ZANDER HULME
Queensland University of Technology
zanhulme@gmail.com

Killing-off the Crossfade

Achieving Seamless Transitions With Imbricate Audio

ABSTRACT

This paper examines common approaches to implementing adaptive musical scores in video games using pre-rendered music that make heavy use of the crossfade transition and are used in the production of both AAA and Indie titles (Phillips, 2014a; Sweet, 2014; Collins, 2008; Collins, Kapralos and Tessler, 2014). The aim of this paper is to critique shortcomings of the crossfade and to propose a way to address some of these shortcomings using existing technology. This is achieved through a new composing, recording, and implementation process, provisionally called "imbricate audio", the viability of which is tested through the creation of an original composition. Where crossfades create "conspicuous game soundtrack signifiers" (Houge, 2012 p. 1), imbricate audio aims to deliver modulations that sound close to those found in live performance or the linear scores of cinema, potentially increasing the ability of composers to immerse players in gameplay with adaptive music.

KEYWORDS: *Adaptive Audio, Composing, Game Music, Immersion, Recorded Music, Pre-rendered Audio, Music Systems*

INTRODUCTION

Like many of history's great inventions, adaptive audio was discovered by accident. When working on Taito's 1978 game *Space Invaders*, the game's designer, Tomohiro Nishikado, found a rather interesting bug: destroying on-screen "invaders" freed up processing power and caused the game to run faster (Paul, 2012, p. 71). According to Karen Collins (2005, p. 2), although the music in *Space Invaders* is only a simple four-note loop, this was the first instance of a game featuring continuous background music (previous titles containing either no music or only brief musical stingers heard at key moments). When Nishikado noticed the game increase in speed, he also heard the music increase in tempo. By intentionally exaggerating this effect, he was able to add tension as the game progressed (Paul, 2012, p. 71) – in fact, the music was responding emotionally to the gameplay.

Audio quality in game music has come a long way since the monophonic 8-bit synthesis chips used in *Space Invaders* cabinets. The technological evolution of arcades, consoles, and home computers has seen the introduction of sophisticated polyphonic synthesis, digital samplers, and fully recorded scores. Since the release of Sony's PlayStation 3 in 2006, home gaming consoles and PCs are capable of greater-than-CD-quality recorded music in 7.1 surround sound (Collins, 2008, p. 71). However, the ability to put a full orchestra into a game has come with a compromise in versatility: a simple musical function such as Nishikado's adaptive tempo increase in *Space Invaders* would be a challenge for a game with a fully-recorded score.

> Improvements in adaptive audio implementation such as imbricate audio may well have the potential to improve the gaming experience as a whole.

Cinema has long made use of changes in music to assist narrative, responding to every nuance of the on-screen action and story. According to Belinkie (1999, p. 1), now that games too can have cinematic-style scores––some of which are even written by renowned film composers (Copeland, 2012, p. 14). Players may in some cases expect the same fluid musical response as they would hear in a film. Liam Byrne explains: "We're used to constant soundtracks in [our] entertainment. The more exactly the video game soundtrack matches your experience, the more involving that experience is going to be" (cited in Belinkie, 1999, online). It is worth noting that the video game medium has its own precedents and narrative devices and that making games more film-like is not necessarily an improvement, but one cannot ignore that many of our expectations of scored music for visual media are derived from decades of cinema. However, a defining characteristic of game music is its nonlinearity, the "ability of the game's music to respond to things happening in the game [that] makes video game music unlike other genres of music" (Lerner, 2014, p. 1). The music matches action in films as both film and music are linear media, and as such the two can be meticulously interwoven. Without a linear form to which the music can be matched, a game score requires "complex relations of sounds and music that continue to respond to a player throughout a game" (Collins, 2008, p. 211). Game music that fails to meet these expectations of fluidity can adversely affect player immersion.

Immersion is a mental state, which Winifred Phillips describes as the "ultimate goal" of game development (2014a, p. 52). When experiencing immersion, players "forget that they are playing a game. For a short while, these gamers have surrendered themselves to the fictional world of the game developers, entering the flow state that allows them to relinquish awareness of themselves and suspend their disbelief in favor of the plausible truths that the game presents to them" (Phillips, 2014a, p. 54).

Laurie N. Taylor (2002) distinguishes immersion from the experience of being "engrossed in a video game just as a reader would become engrossed in a novel, or a viewer in a film," stating that video games can cause "intra-diegetic immersion, which allows the player to become deeply involved in the game as an experiential space" (p. 12). Research on this effect indicates that adaptive game music significantly increases a player's sense of immersion (Gasselseder, 2014, p. 3) and that a game's level of interactivity is related to the level of immersion that players can experience (Reiter, 2011, p. 161). The conclusion that one can draw from this research is that improvements in adaptive audio implementation may well have the potential to improve the gaming experience as a whole.

PROBLEM

The numerous inventive approaches to implementing adaptive game music fall largely into two camps: sequenced audio and pre-rendered audio. Sequenced audio includes the use of synthesizers, samplers, and sophisticated virtual instruments (which blend aspects of both). Sequenced music is produced in real-time during gameplay, and in the hands of a talented audio programmer can transform in almost any way desired. However, the characteristics of current tools constrain sequenced audio to "electronic-sounding" soundtracks. Highly developed modern virtual instruments may be capable of sounding sufficiently realistic to fool the listener into thinking that they are hearing a live recording (Tommasini & Siedlaczek, 2016), but such instruments have too high a memory and processing cost to run during gameplay, as current gaming consoles are already being pushed to their limits to deliver cutting-edge graphics (Sweet, 2014, p. 205).

The alternative, pre-rendered audio can consist of recorded musicians, sequenced music that has been "printed" into fixed audio files, or a mixture of the two. Adaptive implementation can be achieved in code, but various tools for implementing pre-rendered audio in games (known as middleware) have arisen to make the task of creating adaptive pre-rendered soundtracks much easier (Firelight Technologies, 2016). This technology allows for real-time mixing of parallel vertical stems, horizontal resequencing of independent musical sections, and many other complex interactions that are based largely on combining the former two techniques. While these processes are too complicated to cover in depth in this paper, for the interested reader, Winifred Phillips gives an excellent introduction to these concepts on her blog (2014b and 2014c).

One of the limiting factors that prevents pre-rendered video game music from achieving the seamless flow of film music is the use of fade transitions. When games have cinematic-sounding music, players expect game scores to behave like movie scores (Stevens & Raybould, 2014, p.149), but fade-ins, fade-outs, and crossfades are transitions that composer Ben Houge refers to as "conspicuous game soundtrack signifiers" (2012, p. 1), which rarely appear in film. A notable exception is the modular score, which opts for independent sections of music, the end of each section fitting neatly into the start of every other section. This has the benefit of seamless changes without using fades, though the system has to wait for the current section to end before a change can be made, which can result in short periods of inappropriate music (Stevens & Raybould, 2014 p. 150). A good example of an expertly crafted modular score is the soundtrack to *Monkey Island 2: LeChuck's Revenge* (LucasArts, 1991), which made use of the iMUSE system (Collins, 2008, p. 56). Watching a video capture of the game, one can see that the Guybrush (the player character) traverses different areas of the town of Woodtick, a distinct musical theme can be heard in each area (as discussed in Silk, 2010).

When the iMUSE system is triggered to change themes, by Guybrush crossing into a new area, a brief musical transition is selected depending on the playback position in the score. While these transitions create excellent seamless segues between the various themes of Woodtick, they can also take time to respond while iMUSE waits for an appropriate point in the music to transition from, then the whole transition must be played before the new theme can begin (Collins, 2008, p. 53). In the case of the video capture above, the longest response time was just over 7 seconds, counting from when Guybrush leaves the bar just after 1:40 in the video. 7 seconds may not be a significant amount of time in point-and-click adventure games like the *Monkey Island* series, but in faster-moving genres, it can feel like a long period for inappropriate music to be playing.

The Woodtick scene relies on sequenced virtual instruments to grant the moments of transition a seamless fluidity. The remastered *Monkey Island 2: Special Edition* (LucasArts, 2010) features a fully pre-rendered score with many recorded instruments, and unlike the original (which featured no crossfades) the fading used to facilitate these same transitions in the *Special Edition* can be clearly heard––though some pre-rendered modular scores can use clever edits to avoid using crossfades altogether.

The biggest issue with using fades in pre-rendered scores is the way in which the listener anticipates certain familiar instruments (especially acoustic instruments) to sound. Specifically, fades can critically change the start, end, or reverb tails of notes––something that would be odd and jarring to hear at a live performance. Reverb tails can be thought of as the expected natural ringing-out of an instrument. They are the result of a combination of an instrument's decay sound and the reverb of the space (or artificial reverb effect) in which the instrument is played. Human ears are "remarkably sensitive to fine details in audio content, and … sense the subtle artificiality that would occur when the reverb we've come to expect is momentarily absent" (Phillips, 2014a, p. 172). The start of a recording containing the reverb tails of previous notes also produces an unpleasant effect when played out of context (p. 173).

An important musical property that cannot be easily modulated during gameplay is music dynamic. Because manipulating the dynamic with crossfades interferes with reverb tails, and sounds unnatural, I chose this challenge as the first test of an imbricate audio system. The distinction between music dynamic levels and volume is an important one, so let us take a moment to examine their differences. Volume is a measurement of loudness, whereas dynamic is a measurement of performance intensity. Playing violin soft or hard has an enormous effect on the timbral qualities of the instrument––an effect that cannot be achieved by simply increasing the volume of a recording (Gauldin, 2004, pp. A2-A3). Below I use the terms *piano, mezzo piano, mezzo forte, forte, and fortissimo* to express dynamic levels in order of increasing intensity, as these are the terms traditionally used in music manuscripts.

> Imbricate audio consists of recording a score with regular pauses to capture the reverb tails of every branching point. Once all desired variations have been recorded, the tracks are divided into chunks, which are queued in the music system. No cut or fade would be audible.

PROPOSED SOLUTION

The solution I propose to this problem is ”imbricate audio”––a process that results in a densely modular matrix of musical “chunks” which can change states quickly while still preserving the integrity of the instrument sounds used (an example video of this is included below on page 9). The aim of creating such a process is to bring some of the flexibility of sequenced music to pre-rendered scores featuring recorded musicians, and/or virtual instruments that imitate recorded instrumentalists. Imbricate audio is an extension of the concept of a standard modular score, but with two important differences: firstly, instead of dividing the score into musical phrases, it is much more densely modular, with divisions every bar. Secondly, the reverb tails are preserved, eliminating the need for a crossfade to smooth out the transition. Together, these modifications grant many of the benefits of modular scores, but with a much quicker response time.

Imbricate audio consists of recording a score with regular pauses to capture the reverb tails of every branching point. Once all desired variations have been recorded, the tracks are divided into chunks, which are queued in the music system (the same can be achieved by creating sequenced music tracks and dividing them into chunks before they are rendered). The chunks are then played back in order, such that the lingering reverb-tail of each overlaps with the beginning of the following chunk. Without any input from the gameplay, the music will play in a linear or looping fashion without any audible signs that it is actually a collection of chunks and not a single, long recording. When the music system receives a cue or trigger from the game, the queuing is reordered.

In the case of Figure 1 (below), after receiving a trigger from the gameplay, the queued mezzo piano version of the following bar (Bar 4) is swapped out for a mezzo forte version of the same bar. Shortly after this starts playing, another instruction is received, and the mezzo forte version of Bar 5 is replaced with a forte recording.

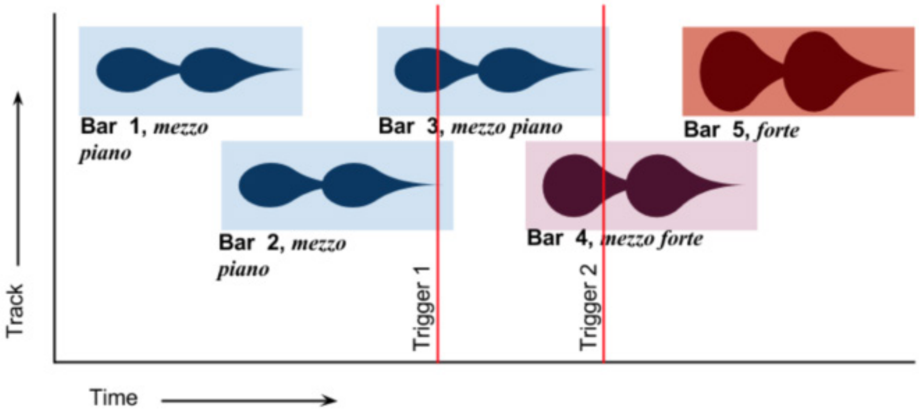


Figure 1 – Imbricate audio in action. Image courtesy of the Author.

From the listener's perspective, the musicians in the recording would be perceived as starting to perform with more intensity in response to an event in the game. No cut or fade would be audible, bringing the adaptive music experience closer to the linear experience of cinema without the distraction of the hallmark fades of game music (Houge, 2012, p. 1).

This is the simplest transition in an imbricate audio system: a trigger is received and the new music state is queued-up for the next branching point. In many contexts this may feel abrupt and sudden, so an intermediate transition chunk can be used. Consider if, in the above example, we ignored *Trigger 2* for the time being. *Instead of Bar 4* being from the mezzo forte music state, we would have a transitional music state, which is positioned between mezzo piano and mezzo forte. Every chunk in this transitional music state would consist of a crescendo from mezzo piano to mezzo forte, and the playback order would now be: *Bar 3, mezzo piano; Bar 4, mp-mf crescendo; Bar 5, mezzo forte.* Although including transitional music states would require considerably greater effort, it could be used for quite complex modulations between different dynamics, keys, tempi, timbres, meters, etc.

EXPERIMENTAL RESULTS

As a test case, I set out to create an adaptive score with the simple one-step process, which would be able to change the musical dynamic less than one second after receiving instruction from the game. For this reason, I composed a score in a 2/4 time signature at a tempo of 130bpm, resulting in each bar being less than one second in duration. I wrote the score to be recorded in its totality at five levels of dynamic intensity, ranging from piano through to fortissimo. Figure 2, below, shows a brief excerpt of the score I gave to the performers, and Figure 3 outlines the performance instructions I gave for each pass of the recordings.

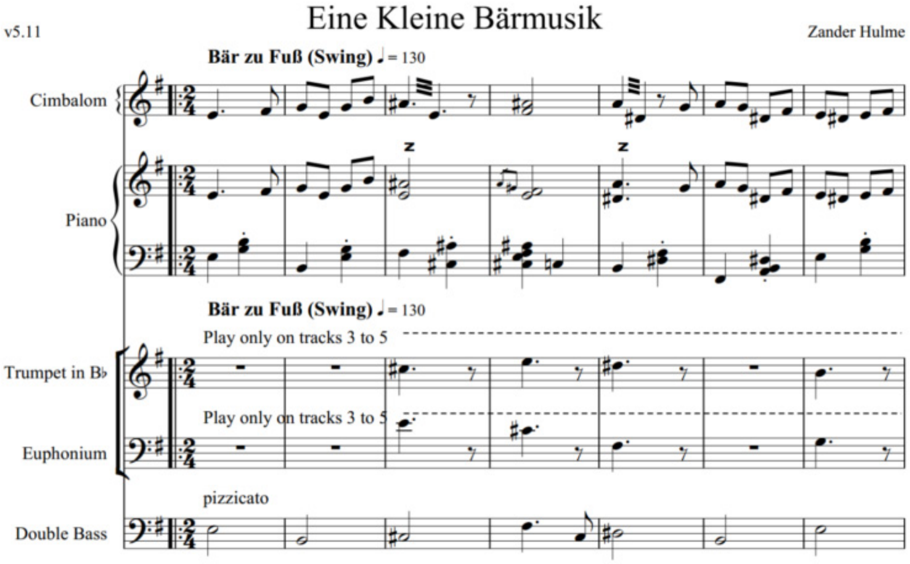
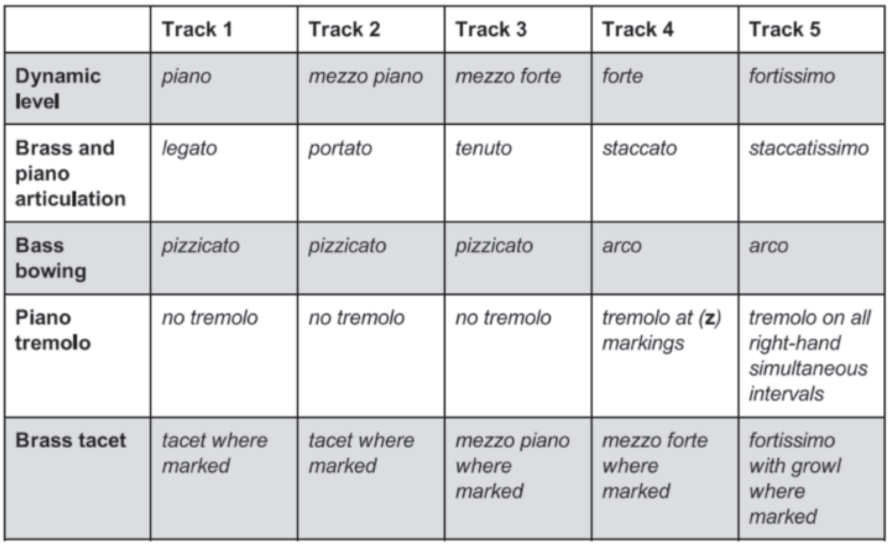

v5.11
Eine Kleine Bärmusik
Zander Hulme
Bär zu Fuß (Swing) ♩= 130
Cimbalom
Piano
z
z
Bär zu Fuß (Swing) ♩= 130
Play only on tracks 3 to 5
Trumpet in B♭
Play only on tracks 3 to 5
Euphonium
pizzicato
Double Bass

Figure 2 – Excerpt from adaptive score for Eine Kleine Bärmusik (Hulme, 2015). Image courtesy of the Author.



	Track 1	Track 2	Track 3	Track 4	Track 5
Dynamic level	*piano*	*mezzo piano*	*mezzo forte*	*forte*	*fortissimo*
Brass and piano articulation	*legato*	*portato*	*tenuto*	*staccato*	*staccatissimo*
Bass bowing	*pizzicato*	*pizzicato*	*pizzicato*	*arco*	*arco*
Piano tremolo	*no tremolo*	*no tremolo*	*no tremolo*	*tremolo at (***z***) markings*	*tremolo on all right-hand simultaneous intervals*
Brass tacet	*tacet where marked*	*tacet where marked*	*mezzo piano where marked*	*mezzo forte where marked*	*fortissimo with growl where marked*

Figure 3 – Performance instructions for recording Eine Kleine Bärmusik.

In the context of "designing a game for music", Richard Stevens and Dave Raybould (2014) discuss the limitations of modular scores (such as the *Monkey Island* example above). They mention the possibility of a process similar to imbricate audio, but quickly dismiss it as "time consuming and unnatural" (p. 152). Having now performed this process, I can attest that it can be "natural-sounding" and the recording process is not much more time-consuming than that of recording a regular modular score. I concede that splitting, rendering, and naming each of the chunks manually was tedious, but the process is simple enough that it would be possible to automate. Additional rehearsals were also required for the musicians to become accustomed to the unusual performance technique.

In contrast to vertical remixing and horizontal resequencing approaches, the composing process for imbricate audio does require a consideration of the outcome of a transition from the end of each and every bar. It also imposes some limitations on the arrangement. For example, particularly long notes that span many bars may cause unintended dissonance. However, imbricate audio's quick responsiveness and lack of audible crossfades make it highly appropriate for some use cases.

An example of imbricate audio can be found in my brief demonstration of *Eine Kleine Bärmusik* attached to a set of buttons and run by a simple C# script in the Unity game engine (Hulme, 2016):

The dynamic intensity of the music is being modulated entirely by software, and although there are some distinct, sudden changes in timbre, these can be attributed to recording or performance mistakes. None of the many cuts in this example are themselves audible, as the natural decay of each chunk overlaps with the next. After 1:45 in the video, the system is instructed to start and stop so that individual chunks can be heard, reverb tails included. This piece is far from perfect, but it serves to demonstrate that imbricate audio is indeed feasible.

To efficiently record these chunks in a way that felt most familiar to the performers, I asked them to play only the odd-numbered bars of the score, treating the even-numbered bars as rests. When this was completed, they were able to listen to their performance of the odd-numbered bars and fill in the gaps by playing only the even-numbered bars. This method also had the unexpected benefit of cutting down on the number of full takes required, as when mistakes were made, a new take could begin only seconds before the error. This process was then repeated for each of the five music dynamic levels. I managed to record the entirety of the piece (5 instrumentalists playing 10 minutes of music each), during one day in the recording studio.

Like Stevens and Raybould (2014), I too had fears that the finished product would sound "unnatural" (p. 152), but I was relieved to discover that it performed seamlessly and adapted to instructions in less than one second as planned. An additional boon was the low processing overhead required to run this system: the simple C# script driving the queue runs in the Unity game engine so efficiently that it is even able to run in the background of mobile games. I have already shipped three games for iOS and Android that run this system, which both play smoothly even on low-end devices.

Fadeless modulations in music dynamics and smooth, fadeless modular score transitions are largely the domain of sequenced music systems, but imbricate audio can bring some of these features to fully-recorded, pre-rendered scores, bringing us one step closer to creating game scores that are indistinguishable from a live ensemble.

CONCLUSION

Imbricate audio can be used to imitate vertical remixing, horizontal resequencing, and other systems by creating a densely modular score made of short chunks. It would be presumptuous to assert that it can serve as a replacement for these approaches in every case, but when it can, it eliminates the need for "conspicuous" crossfades. Imbricate audio may not solve all the challenges that face adaptive game music today, but I hope that the introduction of this system will be of use to composers who want to make innovative adaptive scores. I am still developing imbricate audio, and currently it is the core of a procedural music system that I am designing for an upcoming console title. I hope that other composers will make use of this tool, and combined with the immaculate sound quality now available on consoles and PCs, the power to transition between different music states quickly and smoothly should improve composers' ability to engross the player in the game world. Fadeless modulations in music dynamics (as in the example of *Eine Kleine Bärmusik*) and smooth, fadeless modular score transitions (as in the *Monkey Island 2* scene) are largely the domain of sequenced music systems, but imbricate audio can bring some of these features to fully-recorded, pre-rendered scores. By maintaining reverb tails (and thus the musical integrity of the instruments used), it may even become easier to achieve the elusive goal of player immersion––bringing us one step closer to creating game scores that are indistinguishable from a live ensemble, playing just for you.

REFERENCES

Belinkie, M. (1999, December, 15). *Video Game Music: Not Just Kid Stuff.* Retrieved from http://www.vgmusic.com/information/vgpaper.html

Collins, K. (2005). *From Bits to Hits: Video Games Music Changes its Tune.* Film International, (12), 4–19.

Collins, K. (2008). *Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design* (1st ed.). Cambridge, MA: MIT Press.

Copeland, M. D. (2012). *The Design and Implementation of Sound, Voice and Music in Video Games* (Masters thesis) California State University, Chico.

Gasselseder, H. (2014). *Those Who Played Were Listening to the Music? Immersion and Dynamic Music in the Ludonarrative. In 4th International Workshop on Cognitive Information Processing* (pp. 1–8). Copenhagen: Aalborg University.

Gauldin, R. (2004). *Harmonic Practice In Tonal Music* (2nd ed.). New York City, NY: W. W. Norton & Company.

Houge, B. (2012). *Cell-based Music Organization in Tom Clancy's EndWar. In First International Workshop on Musical Metacreation.* Stanford, CA. Pp. 1–4.

Lerner, N. (2014). *Mario's Dynamic Leaps.* In K. J. Donnelly, W. Gibbons, & N. Lerner (Eds.), *Music In Video Games: Studying*

Play (1st ed., pp. 1–29). New York City, NY: Routledge.

Medina-Gray, E. (2014). Meaningful Modular Combinations: Simultaneous Harp and Environmental Music in Two Legend of Zelda Games. In K. J. Donnelly, W. Gibbons, & N. Lerner (Eds.), *Music In Video Games: Studying Play* (1st ed., pp. 104–121). New York City, NY: Routledge.

Paul, L. J. (2012). Droppin' Science: Video Game Audio Breakdown. In P. Moormann (Ed.), *Music and Game: Perspectives on a Popular Alliance* (pp. 63–80). Berlin: Springerhttp://doi.org/10.1007/978-3-531-18913-0

Phillips, W. (2014a). *A Composer's Guide to Game Music (*1st ed.). Cambridge, MA: MIT Press.

Reiter, U. (2011). Perceived Quality in Game Audio. In M. N. Grimshaw (Ed.), *Game Sound Technology and Player Interaction: Concepts and Developments* (pp. 153–174). Aalborg: IGI Global. http://doi.org/10.4018/978-1-61692-828-5.ch008

Stevens, R., & Raybould, D. (2014). *Designing A Game For Music.* In K. Collins, B. Kapralos, & H. Tessler (Eds.), *The Oxford Handbook Of Interactive Audio* (1st ed., pp. 147–166). New York City, NY: Oxford University Press.

Sweet, M. (2014). *Writing Interactive Music For Video Games: A Composer's Guide* (1st ed.). Upper Saddle River, NJ: Pearson Education.

Taylor, L. N. (2002). *Video games: Perspective, point-of-view, and immersion* (Master's thesis). University of Florida, Gainesville, US.

MUSICAL WORKS

Hulme, Z. (2015). *Eine Kleine Bärmusik* Unpublished manuscript.

Hulme, Z. (2016). *Eine Kleine Bärmusik* (Video of interactive sound recording) (unpublished).

LUDOGRAPHY

Monkey Island 2: LeChuck's Revenge, LucasArts (1991). USA, 1991.

Monkey Island 2: Special Edition LucasArts, USA, 2010)

Space Invaders Nishikado, Taito, Japan, 1978..

Websites:

Firelight Technologies. (2016). FMOD Studio – Why Would You Use It?. Retrieved from http://www.fmod.org/products

Phillips, W. (2014b). *A Composer's Guide to Game Music – Horizontal Resequencing*, Part 1. Retrieved from https://winifredphillips.wordpress.com/2014/04/30/a-composers-guide-to-game-music-horizontal-resequencing-part-1/

Phillips, W. (2014c). *A Composer's Guide to Game Music – Vertical Layering,* Part 2. Retrieved from https://winifredphillips.wordpress.com/2014/03/12/a-composers-guide-to-game-music-vertical-layering-part-2/

Tommasini, G., & Siedlaczek, P. (2016). *Acoustical Sample Modeling Applied to Virtual Instruments.* Retrieved from http://www.samplemodeling.com/en/technology.php

AUDIOVISUAL

Silk, P. (2010). *iMUSE Demonstration 2 – Seamless Transitions.* Retrieved from https://youtu.be/7N41TEcjcvM

AUTHOR'S INFO

Zander Hulme is a freelance game audio designer based in Brisbane, Australia. Specialising in dynamic music systems and mobile games, he is best known for his work on *Steppy Pants*. His film scores have also won awards and screened at renowned international festivals such as Festival de Cannes and Bristol Encounters Short Film Festival.