

- The attached research paper is my own, original work undertaken in partial fulfillment of my degree.

- I have made no use of sources, materials or assistance other than those which habe been openly and fully acknowledged in the text. If any part of another person's work has been quoted, this either appears in inverted commas or (if beyond a few lines) is indented.

- Any direct quotation or source of ideas has been identified in the text by author, date, and page number(s) immediately after such an item, and full details are provided in a reference list at the end of the text.

- I understand that any breach of the fair practice regulations may result in a mark of zero for this research paper and that it could also involve other repercussions.

Vienna, 31.08.2018

.....	.....
Place, Date	Signature

Abstract

This thesis analyzes the differences between composing music for linear and non linear media. In this case video games. Starting with the history of game audio, it will give a short summary about well known composers in the industry. After that it is all about technological developments and how the work environment for composers and sound designers has changed over the past few years.

The motivation in writing this bachelor thesis is to esteem the work that is behind composing music for video games, so that consumers can appreciate that as well by reading it and maybe also understand a little better why it is harder to write music for scenarios that you cannot predict.

Table of contents

1 Introduction

Each story telling medium has its prime time. It is the only way to really understand the message of the author, until a new technological development shines on the market that includes the predecessor and creates something new and more immersive while the origin of the story telling art slowly fades away with the people remembering it.

In ancient times, before the written word was a common skill to use and decrypt, no matter which social status someone belonged to, stories where told by a story teller supported by the crackling noise of a campfire. By skipping literature and theatre on our little time travel, we arrive in the 19th century where radio was developed. It was again THE medium that everybody listened to when someone wanted to stay informed or listen to a good night story. Combined with photography it reached film and television and so on and so forth…

Long story short. Crackling campfire, interfering frequencies,… all those sounds support the emotional impact of a story in a way that nothing else could and the strongest of them is called music.

In times of virtual reality and GPUs (getting stronger and faster and affordable for the normal private user, it is pretty self-explaining that in a few years media consumption as we know it will change drastically. Human beings have always chosen the more immersive media over the other one less immersive and video games can offer something that common story telling formats cannot. The user writes his own story. Sure the surrounding is set, what happens to the protagonist though, that’s in the hands of the consumer and that’s an advantage that other media can’t compete up with.

1 Introduction

Good looking graphics, as well as a week structured story-lined are main parts of a video game, but when it comes to feel the story it comes to music and sound effects.

Titles like “Journey” (2012), “Shadow of the Colossus” (2005), “Zelda – Ocarina of Time” (1998) and a lot of others have already proofed that even without any dialogue, or in the case of Zelda just little dialogue, music is the biggest indicator that brings moving pictures to life and let the recipient feel what the author wants him or her to feel.

This bachelor thesis should give an insight in how non-linear music composing, like it is mostly used nowadays for modern video-game productions, differs from common linear media music producing. As (Xiaoqing Fu, 2015) quotes from (Eladhari, Nieuwdorp, & Fridenfalk, 2006), (Collins & Collins, 2008) & (Cunningham, Grout, & Hebblewhite, 2006) in his article “The Influence of Background Music of Video Games on Immersion” :

“Although we usually call computer games as “video” games, the audio also plays an important part in video games.” (Xiaoqing Fu, 2015)

And not only that. Music is one of the biggest indictors for the so called immersion of a player into a game. Experts like (Vorderer & Bryant, 2006) as well as (Xiaoqing Fu, 2015) agree that background music, sound effects, narrators, the sound design in general enhances feelings, influences the quality of how the players react to certain situations in-game or how it is just supporting the aesthetic and style. Or in other words, it is necessary for a great game experience. (Xiaoqing Fu, 2015) This thesis will also offer a short summary about the history of video game music and its ever shining luminary Koji Kondo.

The man who wrote the music for the Super Mario Bros. and the Legend of Zelda series, games that shaped the world of video games like no others, amongst a lot of other titles by Nintendo. (Schartmann, 2015)

2 History – When video game music became relevant

After the boom of arcade game halls (also arcades = places where people could play video games on machines for a quarter each turn) the “Atari Video Computer System” (VCS) was sold for gaming at home in the late 1970s along with Space Invaders, Odyssey2 and Atari’s Asteroids. Those games were only playable in the arcades before and now consumers could play them on their couches with family and friends. (McDonald, 2005)

“To skeptics, however, arcades could be deeply annoying places, in part because of their noise and hectoring sounds, which served little purpose beyond “hailing” passers-by.” (Hsu, 2015)

The music itself was first generated on synthetic computer chips. Those had only the possibilities of low bit generated sounds (= low bit means, that there were only few possibilities of pitching the sound or even having different types of it). These famous robotic sounding beeps gamers still remember with a grin from Atari’s Pong. This way of composing has become a recognizable style called “chiptune”. (The 405, 2014) Along with the 1980s, 2 games arrived at the market that would change the history of video games. Konami’s Pac-Man and Nintendo’s Donkey Kong featuring the Jumpman (later on better known as Super Mario). During that time the era of 8 bit music began. Until today (2018) 8 bit music is still a style of music producing which not only reminds us of hanging out in the arcades but is mostly used to provide the feeling players have while experiencing a pixelated game. The “bitcrusher” is a common effect in modern DAWs (Digital Audio Workstation) for creating 8 bit music, or even using it only for specific parts in a composition, for instance the cymbals of a drum set (hi hat, ride, crash, etc.). (Collins & Collins, 2008; Hsu, 2015; McDonald, 2005; National Public Radio, 2008; Purves, 2013; Schartmann, 2015; The 405, 2014)

2.1 Koji Kondo

In 1985 Koji Kondo shows the world how much influence game music can have with his first master piece in Super Mario Bros. Andrew Shartmann even wrote a quote from Tommy Tallarico in the foreword of his book about Kojo Kondo's Super Mario Bros. Soundtrack:

> "Koji Kondo is by far the most famous video game composer. In fact, he's probably one of the most heard composers of all time. How many hundreds of millions of people across the globe have hummed the original Super Mario Bros. music at some point over the past 30 years? And Mario was just the beginning! Most incredible of all, though, is that he was able to create such masterpieces within the extreme restrictions of the NES hardware. The original console had just three monophonic tone channels and one that could produce white noise. But Kondo made it work. The reality is that people will be humming his tunes long after we're gone. And 's decision to dedicate their first book on video game music to his work is just further confirmation of how widespread Kondo's influence has become. There truly is a living legend among us, a legend whom I'm proud to call a friend. So let's raise our controllers in honor of him—the man who put video game music on the map." (Schartmann, 2015)

The fact that a whole book is written for one soundtrack as well as his repeating masterpieces in the last few years pretty much show, how famous Koji Kondo has become and how much he has done for the community, art and business of video game music.

2.2 Other successful composers

With the years and continuously evolving game machines the possibilities for game music composers enhanced likewise. 2 big contributors to the industry in the early 2000s were Martin O`Donnel and Michael Salvatori who created one of the first famous orchestral soundtracks for the game Halo from Bungie. (The 405, 2014)

Robert Purves created a list of a few composers and game titles up to 2012 worth mentioning :

- Koji Kondo - many Super Mario Brothers and The Legend of Zelda series
- Nobuo Uematsu - Final Fantasy series
- Masato Nakamura - the early Sonic The Hedgehog games
- […]
- Martin O'Donnell - Myth, Myth II, Oni, and the Halo Series (first games job was sound design for Riven)
- Jack Wall - Myst III: Exile, Myst IV: Revelation, Mass Effect 1 & 2, Jade Empire and Rise of the Kasai
- […]
- Mark Morgan - Fallout series, Planescape: Torment, Zork series, Dark Seed 2, Descent 2
- […]
- Jeremy Soule - Total Annihilation, Elder Scrolls titles: Morrowind and Oblivion
- […]
- Neal Acree - World of Warcraft series, Starcraft series, with "World of Warcraft: Mists of Pandaria" nominated in the Annual Games Music Awards 2012
- Inon Zur - Prince of Persia series, various Star Trek and Baldur's Gate titles, Crysis, Dragon Age: Origins
- […]

- Jeff Rona - one of several composers on "God of War 3" (the others were Gerard Marino, Ron Fish, Cris Velasco and Mike Reagan)
- […]
- Hans Zimmer - themes for Crysis 2 (music by Borislav Slavov, Tilman Sillescu, Hans Zimmer and Lorne Balfe)

(Purves, 2013)

3 Game music today

3.1 Immersion

In times of VR (Virtual Reality), AR (Augmented Reality) and computer systems that are so powerful that video game graphics seem like real recorded footage, the complexity of interactive audio increases as well as the aspect of immersion.(Rumsey, 2009; Schmidt, 2003)

(Xiaoqing Fu, 2015) quotes the definition of Immersion from (Wood, Griffiths, & Parke, 2007) in his article "The Influence of Background Music of Video Games on Immersion" as followed:

"Immersion as understood here is the sense of being "in a game" where a person's thoughts, attention and goals are all focused in and around the game" (Wood et al., 2007) (Xiaoqing Fu, 2015)

Even if critics agree that immersion is necessary for a good or great game experience, it is also discussed a lot whether it causes danger of addiction or not. (Jennett et al., 2008; Xiaoqing Fu, 2015)

3.2 Linearity and non – linearity

The main problem or more likely difficulty for video game music composers nowadays is basically the fact that games are detaching themselves more and more from a linear medium, meaning that as interactivity increases the composers as well as the programmers do not know what the next choice of the gamer will be. Will he/she decide to go into a dark cave or continue walking on a peaceful field and if he or she walks into the cave will it rather be a romantic moment or really

scary and if he or she continues on the field will he or she kill fellow wanders or just talk with them etc. Music and sound effects have to react to all those choices in a way that the immersion doesn't get lost, so that the player doesn't feel confused so to speak. (Rumsey, 2009, 2010, 2015; Schmidt, 2003; Xiaoqing Fu, 2015)

So how are music composers able to adapt to the fact that music has to be programmed rather than written?

In 2003 Brian Schmidt divides the process of interactive sound composing in his arcticle "Interactive Mixing of Game Audio" in three subgroups:

- Spotting
  - Determining when sound events need to occur
- Sound Creation
  - Creating the actual sound assets to be used
- Mixing
  - Specifying how and when the sound elements are presented to the user in context

(Schmidt, 2003)

3.2.1 Spotting

Unlike in traditional linear media where the sound designer sets keyframes according to the video timeline, mostly frame by frame, game sound designers do not have that opportunity of knowing when a specific action is going to happen. That's why actions in game trigger certain pre composed sound effects and music parts. But not only that. The program also have to add parameters according to the action that is taken place in the game. Brian Schmidt explains that

phenomenon with a sword fight. As the player fights against another player or even an AI (Artificial Intelligence) the swords hit each other or the opponent in different angles, velocities and spots, causing other shapes of the original sound. Therefore audio effects have to be attached to the pre-recorded sounds like EQs (Equalizers), compression, bit crushers and so on and so forth, to simulate the laws of physics according to the actions and give the player a more immersive game experience. (Schmidt, 2003)

3.2.2 Sound creation

This is the part where the recording starts. When all the spotted sounds are recorded, the sound designer has to set all the parameters for the different possibilities. (Schmidt, 2003)

Well, that was the standard in 2003. So dear reader, no need to be desperate, things are far more comfortable than back there.

3.2.3 Interactive mixing and audio middleware

The basic idea of creating new possibilities for game sound designers were primary to skip the long waiting times (hours to days) of communicating with the programmers. If the sound designer had to make a change he or she had to wait until that change had been tested by the programmers and accepted or declined. So the whole process had to be started over and over again until the results were acceptable. Because even if a character spoke to the protagonist in-game and the sound designer for instance decided that as he moves away from the speaking person, lets say each meter/foot the volume decreases half a db (decibel = logarithmic unit to measure sound level), the composer wouldn't have got any chance to see or actually hear that parameter in action, he or she had to wait until the programmers told him or her that the estimation was correct or not. In order to

change that never-ending process the idea was to create a software that would allow composers to make changes while playing the game. The first try for such a software has been a special simulation program were the sound designer had a low graphic version of the game with a customized GUI (graphical user interface) that allowed first ways of real interactive mixing. The composer had the chance to check the parametrical changes he or she made for the specific parts in the game but still not change them while actually playing. It still was just a simulation. (Rumsey, 2009; Schmidt, 2003)

Then came middleware software tools came like FMOD and WWISE which can exactly do that. They allow sound designers to structure and implement interactive audio and music content for games and support dynamic mixing in a way that couldn't be done before. It is to this day a common industrial standard, because changes of sound, mixing, creating new sounds,… all that can be done in real-time. While connected with the server of the programmers they only adapt the changes that have been made by the sound designers sometimes without even recognizing. Gone are the days of enormous delays between the communication of the composers and programmers. The game can now be created nearly simultaneously. (Rumsey, 2010)

3.2.4 FMOD

> "FMOD (www.fmod.org) is a set of middleware and sound design tools for game systems developed by Firelight Technologies. There are currently two offerings: FMOD Ex and Designer. Ex is a programmer's API (application programming interface) toolkit. Essentially it provides an audio engine for game platforms that can interface with other parts of the game software using calls to specific functions and game-play data. It is said to be available for a range of platforms including Mac, Windows, Solaris, and Linux computing systems, as well as Sony, Microsoft, and Nintendo game platforms. A DSP software architecture is offered that consists of a 32-bit floating-point mixing engine with flexible multichannel routing and submixing. Various different multichannel surround formats are supported,

including matrixed Dolby Prologic, and a selection of effects are provided such as echo, chorus, and reverb. In addition to a number of different output encoding and streaming formats, FMOD can handle a number of simultaneous sample files in formats such as MPEG, ADPCM, and XMA (the native compressed sound file format used in the Microsoft XBox 360, based on the WMA format). These can be looped or sequenced alongside uncompressed samples without gaps or other artifacts. Alternative output codecs can be added to those provided using a plug-in structure. A sophisticated 3-D sound processing engine is included, which takes user- or object-position data from the game and renders it in spatial terms using appropriate panning, level, filtering, and Doppler shift processing. Obstruction and occlusion effects can be simulated based on polygon scenes, and there is an HRTFbased spatial mixer to add binaurally rendered sounds for use on headphones. Designer enables the sound designer to access the low-level tools in FMOD without needing programming knowledge. An interactive music tool allows the creation of complex multilayered scenes with prespecified responses to game play variables, including beat matching and nonlinear sequencing. It's also possible to monitor the platform's CPU usage and the signal path through the system, in order to optimize the resources required for game audio on different platforms. Content for multiple target platforms and language markets can be handled using wave-bank management." (Rumsey, 2010)

3.2.5 WWISE

"WWISE from Audiokinetic (www. audiokinetic.com), combines an audio authoring tool and a sound engine, claiming to offer a complete "audio pipeline" solution. The audio engine offers a range of effects and plug-ins, including 5.1-channel reverbs, and, like FMOD, it simulates occlusion and obstruction in polygon-based scenes. There is also support for multiple listeners, such as might be needed in splitscreen or multiplayer contexts. It is also possible to test prototype projects by simulating their performance

on different game platforms. There is an interface between the audio engine and third-party “world-building” applications used for game construction, such as Unreal Editor and Maya. The authoring system offers a wide range of different dynamic mixing and audio effects options, as well as a dynamic dialogue system that aims to reduce the overall memory consumption for game audio, based on a set of rules, while also enabling dialogue stitching for different applications. For 3-D sound positioning, sounds can be attached to objects in the scene so that their propagation characteristics can be adapted as they move around. This can be combined with distance-based attenuation and filtering curves as well as source-directivity characteristics. Sounds can be played back according to a set of predefined patterns, such as randomized, sequenced, switched (based on game options), or blended. Groups of sounds can be combined in hierarchical structures that allow their properties to be controlled together, such as in relation to a particular actor (which is essentially the activating event or object representation in a game). sounds that are likely to be inaudible can be managed so that they don’t use up valuable processing resources, and the relative importance of sounds can be controlled so that their priority for available resources can be managed. WWISE is currently implemented for a number of platforms including Windows, Mac, Xbox 360, Sony Playstation, and Nintendo Wii.”

(Rumsey, 2010)

3.3 Looping

Which doesn't mean that there is no linear sound composing any more. In a modern game which consists of normally approximately 30-40 hours of gameplay a lot of loop-music(= you do not recognize that moment when the music ends and repeats itself) is required. This loop generated music is mostly used for journeys (calm background music) or menus. Also normally each stage, level or village has its own composed music theme which enhances the psychological aspect of recognition. These loops or themes have to be composed in a usual linear way. Although when we are talking about a 30-40 hours game usually only 60-90 minutes of music is required, which doesn't seem much, but the process of composing is way different. It has to work over and over again without getting boring or annoying. (Rumsey, 2010; Wood et al., 2007; Xiaoqing Fu, 2015)

Andrew Schartmann writes in his book "Koji Kondo's Super Mario Bros. Soundtrack (33 1/3)" about the fact that Kondo himself uses this technique of repeating his music over and over again. Kondo talks about how he listens to his new composed music for a Yoshi, Mario or Zelda theme sometimes a whole day. Hours and hours of the same 40 seconds only to check if it is getting annoying or not.(Schartmann, 2015)

Also the technical aspect of looping is way more complex than with single track based music that has a beginning and an end. Imagine if there are 50 audio tracks running simultaneously. That's enough to work for the CPU and GPU already, but when each of those tracks have to repeat at certain point and especially not at the same time this can lead up to in-game latencies and destroy the aspect of immersion. Also if we are talking about a stage-level based game, or in the industry better known as side scroller, where you have to start from the beginning when you die, it is most likely that the player is hearing the beginning of the loop created music more often than the end. To avoid causes like this the programmers have to come up with an algorithm that starts randomly in between the bars and the composer has to write the music in a way that it doesn't actually matter where the track starts and ends. There are so much aspects to consider while working with game music. Normally the technical leader of a project suggests an audio engine

and a DAW that is being used so that the sound designers get a better idea of the limitations and resources available.(Rumsey, 2010)

3.4 DAWs and standardization in the audio-game industry

A huge time aspect of composing music with DAWs and audio middleware is still the fact, that there have to be at least 2 different softwares in use and especially that once music is composed, it still has to be programmed. To combine programming and composition and kill 2 birds with one stone so to speak, FMOD began in the year 2013 to embed more production-oriented tools and make it more "DAW like". The goal was not only to change the GUI to look like a DAW but literally to function properly as such. But then Ableton Live and Max MSP came. Ableton has always been a DAW normally exclusively used by electronic music producers and DJs and rather avoided by common music producers, because of its focus on non-linearity. Wait a second… we are talking so much about non-linear music production and Ableton is being avoided? Why that you ask? Well it is the same with every other program or actually everything that is new and different. It is being avoided until someone realizes that it actually can be useful. The same with Ableton Live. The fact that it has both options, a linear and a non-linear workflow, came in pretty handy for game music producers over the last few years. That and the fact that you can work with Max-MSP directly in the DAW made it quite unique. Max MSP is an object oriented programming environment for creating new audio effects, instruments, manipulating nearly everything with MIDI so on and so forth. It has the possibilities to control each parameter inside Ableton Live, triggered by nearly anything the user wants it to be triggered from.(YOUNG, 2013) Doesn't matter if it is the amount of red color value from the computers webcam that enhances the reverb from a specific instrument in a song, or it measures the mouse speed and enhances the bpm (beats per minute) of a song, or even work with facial expressions. There are nearly no boundaries for creative composers/programmers while working within Max-MSP.(Rumsey, 2015; YOUNG, 2013)

So again, it is the combination of producing and programming in one software. As DAWs are structured with a higher complexity, it made more sense to transform a DAW like Ableton into a audio middleware software than the other way around like FMOD and WWISE did. But all this converting, transferring, exporting, importing, implementing and other steps are still way too time consuming for the industry. Standardization is required. (YOUNG, 2013)

A first great step in that direction was made by IASIG (Interactive Audo Special Interest Group) with their shell iXMF/Interop format. Based on the existing architecture of XMF (eXtensible Music Format = meta file format for gathering collections of data resources) iXMF should be the interactive version of XMF, meaning that audio implementation instructions as well as general information could be bundled in one file by the audio designers. The goal of IASIG was to create a standard format for interactive music composing and as they propose:

> "This new file format will put artistic control into the hands of the artists, keep programmers from having to make artistic decisions, eliminate rework for porting to new platforms, and reduce production time, cost, and stress."(IASIG, 2009)
>
> (YOUNG, 2013)

While decreasing storage needs and increasing flexibility, the iXMF allows sound designers to work on audio files in real time, instead of preparing and exporting them with complex audio effects which enhance file sizes and cant be reworked when put into the algorithm. This allows a lot of new possibilities. By combining the potential of the iXMF/Interop and the open programming within DAWs like Ableton Live, the days of standardization within the nonlinear sound and music production industry seem to be neigh. The possibilities would be literally game changing. Sound designers could take a DAW's whole multi-track session, with all effect parameters like EQs, compressors, VST-Plug in settings, cuts, volume and panorama - leveling, even automations and save it into the iXMF/Interop format. This file could be opened in a audio middleware software which is mapped by the

game engine with all those parameters and the composers could continue working on it as if it would be the DAW itself. As well as all the parameters can now be triggered, controlled and changed anytime by in-game events.(YOUNG, 2013)

As David M. Young writes in his article “The Future of Adaptive Game Music: The Continuing Evolution of Dynamic Music Systems in Video Games” :

> “The game itself could “ride the faders” of the different stem mixes, DSP parameters, and even the master transport. This could allow for a significantly adaptive and immersive score, without having to port different score elements into a new middleware session or proprietary adaptive music engine. In this scenario, the strain of the transfer from production to implementation, composer to programmer, would be greatly alleviated – all by exporting a DAW session in the correct format.” (YOUNG, 2013)

3.5 Generative music systems

The surrounding is set and the computers are fast enough to process all that new technology, which means the paradise for game music composers is set right? Well, not quite. As the work of machines is way cheaper than human’s, the way of thinking in the game music industry is also going in that direction nowadays. Meaning that by the constant evolving technologies of AI (Artificial Intelligence) there are already algorithms in progress that should replace human composers. Not entirely, but nearly. The next level in the industry is called generative music systems. The idea of generative music systems is not only (as discussed earlier) saving time and money, but also that music can be composed while the game is evolving. On the fly, so to speak. While the mood of the game changes, the AI should react to those mood or scene changes and compose the music according to that. Which also means that such a generative music system only needs little, up to no storage because there is no music file until it is composed in real time,

triggered by the actions of the game itself. According to Francis Rumsey in his article “Game Audio - Generative music, emotions, and realism” one of the first breakthroughs in terms of generative music systems, discussed by Alper Gungormusler and his colleagues from Trinity College Dublin, is the adaptive music engine for video games called “barelyMusician”.(Rumsey, 2015)

3.5.1 barelyMusician

What barelyMusician does, is composing music by precomposed loops which are segmented in a vertical order using branching and layering. This still requires the hands of a human composer, but not as much as with conventional means. Only a fraction of the usually 60-90 min of music is needed that the system works. The idea is that the composers only handle out tools and the software does the rest. Sounds simple and convenient but still isn’t. Those few bars of music have to be composed in a way that no matter how the software mixes them together, they should still sound acceptable which decreases the verity of the music being created. It’s because of that, that up to now no industrial standard of a generative music system exists, because experts as well as game companies still require quality that such a full autonomic composing system cannot deliver yet. According to Rumsey, the music that is being generated within barelyMusician is structured in three sub groups or as he writes, levels. The macro, meso and micro level. (Rumsey, 2015)

> “The macro level corresponds to the musical form or structure, broken up into a number of sections. At the meso level is the harmonic progression of a sequence of bars in a section, while at the micro level is the note sequence for each bar. The mode generator defines a musical scale for the rest of the system, which can be used for filling scores with appropriate notes.” (Rumsey, 2015)

But how does such a system know in which emotional state the gamer is currently in and which database does it use?

The most obvious way is of course the gamers surrounding, meaning the map, levels or stages. Using light, size and color parameters generative music systems learn, generate and transform soundtracks with a sequencer from a database of precomposed loops by a sound designer. It is then when the system itself should decide if a cave (as spoken of before in the thesis) needs a exciting, scary, relaxed or happy sound. Generative music systems then save presets of music bars according to different emotional states using a Markov chain (Rumsey, 2015) (= "a random process in which the probability distribution of the next step in a sequence only depends only on its current state." (Rumsey, 2015)).

> "Essentially the system learns the musical structure of seed pieces and creates a probability vector for each of the pitch and rhythm values in the seed material. It can then create new material based on similar probabilities, such as the likelihood that notes of a certain duration and pitch will follow each other." (Rumsey, 2015)

Simply put. The system knows from specific parameters in which emotional state the gamer is probably in and generates music from the already learned and saved emotional state preset databank according to the users feelings to again, enhance the aspect of immersion. But it is not only the surrounding parameters that are used for creating those presets. With modern technologies and object oriented programming environments such as PD (Pure Data) or Max-MSP it is possible, to determine the emotional state of a player with his or her face expressions and eye movements, analyzed via the webcam of the computer. Also in combination with smart watches, heart beat rates could be analyzed to tell if the gamer is nervous,

uncomfortable or totally relaxed. The only problem remaining is that even with such amount of data, algorithms of those programs are still not planned through smart enough, that the performance of the system wouldn't suffer from it and thus causing lags and other bugs (=errors). (Rumsey, 2009, 2010, 2015; Schmidt, 2003)

4 Conclusion

And the circle closes itself. From a few beeping electronical signals, to pages of orchestral music written by great composers and back again to programs that write binary music autonomously.

Leaving the question. Will there be any video game music composers left?

Of course there will be. Just not in the way we used to know. The work environment for video game sound designers, as it is going hand in hand with the IT industry and actually depending on it, is changing and evolving likewise. With new technologies come new opportunities as well as new challenges and obstacles. In the future video game composers and sound designers will probably have to distance themselves more and more from the common linear way of composing music, rather than learn how to program it for special occasions.

Even if generative music systems are going to be programmed in a way that they can nearly compete with human composers, they most likely won't excel them. When it comes to a non-physical artform that only exists for a few moments, like music is, people are tending to choose reality over virtuality. Because no matter how high the bitrates and dynamic ranges are going to be, they will never cause the same emotional impact as reality does. And no matter how well structured algorithms of generative music systems, or in other words, artificial composers are going to be, they always will be synthetic.

That being said, I attempt to think that there is going to be a point where musicians just have to accept the fact that they did not become musicians in order to program

music, but to write it. Up to this day Koji Kondo is writing his music in a linear, a bit of loop based, old fashioned way with self-built instruments or plastic ones from the toy store and is outsourcing the programming part to more competent people in that aspect. That's what's probably going to happen to the industry. Outsourcing and splitting up those work areas in order to get the results the audience desires and deserves.

References

Collins, K., & Collins, C. R. C. in I. A. K. (2008). *Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design*. MIT Press.

Cunningham, S., Grout, V., & Hebblewhite, R. (2006). *Computer Game Audio: The Unappreciated Scholar of the Half-Life Generation*.

Eladhari, M., Nieuwdorp, R., & Fridenfalk, M. (2006). The Soundtrack of Your Mind: Mind Music - Adaptive Audio for Game Characters. In *Proceedings of the 2006 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*. New York, NY, USA: ACM. https://doi.org/10.1145/1178823.1178887

Hsu, H. (2015). How Video Games Changed Popular Music | The New Yorker. Retrieved August 19, 2018, from https://www.newyorker.com/culture/cultural-comment/how-video-games-changed-popular-music

IASIG. (2009). IASIG - Home. Retrieved August 26, 2018, from http://www.iasig.org/wg/ixwg/

Jennett, C., Cox, A. L., Cairns, P., Dhoparee, S., Epps, A., Tijs, T., & Walton, A. (2008). Measuring and defining the experience of immersion in games. *International Journal of Human-Computer Studies*, *66*(9), 641–661.

McDonald, G. (2005). A History of Video Game Music - GameSpot. Retrieved August 19, 2018, from https://www.gamespot.com/articles/a-history-of-video-game-music/1100-6092391/

National Public Radio. (2008). The Evolution of Video Game Music. Retrieved August 19, 2018, from https://www.npr.org/templates/story/story.php?storyId=89565567

Purves, R. (2013). A Video Game Odyssey - a brief history of video game music. Retrieved August 19, 2018, from https://www.mfiles.co.uk/video-game-music-history.htm

Rumsey, F. (2009). Sound and Music for Games: Prokofiev to Pac-Man to Guitar Hero. *J. Audio Eng. Soc.*, *57*(6), 5.

Rumsey, F. (2010). The Art and Business of Game Audio. *J. Audio Eng. Soc.*, *58*(4), 4.

Rumsey, F. (2015). Generative music, emotions, and realism. *J. Audio Eng. Soc.*, *63*(4), 5.

Schartmann, A. (2015). *Koji Kondo's Super Mario Bros. Soundtrack (33 1/3)*. Bloomsbury.

Schmidt, B. (2003). Interactive Mixing of Game Audio. *NEW YORK*, 6.

The 405. (2014). The evolution of music in video games. Retrieved August 19, 2018, from https://www.thefourohfive.com/music/article/the-evolution-of-music-in-video-games-139

Vorderer, P., & Bryant, J. (Eds.). (2006). *Playing Video Games: Motives, Responses, and Consequences* (1 edition). Mahwah, N.J: Routledge.

Wood, R. T. A., Griffiths, M. D., & Parke, A. (2007). Experiences of time loss among videogame players: an empirical study. *Cyberpsychology & Behavior: The Impact of the Internet, Multimedia and Virtual Reality on Behavior and Society*, *10*(1), 38–44. https://doi.org/10.1089/cpb.2006.9994

Xiaoqing Fu, J. Z. (2015). The Influence of Background Music of Video Games on Immersion. *Journal of Psychology & Psychotherapy*, *05*(04). https://doi.org/10.4172/2161-0487.1000191

YOUNG, D. M. (2013). THE FUTURE OF ADAPTIVE GAME MUSIC: THE CONTINUING EVOLUTION OF DYNAMIC MUSIC SYSTEMS IN VIDEO GAMES, 4.