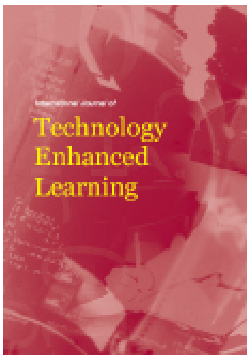


**International Journal of Technology Enhanced Learning**

ISSN online: 1753-5263 - ISSN print: 1753-5255
https://www.inderscience.com/ijtel

# The use of generative music in nonlinear music composition for video games

Alexi Harkiolakis

**DOI:** 10.1504/IJTEL.2023.10055011

**Article History:**

| | |
|---|---|
| Received: | 24 January 2023 |
| Last revised: | 26 January 2023 |
| Accepted: | 30 January 2023 |
| Published online: | 12 December 2023 |

# The use of generative music in nonlinear music composition for video games

Alexi Harkiolakis

King's College London,
Strand, London,
WC2R 2LS, UK
Email: alexi.harkiolakis@kcl.ac.uk

**Abstract:** The future of video game music will call for more nonlinear musical compositions as interactive audio technology and algorithms advance. There is a lack of literature exploring in-depth the potential of generative music in a commercial product. The purpose of this archival case study is to explore how the inclusion of generative music systems enhances nonlinear music composition in video games. This archival case study consists of a thematic analysis of Jolly and McLeran's (2008) presentation at the *Game Developer's Conference* on applying the innovative areas of composing generative music for the video game *Spore*. The findings of this study indicate that wider adoption of generative methods in video game music provides many benefits but poses new challenges for game developers. Despite this, the case of *Spore* shows the feasibility of integrating a generative music system into a video game, even if it is restricted to specific areas of gameplay.

**Keywords:** generative music; nonlinear music; music composition; video game music; video games; archival case study; audio technology; Spore; interactive music; algorithms.

**Reference** to this paper should be made as follows: Harkiolakis, A. (2024) 'The use of generative music in nonlinear music composition for video games', *Int. J. Technology Enhanced Learning*, Vol. 16, No. 1, pp.16–26.

**Biographical notes:** Alexi Harkiolakis is an Academic Researcher and Freelance Composer specialising in music for film, television, videogames and concert. He holds a Master's of Music in Composition from King's College London and is a Doctoral candidate in the Music PhD programme at Royal Holloway, University of London. He is a member of The Society of Composers and Lyricists, the Incorporated Society of Musicians, the Game Audio Network Guild, Business Fights Poverty and WhyHunger. His research interests include screen music, generative music, interactive music, and microtonality.

## 1 Introduction

An interactive medium like video games can challenge the traditional notions of what constitutes a music composition. Throughout the history of western music, music compositions have been generally thought of as linear, with a clear beginning, progression, and end, regardless of the performer's interpretation (Burkholder et al., 2014). In time, such notions of linearity began to be challenged, leading the field into the new realm of real-time evolution in music composition. Procedural music, whose features evolve in real-time, has already been explored in contemporary classical and academic settings (Eigenfeldt and Pasquier, 2009; Kaliakatsos-Papakostas et al., 2016; Lopez-Rincon et al., 2018; Rowe, 1999; Scirea, 2017; Wallis et al., 2011; Wooller and Brown, 2005). The idea that a musical composition can sound different on every playthrough is nothing new, and one of the earliest examples is Mozart's *Musikalisches Würfelspiel* (musical dice game), in which a roll of the dice determined the order of musical sections (Plut and Pasquier, 2020).

In 2021, roughly a year before the writing of this essay, the video game industry was valued at 173.70 billion USD and is expected to rise in the coming years (Mordor Intelligence, 2020). The future direction of video game music has led to some fascinating methods for constructing nonlinear musical compositions for this peculiar setting and continues to be fine-tuned due to advances in interactive audio technology. Of interest in this discussion is the development of generative music, which can achieve a greater nonlinearity than traditional approaches to video game music. While many have already written about the various aspects of generative music and demonstrated them with original generative systems, there is a lack of literature exploring in-depth the potential of generative music in a commercial product. There is also the aspect of the granularity of a generative system and how to approach an effective system with greater granularity. This concern has also not been explored in great depth in the literature regarding commercial products rather than original experimental systems created by the researcher(s). To address this issue, the purpose of this archival case study is to explore how the inclusion of generative music systems enhances nonlinear music composition in video games.

This case study paper will first present a literature review of traditional nonlinear compositional practice and the current scope of generative music in the video game industry. An archival case study will consist of a thematic analysis of Jolly and McLeran's (2008) presentation at the *Game Developer's Conference,* applying the innovative areas of composing generative music for the video game *Spore*. This study is significant as it contributes unique knowledge to the music composition and interactive audio technology literature through this archival case study results. The relevant research question aligning with the purpose of the study and the archival case study design is as follows:

- What impact will the wider adoption of generative music systems have on the future of nonlinear music composition in video games?

This archival case study is grounded in the concept of interactive audio technology as the new determining factor for future developments in music composition practice (Sweet, 2015).

## 2 Literature review

### *2.1 A brief introduction to the concept and history of video game music*

One crucial aspect of understanding video game music development is the technological limitations of the various mediums used to play video games. In the beginning, the availability of various musical parameters, such as timbre and number of voices, was determined by the sound circuits of the custom-built arcade machine. This technology's sounds were severely limited and restricted to simple waveforms. The applications of this technology ranged from the simple bleeping sounds of *Pong* in 1972 to the featuring of monophonic melodies in *Boot Hill* in 1977 for three basic game states (Sweet, 2015).

Indeed, the easiest technique for adaptive music was immediately changing a piece of music, played on an endless loop, to another after a switch of the game state, like starting a level or beating the game (Plut and Pasquier, 2020). The arcade game *Space Invaders* is of great interest because it features one of the earliest examples of music that can change in real-time. The music itself was highly primitive, featuring only a descending four-note pattern, but gradually increased in tempo as the enemies got closer to the player (Sweet, 2015). *Space Invaders* would mark the first proper step towards nonlinear music in games.

The first step to opening the possibilities of adaptive music was the invention of the programmable sound generator (PSG) in the early 1980s. PSGs function similar to conventional synthesisers in that, apart from basic waveforms, the composer also has access to ADSR (attack, delay, sustain and release) parameters that could change the volume-based characteristics of the sound. One of the earliest games utilising this technology, *Frogger* (1981), featured switching between different music tracks after completing a level (Sweet, 2015).

As the decade progressed, the incorporation of FM sound chips in arcade machines, *Marble Madness* (1984) being one of the first to utilise it, facilitated the inclusion of eight simultaneous voices leading to greater musical complexity. The ability to compose music with greater sonic detail was helped significantly by the more extensive storage offered by adopting the CD-ROM in the 1990s and DVD-ROM in the 2000s. Also, the development of CPUs in consoles and personal computers, and computer software meant greater possibilities for adaptive systems (Sweet, 2015).

The technology crucial to understanding present-day video game scoring is known as audio middleware (Phillips, 2014). Audio middleware refers to any software that can build adaptive music systems and integrate them into the game engine. These programs came about as programmers began to build reusable libraries of code over time in order to streamline the game development process. While there are several different audio middleware programs available at the time of the writing of this paper, two of the most popular programs used in game development include Firelight's *FMOD* and Audiokinetic's *Wwise* (Sweet, 2015). Using these programs, composers can program their music without knowing how to code, loop, transition, and change in real-time and export these functions for various platforms with varying forms of optimisation (Audiokinetic Inc., n.d.; Firelight Technologies Inc., n.d.).

With this technology comes the four basic techniques of present-day video game scoring: transitioning via bridges, stingers, vertical remixing, and horizontal resequencing (Sweet, 2015). Rather than switching directly from one track to the other or simply crossfading between them, the music can utilise a short, musically sensible bridge to crossfade smoothly between two contrasting tracks. Transition points between tracks can also be masked using a stinger, a quick, prominent sound that plays during the crossfade. Nevertheless, if the player were to change game states quickly and in the same area, it would be more sensible to have the same track but to re-orchestrate or 'remix' the track to reflect these changes. The fourth technique, horizontal resequencing, breaks up a track into segments of varying size, allowing the music to switch to another track without interrupting the music in the middle of a phrase (Phillips, 2014; Sweet, 2015). While these techniques alone can offer many possibilities for creating a seamless flow of music throughout the game, is it possible to take the adaptability factor a step further for greater variability and immersion?

### *2.2 The scope of generative music in the video game industry*

The next significant development in video game music is generative, or procedural, music, a catch-all phrase referring to music generated in real-time with varying degrees of granularity. Various musical properties can be defined and randomised with added constraints to ensure stylistic consistency in a generative system. Such properties can be horizontal, ranging from melodies, phrases, motifs, individual notes, time signatures, and subdivisions of the beat, or vertical, including harmony and instrumental timbre and arrangement. Other properties that can also be automated include dynamics, tempo, ADSR characteristics, effects like reverb and chorus, and, in the case of synthesised sound, filter (timbre) parameters. The musical material used by the system can take the form of synthesised sound or sampled audio.

Algorithms that can be used to create generative systems include rule-based algorithms, stochastic algorithms, which balance randomness with higher probability outcomes, genetic algorithms, which determine the fitness of each solution during the process; and artificial neural networks, which mimic the human brain in that it relies on making connections based on user feedback. The composer or audio programmer can input how the system uses the material or learn from the system itself by analysing pre-existing music (Farnell, 2007; Plut and Pasquier, 2020). While middleware can be used to achieve some of these goals (*Wwise*, for example, can randomise musical phrases with variable probability for each phrase), other software like *PureData*, used and modified for the generative portions of *Spore*, can be used for more complex solutions (Plut and Pasquier, 2020; Sweet, 2015).

One of the earliest examples of a game that uses generative music is *Ballblazer* (1984), which randomly generates melodic improvisations using pre-defined scale material, and various weighted choices to give the melody structure over a repeated four-bar rhythm section track (Collins, 2009; Plut and Pasquier, 2020). Another early example is *Otocky* (1987), which generates pitches when the player shoots their weapon and is programmed to fit the changes in the main music track (Plut and Pasquier, 2020). A recent game that uses some form of randomised using musical phrases or clips as the smallest unit includes *Anarchy Online* (2001), *Diner Dash* (2004), *No Man's Sky* (2016), and *DOOM* (2016), and games like *Dark Void* (2010) and the *Halo* series use real-time re-arrangement of instruments over time (Plut and Pasquier, 2020). While adopting these techniques in the above products is a good step for exploring adoption generative systems, this discussion explores the potential of an effective system with greater granularity than the ones above, and that is where *Spore* comes in.

## 3 Methodology

This study applies an archival case study design (Yin, 2017) to explore current developments in interactive audio technology and the future practice of nonlinear music composition. Using a qualitative approach for an exploratory study allows the researcher to gather much richer, in-depth data than quantitative approaches using open-ended surveys that only provide limited statistical findings (Halkias et al., 2022). Case study design generally allows the researcher to be more flexible when examining contemporary issues, given that, in this case, the researcher cannot control behavioural events (Yin, 2017). Single-case research designs follow an inductive approach where researchers identify patterns in the qualitative data and categorise data into themes using a descriptive approach (Saldana, 2016; Yin, 2017). The archival data collected underwent thematic analysis to address the study's purpose and presented data to answer the study's research questions (Yin, 2017).

## 4 The case study: procedural music in SPORE

### *4.1 Background*

The case study will use a generative music system in *Spore*, a simulation strategy game that allows players to create and develop a civilisation of customisable creatures (SPORE™, 2008). The source material used is a transcribed audio recording of 'Procedural Music in SPORE,' presented by audio director Kent Folly and audio programmer Aaron McLeran at the 2008 Game Developer's Conference in San Francisco, CA. Rather than follow the presentation linearly, this case study will switch between Jolly and McLeran's segments so that the subjects remain appropriately stratified. To start, later in the presentation, McLeran explained the reasoning behind the decision to use generative music in *Spore*:

> Why use procedural music? And our goal I think for both of us, Kent and I, was to create music that all ultimately supports the game itself. The objective of the game in Spore is clearly creativity, so we wanted to make music that sort of gets you in the mood to be creative. And so it doesn't pumble you over the head with crazy beats or complex music or something, but just something that sort of supports the idea in each of the given editors what kind of mode and what type of editor that they are… (Jolly and McLeran, 2008)

Furthermore, McLeran described the basic principles the team laid out for approaching this process:

> So there's just four things that sort of a rule of thumb that we use when we're writing these things. So make the music so it's not distracting, make sure that when they're playing they're not, in a bad way they're not like, what was that sound? That was weird. Two, doesn't repeat…You want to have these people spend as long as possible in these editors when they're playing it and if they're having to listen to the same cheesy melody over and over then it'll get annoying. The other thing too is we want to make it playful. And then the last thing is that we want to make sure that the music responds to the user, but in a way that makes sense musically. (Jolly and McLeran, 2008)

According to Jolly, various places in the game use procedural music, notably in the different editors and the city music planner segment that features user-generated music. While CPU load is a notable technical hurdle in implementing generative systems, Jolly explained that,

> We pretty much worked around that because a lot of the music that we wanted, the place where we were placing the heaviest procedural music was in the editors, and there was no game simulation happening there so it was actually a

> good opportunity for us to get more bandwidth than you would usually get in games. (Jolly and McLeran, 2008)

Also, as McLeran noted in the discussion, the non-generative, or precomposed, music was used as an aesthetic blueprint for the generative segments.

## *4.2 Thematic analysis*

### *4.2.1 Use of interactive audio technology beyond middleware*

In *Spore*, the generative system was programmed with a modified version of PureData called 'EAPD' (EA stands for Electronic Arts, the game's publisher), a programming language that has allowed for great flexibility in implementing their ideas for the game. Towards the beginning, Jolly demonstrates the various programming objects used in the system, including the play object, which can play an individual sound or a group of sounds, a bang object, which represents an event, and the metro object, which acts as a metronome and can "output events at some rate that you specify" (Jolly and McLeran, 2008). The core element of this system is the use of random number generators. A number, or 'seed', representing a pattern of MIDI notes, is entered into the generator and plays the MIDI notes in random order on each repeat. By adding a new seed or 're-seed,' these generators transition to a different set of notes that will also play in random order. Jolly first explains how these generators work with musical material:

I will type in a seed number that I want to give to the random object. And then I will type in another object called select, which says that every time you see some particular number do one of these bangs here, do something. So what we're going to do is we're going to re-seed this random number generator every eight count. And all of a sudden you have a repeatable drum pattern, which is pretty amazing because nobody wrote this, it just came out. If I put in a different random seed you'll get a different random pattern, but sounds sort of intentional… But certainly it didn't deal with a lot of issues in procedural music, mainly one being pitch. So our audio engineer, Justin Graham, built us a very simple sampler playback. You can give it a MIDI note, velocity, this, and we can make sampler instruments that have key mappings and velocity ranges, that type of thing... So we have that, and then here's another example, the exact same thing as the drums but now we're giving it MIDI pitch (Jolly and McLeran, 2008).

Jolly then demonstrated this setup in the creature editor:

> So there's several things that it does here, there's several things that we're responding to. One of them is with the music there's this big patch running sort of in the background, and it's listening to all kinds of different events from the game. So if I pick on a different palette of creature parts it'll actually switch instruments, it'll change seeds and do different things sort of quasi randomly. And if you add a part it'll play a little melody here as you're building your creature. (Jolly and McLeran, 2008)

Another technique they used in this setup, which was encouraged by Eno, was the use of DSP effects. Specifically, Jolly mentioned the use of modulation in the creature editor:

> So one of the things that happens is that as you switch pallets it'll actually run and will ring modulator across the drum beats as you're doing it. And as you're hearing earlier, you'll hear this sort of cool sound over these things, that's something that was really surprising that we could just, automatically we had

> all this music and then latter on just apply this effect to it. (Jolly and McLeran, 2008)

Within this system, they were also able to apply the previously mentioned technique of reducing the orchestration over time:

> One is that one of the big focuses on this, one of the big design goals here was that we wanted to make music that people could listen to for a long time, because we expected that they spend a lot of time in here making creatures. And so one of the things that they do, or that the music does is that over time the density of the rhythmic loops will actually go down. So if you've been in the game playing 15 minutes it'll be playing say 25% of the time, as opposed to when you started maybe the beats were playing 80% of the time. (Jolly and McLeran, 2008)

Although we have just scratched the surface of *Spore*'s generative music system, we can now delve deeper into how the audio team used this new, versatile system to create varied yet convincing musical results.

### *4.2.2 Creating a good aesthetic with synthesised and sampled sound*

This game, which is relatively old by now, was chosen for the case study because of the greater granularity the team achieved compared to the more 'block-like' approach with games like *DOOM*. Of course, this necessitated using sampled sounds via MIDI and synthesised sound, which was one of the causes of the initial hesitation by others in the team due to aesthetic concerns, another challenge that generative music faces.

As Jolly described it: "there was this sort of a payoff for having note-based music because it's very responsive and you have a lot of flexibility, but it also oftentimes just doesn't sound like good production in my opinion" (Jolly and McLeran, 2008). Jolly later explained that this hesitation was eased with the collaboration with composer Brian Eno and the fact that EAPD was already working with the game.

To create a good musical aesthetic using this approach, they decided that the orchestration would focus on timbres that sound the best when sampled, hence the frequent usage of percussion. As an observation, it also makes sense to use synthesised sound in a generative context because such timbres already sound artificial and are thus immune to the aesthetic issues of sampled real instruments. McLeran mentions this aspect of the music in the Q&A:

> The thing that we found though is that the most successful instruments that we used a lot were the plunky ones, because they were the most responsive. But we do use longer instruments for more the ambient stuff, but the things that you hear the most, the ones that respond quickly, and that sound the best as MIDI instruments… no saxophones were used, no trumpets. So we went to stuff that sounded naturally good in the MIDI context, which is sort of percussive, plunky instruments. (Jolly and McLeran, 2008)

### *4.2.3 Balancing randomness with order*

As we have mentioned in the literature review, one of the challenges of creating effective generative music is to ensure that excessive randomisation does not disrupt the stylistic continuity of the music. In the first theme, we have seen how seeds and the restriction of musical material to specific scales were used to control the amount of randomisation and produce musically satisfying material. Jolly explains how they stratified specific musical material to reflect different kinds of creatures in the creature editor:

> One of the strategies that you could do when you're playing the creature game is you can choose whether you're going to be in sort of an attack creature or a social creature, or some mixture of those two elements. And so if you decide let's say that you wanted to be a very attack-based creature, then we wanted the music to sound maybe a little bit darker if you're generally meaner. So as you go on here, let's see if I can get some good attack-only parts on here, buzz saws maybe. And now you'll hear that the music kind of goes into more minor mode, it sounds a little darker. So it starts out as sort of a neutral sort of minor pentatonic, and if you go all social you get major and if you go all dark you get darker. (Jolly and McLeran, 2008)

McLeran describes a similar scenario when demonstrating the UFO editor, which has more futuristic-sounding music than the simpler, primitive style of the creature editor. McLeran also explains the additional musical restraints they programmed in order to give the music more structure and sound more tonal:

> So basically you have to create a pulse so that's what this counter does, pulse maker. Then it comes through here, counts one to eight over and over and over. So that's your eighth notes or whatever. Then I use the selection object, pick out the first and the fifth beat out of eight. So that's going to be one, two, three, four, one, two, three, four kind of a thing. And if it picks the one and the five it hits this, which is a gate, it sends this message to here which opens up this gate so that messages over here get sent out here. So these are your notes that you want to emphasize, your tonic, and your dominance, and your thirds, so basically the cords. (Jolly and McLeran, 2008)

McLeran then explains how he programmed the seeds to change every bar in order to create phrase structure, as well as how they used the influential eighteenth-century counterpoint treatise *Gradus ad Parnassum* by Johann Joseph Fux to restrict the polyphonic aspects of the music for a musically satisfying sound:

> And so I opened up Johann Fux's Gradus ad Parnassum, which he wrote in the 1600s and was studied by Bach himself, outlines basically on like page eight or something perfect pseudo code to implement your own counterpoint generator…basically what you're doing here is it's a bunch of stuff to avoid parallel fifths and octaves, that's basically all you're trying to do…And the nice thing about counterpoint is that it was pre-chord theory, and it turns out chord theory was derived from analyzing counterpoint. So you actually have packaged in a chord generator, so this is actually generating chords and I'm not even thinking about chords, which is kind of cool… (Jolly and McLeran, 2008)

McLeran continues by describing the process of creating this kind of music as 'composing with probabilities'; this is how they managed to achieve some form of musical order without sacrificing the variability. As McLeran puts it:

> It's not that high tech. I think a lot of composers here probably are like oh yeah, that doesn't seem too bad, hopefully. And it's basically using all this, there's nothing fancy, no AI, no computer crazy, it's not like anybody's going to be out of a job as a composer. It's just a different way to compose, you're just using music theory in probabilistic ways, that's it. (Jolly and McLeran, 2008)

## 5 Implications and recommendations for professional practice

Video game music functions similar to its film and television counterparts to support the dramatic arc and emotions of the fictional characters of the plot, provide a sense of time and place and reinforce the aesthetic style of the work. While film and television have an unchanging duration and progression of events, video games can be played for hours on end, and even the progression of events can be altered depending on the player's choices (Davis, 2010; Sweet, 2015). Because video games cannot function without the consumer's active participation, rather than being a 'passive' participant in linear media, the issue of 'player immersion' arises in the discussion of effective game design (Sweet, 2015; Zhang and Fu, 2015). Moreover, even though the successful implementation of a highly granular generative system in *Spore* did not cause an immediate 360 turn in the industry, it does demonstrate the bright future generative systems have in a medium in which audio teams have strived for greater immersion.

It is also important to note that generative music is not the only technology that has the potential to develop the nonlinearity of video game music further. One new area of interactive technology that has yet to be adopted by the industry is player biofeedback for a more personalised audio experience. Various researchers, including Dekker and Champion (2007), Kuikkaniemi et al. (2010), Garner and Grimshaw (2013) and Houzangbe et al. (2018), have explored the use of various devices to measure heart rate and, in some instances, electrodermal activity in order to gauge the player's psychological state and to program the readings to trigger changes in the game audio.

Although these studies focused more on sound effects, most commonly a regular heartbeat, Dekker and Champion (2007) also programmed the background music in their modified game to increase or decrease volume as the player's readings changed. These studies have had mixed results, but all point to the potential of adopting such technologies. Thus, while more work needs to be done to assess the commercial viability of biofeedback sensors, it is clear that this is potentially a new pathway for game developers to expand on the nonlinearity of video game music.

The previously mentioned growth and importance of the video game industry will make it an attractive choice for aspiring composers, which means they will be required to adapt to the new technology that the industry embraces. This fact is crucial because many of the experiments in generative music, based on the literature, have been restricted to purely academic settings. Thus, because more composers will be learning these new technologies, generative music systems will likely become increasingly prominent in music composition practice outside the video game industry.

## 6 Conclusions

Wider adoption of generative methods in video game music provides many benefits but poses new challenges for game developers (Farnell, 2007). The most obvious benefit is increased musical variety, which is especially important for games of greater length. Repetition has long been an issue in video game music as it can end up annoying the player and reduce immersion, which is why some games have opted to fade out the music after a certain amount of repetition (Collins, 2009).

One of the issues holding generative music back has to do with the musical aesthetics compared to recorded linear music, which has to do with the former's frequent use of sampled sounds for greater granularity (Farnell, 2007; Plut and Pasquier, 2020). There is also the issue of generative music being more CPU intensive and the possibility of hiring more audio programmers who are more costly than composers (Farnell, 2007; Plans and Morelli, 2012). Despite this, the case of *Spore* shows the clear benefits and feasibility of integrating a generative music system into a video game, even if it is restricted to specific areas of gameplay. The future viability of generative music will probably depend on further developments in processing power. However, a highly granular system like this one made its way into a commercial product reinforcing the potential for generative music for greater adoption in future projects.

## References

Audiokinetic Inc. (n.d.) *Wwise Certifications* [online] https://www.audiokinetic.com/education/learn-wwise/ (accessed 7 April 2022).

Burkholder, J.P., Grout, D.J. and Palisca, C.V. (2014) *A History of Western Music*, 9th International Student ed., WW Norton & Company.

Collins, K. (2009) 'An introduction to procedural music in video games', *Contemporary Music Review*, Vol. 28, No. 1, pp.5–15, https://doi.org/10.1080/07494460802663983.

Davis, R. (2010) *Complete Guide to Film Scoring: The Art and Business of Writing Music for Movies and TV*, Hal Leonard Corporation, Milwaukee, WI.

Dekker, A. and Champion, E. (2007) 'Please biofeed the zombies: enhancing the gameplay and display of a horror game using biofeedback', *DiGRA'07-Proceedings of the 2007 DiGRA international conference: Situated play*, pp.550–558, Digital Games Research Association (DiGRA), https://doi.org/10.25917/5d1443e8af4a0.

Eigenfeldt, A. and Pasquier, P. (2009) 'A real-time generative music system using autonomous melody, harmony, and rhythm agents', *12th International Conference on Generative Arts (GA2009)*, December, Milan, Italy [online] https://generativeart.com/on/cic/GA2009Papers/p7.pdf.

Farnell, A. (2007) 'An introduction to procedural audio and its application in computer games', *Audio Mostly Conference Proceedings 2007*, 23 September, Association for Computing Machinery (ACM) , pp.1–31 [online] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.531.2707&rep=rep1&type=pdf.

Firelight Technologies Inc. (n.d.) *FMOD Studio Concepts* [online] https://fmod.com/resources/documentation-studio?version=2.02&page=fmod-studio-concepts.html (accessed 7 April 2022).

Garner, T.A. and Grimshaw, M. (2013) 'Psychophysiological assessment of fear experience in response to sound during computer video gameplay', *IADIS International Conference Interfaces and Human Computer Interaction 2013*, July, International Association for Development, IADIS, pp.45–53 [online] http://www.iadisportal.org/digital-library/a-psychophysiological-assessment-of-fear-experience-in-response-to-sound-during-computer-video-gameplay.

Halkias, D., Neubert, M., Thurman, P.W. and Harkiolakis, N. (2022) *The Multiple Case Study Design: Methodology and Application for Management Education*, Routledge, Abingdon-on-Thames.

Houzangbe, S., Christmann, O., Gorisse, G. and Richir, S. (2018) 'Fear as a biofeedback game mechanic in virtual reality: effects on engagement and perceived usability', *Proceedings of the 13th International Conference on the Foundations of Digital Games*, Association for Computing Machinery (ACM), August, pp.1–6, https://doi.org/10.1145/3235765.3235787.

Jolly, K. and McLeran, A. (2008) 'Procedural music in Spore', *Game Developer's Conference*, San Francisco, CA, USA.

Kaliakatsos-Papakostas, M.A., Floros, A. and Vrahatis, M.N. (2016) 'Interactive music composition driven by feature evolution', Vol. 5, No. 1, Article 826, SpringerPlus, https://doi.org/10.1186/s40064-016-2398-8 (accessed 3 March 2022).

Kuikkaniemi, K., Laitinen, T., Turpeinen, M., Saari, T., Kosunen, I. and Ravaja, N. (2010) 'The influence of implicit and explicit biofeedback in first-person shooter games', *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems 2010*, April, Association for Computing Machinery (ACM), pp.859–868, https://doi.org/10.1145/1753326.1753453.

Lopez-Rincon, O., Starostenko, O. and Ayala-San Martín, G. (2018) 'Algoritmic music composition based on artificial intelligence: a survey', *2018 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, February, Institute of Electrical and Electronics Engineers (IEEE), pp.187–193, https://doi.org/10.1109/CONIELECOMP.2018.8327197.

Mordor Intelligence (2020) *Gaming Market – Growth, Trends, COVID-19 Impact, and Forecasts (2021–2026)* [online] https://www.mordorintelligence.com/industry-reports/global-gaming-market (accessed 3 March 2022).

Phillips, W. (2014) *A Composer's Guide to Game Music*, MIT Press, Cambridge, MA.

Plans, D. and Morelli, D. (2012) 'Experience-driven procedural music generation for games', *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 4, No. 3, pp.192–198, https://doi.org/10.1109/TCIAIG.2012.2212899.

Plut, C. and Pasquier, P. (2020) 'Generative music in video games: state of the art, challenges, and prospects', *Entertainment Computing*, Vol. 33, Article 100337, https://doi.org/10.1016/j.entcom.2019.100337.

Rowe, R. (1999) 'The aesthetics of interactive music systems', *Contemporary Music Review*, Vol. 18, No. 3, pp.83–87, https://doi.org/10.1080/07494469900640361.

Saldana, J. (2016) *The Coding Manual for Qualitative Researchers*, 3rd ed., Sage Publications, Thousand Oaks, CA.

Scirea, M. (2017) *Affective Music Generation and Its Effect on Player Experience*, Doctoral dissertation, Publication No. 136, IT University of Copenhagen [online] https://pure.itu.dk/portal/en/publications/affective-music-generation-and-its-effect-on-player-experience(f0386da8-7944-4afd-9493-2d252460a25a).html (accessed 15 February 2022).

*SPORE*™ (2008) Steam [online] https://store.steampowered.com/app/17390/SPORE/ (accessed 15 March 2022).

Sweet, M. (2015) *Writing Interactive Music for Video Games: A Composer's Guide*, Pearson Education, London.

Wallis, I., Ingalls, T., Campana, E. and Goodman, J. (2011) 'A rule-based generative music system controlled by desired valence and arousal', in Zanolla, S., Avanzini, F., Canazza, S. and de Götzen, A. (Eds.): *Proceedings of the 8th International Sound and Music Computing Conference (SMC)*, July, Padova University Press, pp.156–157 [online] http://smc.afim-asso.org/smc11/papers/smc2011_107.pdf (accessed 2 February 2022).

Wooller, R.W. and Brown, A.R. (2005) 'Investigating morphing algorithms for generative music', in Innocent, T. (Ed.): *Proceedings of Third Iteration: Third International Conference on Generative Systems in the Electronic Arts*, December, Queensland University of Technology (QUT) ePrints [online] https://eprints.qut.edu.au/24696/.

Yin, R. (2017) *Case Study Research: Design and Methods*, 6th ed., Sage Publications, Thousand Oaks, CA.

Zhang, J. and Fu, X. (2015) 'The influence of background music of video games on immersion', *Journal of Psychology & Psychotherapy*, Vol. 5, No. 4, Article 1000191, https://doi.org/10.4172/2161-0487.1000191.