Droppin' Science: Video Game Audio Breakdown

Leonard J. Paul[1]

1 Introduction

To begin, we can break the title down into two parts; the first part being the term "droppin' science" and the second being the subtitle "video game audio breakdown". In this discussion of interactive music in video games, I have chosen to use the term "interactive music" as I prefer to believe that there should be the possibility of a bi-directional feedback loop with the music and the game rather than simply having the music adapt or react to the game actions. I believe that the audio of a game is part of the interactive feedback loop between the player and the game system. So the notion of labelling it "interactive music" is meant to indicate that the music is part of an interactive system rather than simply interacting with the game itself. If the music doesn't react effectively, or directly suit the game player's actions, it risks become musical wallpaper and can even be turned off with no to little effect on the game play.

"Droppin' science" is a term from rap music which is generally understood to be the process of imparting knowledge or teaching someone through the rap's lyrics and performance style. It is as difficult to argue with the truth of the lyrics as it is with science. In reference to this chapter, it is meant to describe the science or techniques of interactive music in video games. The specific origins of the term are thought to be from the song "My Melody" by Eric B. & Rakim on the album *Paid in Full* in 1987. The first mention in the song are the lines: "I drop science like a scientist, My melody's in a code." I find this reference fitting as interactive music is a combination of the scientific with melody encoded onto the playback system. Due to Rakim's religious beliefs, the phrase can also be interpreted as a reference to the mystic order of his view of the world as well as of the construction of black history.

A "breakdown" in a song is when the instrumental elements are reduced one by one from the mix to reveal the song's structure as well as stripping it down to its essential elements. So, for a video game audio breakdown, I am examining several works within the canon of video game audio - interactive music specifically – and

[1] This chapter is meant to be an introductory look at the colourful history of interactive music in video games. It is based on a presentation given at the A MAZE Festival in Berlin for the "The Future of Music, Games + Art" Symposium at the HBC (Karl-Liebknecht-Straße 9, 10178 Berlin) on January 31, 2010.

P. Moormann (ed.), *Music and Game,* DOI 10.1007/978-3-531-18913-0_3,
© Springer Fachmedien Wiesbaden 2013

highlighting techniques and methods that are exemplary for illustrating effective uses of interactive techniques to enhance game play. The games examined are not meant to be an exhaustive list, but good examples that clearly show principles that are commonly used in interactive music in games. My own first exposure to the term "breakdown" was with the movie *Beat Street Breakdown* from 1984, which featured both break-dancing music and dancing.

So the title is a combination of references from the mid-1980's, in which I was just starting to play video games on the Commodore 64, as well as both being references to black culture as well. It was during this time that I was also being introduced to rap music and break dancing, which has intertwined my first interest in video game music with the rap music of the time. Ten years later, the first video game title I coded for was *NBA Live '95* (1994)*,* which had a funky soundtrack by composer Traz Damji. One of my most recent titles incorporating a funky music style is *NBA Jam* in 2010. For this title, I remixed the music to fit the interactive scheme, edited the music for the interactive music structures, and integrated all the music with my scripting to realize the final interactive music structure for the game.

It is my hope that an investigation of several key video game titles in history will allow me to "drop science" on the topic of interactive music in games. These games are not meant to construct an entire history of interactive music in games, but rather serve as a collection of examples that have a personal resonance to me and clearly illustrate a point.

2 Branching and Layering

To help with defining a few terms first, I will use the notion of both branching and layering to refer to when the game chooses which musical segment to play next and how the game decides which musical layers of a song to play at any given moment. Sometimes these are also referred to as vertical and horizontal approaches respectively. But these terms are potentially confusing as others reverse them. So for the purposes of this discussion, I will use the terms branching and layering as I believe they are less confusing.

To clarify, branching is when the game determines what musical section to branch to when crossing a musical boundary and basing the next piece of music to play on the current state of the game such as the current game world location. In contrast, layering allows the game to define the mix level of each of the tracks (sometimes groups of tracks or stems) of the song based on the game state, such as the health level of the player. At its most simple, two entirely full mixes of a song can be played in synchronization and the game determines the crossfade level between the two (*Chronicles of Riddick*):

> "Basically all action tracks were composed in two versions, one battle track and one sneak track. The sneak track will play when you enter crouch/sneak mode, and will add tension and a feeling that something's going on. Then as soon as the guards spot you, starts shooting at you, or if you attack something, the sneak track will cross fade into the battle track. (...) Aside to this, there is also an exploration music track, which plays at a much more subtle volume, and is completely void of rhythm. It consists of gentle pads, swells, and soft orchestration combined with "sound effect" instruments, and works to give the feel of the area rather than the event taking place."[2]

More complex interactivity would have more tracks and alternate tracks available for mixing in synchronization. For example, the tracks could consist of a separate track for the bass, drums and guitar that can all be mixed at different levels. A more complex scheme could have alternate tracks for each of the instruments that perhaps is used for a double-time feel version of the parts to give the potential of mixing in a denser and more intense feeling mix. With an increasing number of concurrent tracks, an increase of complexity results. Imagine having individual control over each instrument in a layered interactive orchestral soundtrack! These methods are not mutually exclusive and a technique used in *God of War* (2005) was to utilize the strength of rapid branching to respond to game locations while allowing layering by cross-fading between two versions of a full mix to closely follow the intensity of the action. For example, when entering a boss area, a branch to a new segment of music is chosen by the game to underscore the boss, but horizontal mixing is used to modulate the tracks in the new segment to closely follow the player's progress with the boss such as mixing in a more intense version of the score as the boss nears defeat.

Table 1: Interaktive Music Structures

Type	Pros	Cons
Branching	▪ Relatively easy to design a pool of segments which work well together and don't sound too repetitive unless overplayed. ▪ Can create a good musical break between music segments to clearly indicate to the player that they have entered a new location or game state, such as a battle, or that they have been detected by an enemy character. ▪ Branching happens at a logical break in the song.	▪ Complexity issues with using transition segments with a large number of branching segments. ▪ Transitions may be necessary between branching segments. ▪ May have to wait a long time until the next allowable branch point when using custom marker points.

[2] Gustaf Grefberg on *The Chronicles of Riddick: Escape from Butcher Bay* (2004), http://web.archive-org/web/20041116084639/http://www.music4games.net/f_gustafgrefberg_chronicles_of_riddick.html [05/05/2012].

Layering	▪ Smooth musical transitions. ▪ Quick transitions easy. ▪ Technically easy to implement with a multi-streamer. ▪ Can create almost limitless mixes with volume control over a large number of layers.	▪ Many layers can cause bandwidth from storage media. ▪ Many streams take up large amount of storage space. ▪ Musical structure remains consistent for each layer and may become repetitive on repeated plays. ▪ Transitions might not be obvious enough to indicate large changes. ▪ Using multiple versions for each layer can become complicated. ▪ Phasing issues when cross-fading similar material.
Branching + Layering	▪ Best of both worlds of branching and layering techniques.	▪ Worst of both worlds and combinatorial explosion issues, especially if a lot of layers are used.

3 Stingers and Overlays

In addition to branching and layering are stingers and overlays. Stingers are used to punctuate actions in the game by a short musical phrase in the music. In the Monolith game *No One Lives Forever* (2000), audio director and composer Guy Whitmore used short guitar hits and arpeggios in time with the music to signal when enemies had been eliminated. Overlays are typically longer and less percussive musical phrases which also overlay the score in synchronization. Instead of being an entire music track in a horizontal scheme, they are meant to be interchangeable within a certain musical segment to add variation. For both the stingers and overlays, there needs to be a certain amount of quantization defined to ensure that the transition strikes the delicate balance between responsiveness and musicality. At one extreme, we simply transition immediately, which is fast, but relies on luck to have the change sound musical. Rather than attempt to implement a system which knows the tempo of the underlying music, games will often mask the rhythmic shift with a loud-sounding transient effect such as an orchestral stab or a cymbal crash. Middleware systems such as Wwise allow the composer easy access to the definition of the tempo of the music, and the system's ability to make use of it helps to keep things in synchronization.

4 Transition Speed Quantization

At the other end of the extreme are long user-defined phrases or custom markers which allow the music to transition on a longer compositional arc within the song, such as waiting until the end of the drum fill or trumpet solo respectively. One might also wait until the current bar is done before allowing a stinger to play, which makes it sound much more intentional and as though it is a well-integrated section of the score, thus allowing things to respond quickly but, while acknowledging the tempo, quantizing on a beat where the game waits until the current beat is complete before branching to the new segment. If we want the transition to occur quickly, then we can often transition immediately, especially when going from low to higher intensity music.

Table 2: Interactive Music Trasitions – Speed Quantizations

Quantization	Description	Pros	Cons
Immediate	Don't wait, start new event now.	Quick and easy.	Can sound like an error if it is very unmusical.
Beat	Wait until the end of the current beat.	Only a short delay to avoid the possibility of a flammed strike possible with the above transition type (immediate).	No notion of which beat, so a stinger may hit on a strange upbeat. Unlike the above it requires knowledge of both the tempo of the song and when the beat begins.
Bar	Wait until the end of the bar.	Can be more musical than the above.	Sometimes waiting a nearly a measure (worst case) can be a long period of time.
Phrase	Define a phrase length as a number of measures (commonly four) to allow the current phrase to complete.	Can often provide a musical transition but has no notion of musicality like the below.	Not all transitions happen at regularly spaced intervals.
Custom Marker	A defined point at which the music is good for a transition, such as the end of the start, ends of verses, and choruses in a pop song.	Specifically defines musical edit points.	A large amount of work to define all the markers.

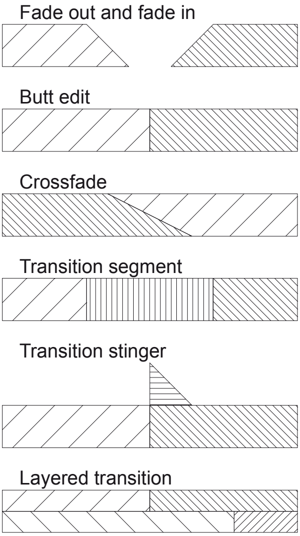
5 Transition Types

The final piece missing in the interactive music puzzle is the transition segment. There are several ways to move from one musical segment to the next. The figure on this page illustrates: fade out and fade in, butt fade (i.e. no fade), crossfade, transition segment, transition stinger and a layered transition. The fade out and fade in transition is a simple transition but often quite effective as it gives the listener a brief respite from the music as the music from the first piece gradually fades out and eventually the next piece of music is faded in. Transitions are used when branching from one section to another directly would compromise the musical integrity of the underscore. This could be when the tempos of the two segments are jarringly dissimilar or when the harmonic structures of the segments do not transition well. In these cases, a separate transition segment of music is used to better transition from one segment to the next. The difficulty with this approach is that it rapidly produces a combinatorial overload. In any branch from one music segment to a different musical segment, the formula is where "n" is the number of transitions and "x" is the number of segments: n = (x * x) – x.

Table 3: Interactive Music Transition Types

Transition	Description	Pros	Cons
Fade out and fade in	Fade out the current music segment then eventually fade in the next music segment.	Simple and only requires one stream.	Not much of an actual transition that can make music sound choppy.
Butt edit	Start playing the next piece of music immediately after current piece completes.	Simple single stream.	Can cause a pop or click if audio data is not correctly prepared or next piece doesn't start immediately.
Crossfade	Fade out current segment while fading in next segment.	Only two streams required and can result in seamless transitions.	Can sound bad if rhythmic, downbeat or harmonic content not equal or material is phase-related and not in phase.
Transition segment	Play a separate transition music segment between current segment and next segment.	Can produce very musical transitions given enough possibilities for transitions.	Combinatorial overflow possible if there are many music segments that require custom transition segments.
Transition stinger	Play a short stinger when the next segment starts.	Easy to implement and only requires one stream if stinger is in RAM which can mask an otherwise poor butt fade transition.	Can sound like a mistake if stinger does not mix well with next segment or underlying material.
Layered transition	Remove one layer, start a similar new layer in next segment and continue until all layers have been replaced.	Can create very complex musical-sounding transitions.	Complexity can be an issue with layer transitions and timings plus the number of required streams can be high.

Figure 1: Transitions



Of course, it is definitely a possibility to reuse transitions, but the matrix of possibilities still needs to be populated. When working on the transitions for *No One Lives Forever,* it was found that going from a slower tempo to a faster tempo needed fewer transitions than going from a faster tempo to a slower tempo. It seems it is easier to jump into a faster tempo than hit a musical speed bump when the tempo decreases too quickly.

6 Generative Scores

Other advanced schemes can be created for interactive music to the degree where real-time track parameters such as a synthesizer's centre frequency can be modulated to match the feel of the game. For generative scores, the entirety of the music from notes to instrumentation and effects can be defined algorithmically using a set of rules. I will focus my attention on the construction of interactive scores that have a large degree of the composer's intent present in the final result. One of the most successful recent games to use a highly generative score is *Spore* (2008), which used rule sets with the music of Brian Eno as a source. In the near future, when streaming bandwidth, storage and DSP power become less of a concern, it can be more easily possible for the composer to create an entire score in their digital audio workstation (DAW) of choice and export the song structures directly to the video game to be immediately modulated.

7 Creative Composition and Editing

One can envision the interactive music behaviour (such as branching, layering and transitions) that can be described by Audiokinetic's Wwise to be described directly in the composer's DAW such as Ableton's Live. Although this method could give "total" control to the composer, it is also a cognitive difficulty of information overload by possibly allowing for the modulation of too many parameters at any given time. In Clint Bajakian's GDC 2010 *Adaptive Music: The Secret Lies within Music Itself* presentation, he clearly draws lines between music composition, music production, music editing and music implementation. I have personally experienced that with my remix and implementation work on EA's *NBA Jam*, that I found having the possibility of playing any role to be potentially overwhelming. I found it much more effective to create or produce and remix the music separately from the other stages, especially implementation. Once I was happy with the composition or the remix of the existing score, I would edit the musical segments to fit the interactive scheme. Once the music integration was defined, I could quickly work at creating and editing new musical assets.

Overall, I believe that having the ability to understand the entire musical pipeline allowed me to be more effective while creating or modifying content. However, I believe that using music that wasn't necessarily composed for interactivity in mind could be very effective in the game as long as the music could be remixed to fit the interactive structure when needed. I personally believe that having a composer focus on the music and having the audio director determine the interactive music behaviour and implementation is a very effective (and common) workflow in interactive game music pipelines. However, it is helpful if the audio director has the final signoff on the music, similar to the music editor in a film production pipeline. Often during the editing stage, difficult decisions to cut material to better focus the work is not necessarily the best decision to be made by the composer.

8 Space Invaders: Interactive Beginnings

Although it is not necessarily the first game to have music that interacted with game play, *Space Invaders* is definitely one of the most popular early games to utilize a soundtrack that changed with game play. *Space Invaders* was released in 1978 and eventually became so popular that in Japan it caused a temporary shortage of the 100 yen coin that was used to operate it. The entire game and all the actual hardware that the game runs on, is the work of one man, Tomohiro Nishikado. Tomohiro clearly considered sound and music to be important to the game, as each sound has its own dedicated hardware instead of a single unit to produce all the sound effects which added to the cost and complexity of the arcade cabinet construction. It had only 1KB of RAM, 7 KB of video RAM and used 8 KB of ROM for the entire game, including all graphics, AI and sound code. Its soundtrack is a simple repeating four-note chromatic half-step descending pattern whose tempo increases as the aliens are eliminated and they descend upon the player's spaceship. While coding the game, he found that the game loop would run faster the fewer aliens that needed to be drawn. He liked the way that the game sped up as aliens were eliminated and thus also sped up the music. The accelerando of the music is similar to the speeding of the player's heartbeat as they are playing the game. In this way the music is not only adapting to the game play, but it is interacting with the player in a tight feedback loop which is also reflected in the visuals. It fulfils the basic requirement of underscoring the increasingly stressful pressing attack by the aliens in a very simple format. The eloquence of the design of *Space Invaders* had even inspired legendary Nintendo game designer Shigiru Miyamoto to become interested in making video games. *Space Invaders* is a classic game for many reasons, including its effective use of interactive audio for its time.

9 Koji Kondo and Super Mario Bros.

The main theme music to *Super Mario Bros.* is arguably the most recognizable music of any video game in history. The music and sound effects were composed by Koji Kondo. Koji Kondo was hired by Nintendo in 1984 and the game came out a year later. The main theme music is an interesting syncopated calypso-style song filtered through the sound of the Nintendo Entertainment System (NES) and Koji Kondo's compositional style. The NES sound capabilities were two pulse waves, a triangle wave, a noise channel and a 1-bit DPCM sampled sound channel that was typically used for sound effects. The NES also had very limited cartridge space and just two kilobytes of RAM which meant that all the music and sound effects for the game needed to be very small, thus the theme song is quite a short loop, yet remains enjoyable due to the style of the rhythms in the lead lines, bass line and drums being on the pulse waves, triangle wave and noise channel respectively. The main intertwined lead melody of the two pulse waves maps perfectly onto the way the steel drum is played with two hands and emits a warm happy feeling of the Caribbean that speaks simultaneously to my computing and cultural background. Although the visual location of *Super Mario Bros.* is set in an imaginary miniature garden where a small Italian plumber is the same size as flowers, the music reflects the happy sunny mood of the player as they explore and interact with the puzzles and pitfalls of the gameplay. The music and soundscape reflects the mood of the actual human player within an imaginary world.

The music changes for each game level, with the underground level sounding more dark and ominous with chromatic sets of notes to give a sharp contrast to the above world music. In the water level, the music becomes more liquid with the use of a waltz rhythm to give the feeling of gently dancing underwater. The castle music leading up to the boss, Bowser, becomes increasingly more intense with scary fast arpeggios. However, not only does the music change when locations change, but also it musically shifts to a new piece of music to indicate that the level has been completed or when the player has become temporarily invincible with the star power. To highlight the amount of change, the music even switches genres with the level complete, the music being similar to a medieval horn fanfare to make you feel like the "king" of the level. Similarly, the star music is a jazzy little number that makes you feel like you're so cool that you can just breeze through all the enemies in the level. So the music serves to score the gamer by rushing through the level as the world blurs past and also gives the important information of how much invincibility time they have left. Sadly, when the invincibility is over, it quickly switches back to the original tune and Mario is back to his regular vulnerable self. When time is running out in gameplay, the music switches to a faster tempo version of the score to spur you on to the end of the level. Even the music when Mario meets an untimely end is less of a dark ending and more of a winding down of the music to a lively fresh beginning.

When presenting his work and philosophy in his presentation "Painting an Interactive Musical Landscape" at the Game Developer's Conference in 2007, Koji Kondo described a key element in his work is balance. In the case of the balance between the sound effects and the music, the sound effects quite literally become part of the musical score, not only because they are in the same key as the music, but also because they steal a pulse wave channel away from the music. The stealing of channels has an interesting effect as the intervals embody a type of redundancy by stealing the second note of the interval and leaving the root to hold the melody. This also has the effect of the score slightly changing each play through in direct response to the interaction with the sound effects which rarely occur at predetermined points in the song and gives the player a sense of interacting with the score with each jump, smashed brick or squashed enemy. From a mix perspective, it also keeps the complexity constant as the sound effects do not add additional audio layers to the mix. Kondo is also very careful in how all the different pieces of music fit together to create different facets of an overall whole within the game: "Game music is made up of the many pieces that appear in a game but it's essential to think of all the music as one piece for the entire game, not simply a collection of individual pieces" (GDC 2007). This holistic approach makes all the music flow together in a seemingly simple fashion to create a very consistent soundscape for the imagined world of the game. In Koji Kondo's design, the entire sound from the NES is the musical score.

10 Koji Kondo after Super Mario Bros.

One of the exciting aspects in working with audio for games is the way in which game design adapts to changes in gaming hardware technology. With the original NES, there was little possibility for any sort of digital effects, similar to echo or chorus, unless you actually simulated the effect with the synthesized voices. For an echo effect, the note would be repeated at a quieter volume until it entirely faded away and a chorus effect could be achieved by slightly modulating a pair of unison notes both on the pulse wave channels with the same duty cycle. With the release of the Super Nintendo Entertainment System (SNES) in 1990, the audio capabilities were expanded to allow for eight sampled sound channels with an echo effect that could be used to simulate a sound's audio reverberation in an environment. *Super Mario World* came out the same year and made full use of the echo effect to simulate the echoes of the sound within the underground levels. In a departure from *Super Mario Bros.*, the underground music is a more minimal version of the above ground music to give more space for the sounds to ring out in the environment and make things more of an acoustic simulation. A welcome return in the above ground music is the sound of a steel drum sounding samples for the lead melodies. With more channels, the rhythm accompaniment was given to the sound of an upright ragtime piano. The combination of steel drum and ragtime gives an interesting blend of the sound of the Caribbean and the slapstick comedy of early films. This combination adds levity to the action of the game and the tempo of the ragtime gives a push to the player during gameplay. With the expanded amount of channels, layering became more of a possibility in *Super Mario World* and you would actually hear an extra bongo percussion pattern when riding Yoshi. When presenting this concept at GDC 2007, Koji Kondo said that he had tried using a branching scheme, but since it was easily possible to hop on and off of Yoshi, he found that this was jarring to gameplay to have it switch between music each time the player decided to use Yoshi as a steed or not: "I could have simply used a different theme here to let players know they had been powered up," he admits, "but that would mean the music would change every time you mount or dismount Yoshi, and that would have disrupted the gameplay."

Once again, there are several subtle key points to show how the audio design interacts with gameplay. Koji Kondo has a strong belief that if the music doesn't synchronize directly with the gameplay, then it creates a disconnect between the player and the game which may even cause the player to turn the sound off entirely. This is very logical as, if the music and sound do little to inform the gameplay and are by nature fairly repetitive, there is then little incentive to continue to listen to the audio of the game.

In *Super Mario 64* (1996) for the Nintendo 64, Koji Kondo utilized layering to a different effect by having a string section be added to the music when underwater. The sound of the stringed instruments was added to give a sense of floating and tranquility. Once you reached the underwater cavern, then the stronger and more percussive sounds of a bass and drums were used to replace the strings and nudge the player to continue through the game. So, in this case, musical layering was used to indicate a location rather than a state as seen above with riding Yoshi in *Super Mario World*.

New Super Mario Bros. (2006) for the Nintendo DS uses the actual rhythm of the music to set when the enemies jump. If one is able to listen closely to the score, one can time Mario's jumps to effectively squash the enemies before they jump. In this final Mario example, we can see how music ties in directly with the design of the game. In this case, the music is truly interactive instead of many scores which simply adapt to the game. There are many modern games which utilize the music as a key interactive element to gameplay.

11 Def Jam: Icon

Def Jam: Icon (2007) could be considered a standard fighting game but there are a few elements in its use of music which help differentiate it from other similar titles. It strives to make the music really come to life by being reflected in the game environment. Similar to *New Super Mario Bros.* the big beats in the music can cause special events to occur in the environment which cause extra damage to the foe you're battling. For example, speakers can pulse so loud that they throw opponents across the screen, and throwing your opponent onto the flames shooting out of a gas station on a particular beat can cause extra damage. In general, the entire scene also modulates to the beat from sparkling hub cabs and falling ceiling tiles to lightning flashes and shaking camera cinematography. Although it only gathered a 69 rating on Metacritic, the visual correspondence with the music is an interesting and effective way to bring the intensity of the music directly into the game play.

12 Rez

Rez (2001) is a unique game which has been described as a "music shooter" and is basically a rail-shooter game with vector graphic-style visuals reminiscent of *Tron* (1982) accompanied by a trance music soundtrack. The thing which sets *Rez* apart from other games is that it attempts to evoke moments of synesthesia between the music and the visuals, and even touch, with the use of the trance vibrator included in the Japanese release. The entire soundtrack is produced by selecting and eliminating the opponents along the path the avatar flies on. The interesting thing is that the game waits to trigger the destruction of the opponents until an appropriate spot in the music in order to create a score where all the music hits are in synchronization with each other. In this fashion, all the level design of the game hinges off the music and, when accelerating toward the end of the level, the music tempo increases as well as the density of the obstacles to increase the difficulty level. The music isn't incredibly varied, but it matches the trance music genre of the game.

13 Vib-Ribbon

An experimental game which also used music as a basis for the level design was *Vib-Ribbon* (2000). *Vib-Ribbon*'s entire level design was based on the game's music, which had the player respond to changes in the music. An interesting feature of this game was that the user could put their own music CD into the PlayStation 2 and the game would generate a course based on features it detected in the music.

Although the music for *Braid* (2008) is not highly interactive in the strictest sense, its use of music to fit the time-warping game play of the game is highly effective. Game designer Jonathan Blow wanted to find music that was cost-effective, high quality and suited the feel of the game play as it directly related to the dynamic rewinding and fast-forwarding of time during game play:

> "I wanted the music to be composed with instruments that worked well with the rewind. When you rewind something, the song should sound interesting going backwards, the texture of the instruments should be different and interesting going backwards and feel different."[3]

The music was chosen from various sources of acoustic-based instrumentals from online label Magnatune. When the player rewinds time, the song goes backwards, and goes forward when fast-forwarding through time. The music gives the direct feedback that time has been warped and sets a similar space to the watercolour-style graphics. This shows how a simple effect of just changing the playback speed and direction of the music can have a strong hand in supporting the game design and gaming experience.

The adaptive music in *Dead Space* (2008) approaches the triggering of the music in a very high-level manner. The game is a horror-survival game based in space in the science fiction genre. The entire sound design and music are combined to simply heighten fear by the use of "fear emitters" that are placed in the level and

[3] http://www.gamespot.com/news/spot-on-the-music-of-braid-6197644 [05/05/2012]

scripted in Lua in order to directly affect the player's sense of fear. With a linear medium, horror films are able to build tension toward a known result, whereas the score of *Dead Space* needed to build the same tension without knowing in advance what the player would do. As with many films and games of the horror genre, the sound design and the score are combined, and audio director Don Veca had the following to say:

> "The plan from the very start was to create mood through overall sound design. We weren't going for traditional music composition or memorable themes, but instead approached the entire soundscape as a single unit that would work together to create a dark and eerie vibe. This is not to say that we didn't use music to help create the atmosphere, because we certainly did; however, the music was used much more texturally than thematically. In this way, *Dead Space* has really blurred the line between music and sound design. When you get right down to it, music is really just sound design with a lot more rules."[4]

Veca puts an emphasis on the importance of the use of scripting to control the dynamics of the sounds and music which is even able to tie into the visual effects of the game. The designers were able to have the high-level design task of adding fear emitters to areas or creatures in the level and the audio system took care of the details:

> "Fear emitters are simply a 'sphere of influence'. However, with this one tool, we can affect a myriad of audio sources, such as music, streamed ambience, adaptive ambience, reverb control, general mixing parameters, or whatever."[5]

The distance values from the fear emitters would be summed and then used to control the parameters for the "creepy ambi patch" for the horror-style sound design and the layers of the music in the adaptive music patch. Using the scripting language, complex interactions could be individually accounted for and implemented without the use of a coder. The additional benefit of using a scripting language is that the scripts could be changed and reloaded while the game was running to help cut down on the turnaround time for changes in audio behaviour. The disadvantage in utilizing a complicated scripting language is that the complexity of the implementation can be beyond the coding capabilities of an average sound designer. The solution to this is to either find a new person specifically as a technical sound designer to deal with complex scripts, but these types of sound designers that also have coding skills such as Don Veca can be difficult to find. There is also often a split between those that are able to formulate their own custom scripts and those

[4] http://www.originalsoundversion.com/dead-space-sound-design-in-space-no-one-can-hear-interns-scream-they-are-dead-interview/ [05/05/2012]

[5] Ibid.

that require a template to work from and are more comfortable tweaking existing parameter values rather than creating original script behaviour.

The adaptive music engine for *Dead Space* was a basic layered scheme which allowed the mix level between four layers at any given moment:

> "For Dead Space, Don invented a "Fear Emitter" that the music reacts to on scalar level, from 1 to 4. Maybe there's a Necromorph waiting just around a corner that has a Fear Emitter attached to it. As you get closer to the corner the music builds, until it's in full frenzy mode when you're right about to turn the corner. However, you change your mind and walk away—the music begins to calm down. But then you stop halfway down the hall and just listen—at that point the music isn't building or getting quieter. It's still really creepy, but it's a stagnant creepy, a "sitting still and not moving" creepy, because that's exactly what the player is doing at that time."[6]

The interactive music design for *Dead Space* can be seen as a clear execution on a key concept: fear. The success of the resulting soundscape is due to the combination of how the game informs the music to modulate the pre-recorded music segments and derives its own compositional structure through the choices it makes through the data set, which results in the particular sound of *Dead Space*. The design supported the interactive possibility of approaching an enemy and having the music react by escalating the fear-factor level and eventually concluding in the frenzied final fear-factor layer which used an aleatoric orchestral score to push the player to even new heights of fear.

14 EA's *NBA Jam*

The adaptive music design for EA's *NBA Jam* (2010) is quite basic in concept but ended up being somewhat complex in execution. The music follows a branching scheme with various custom-branching markers set in the song on bar boundaries to define the start of a branch point and the end of the branch. The branches are between a low full-mix intensity version of the song and a high full-mix intensity version. In general, the two intensities were fairly parallel mixes to reduce complexity, but there was enough variation in their structure to keep things quite different when they looped separately, and often the lower-intensity loop was shorter as it was assumed that it would receive less playing time than the high-intensity loop in most games. The intensity level of the game was set directly by the artificial intelligence (AI) that set the level of intensity for the crowd, which included basic parameters such as difference in score, quarter and other similar variables.

[6] http://www.jasongraves.com/press/pdfs/Dead-Space-Graves-FSM.pdf [05/05/2012]

The songs are chosen by what game mode the player is playing. To mirror the original arcade version, the songs were also switched at half-time to give some extra variation to the underscore. There was no layering performed at all in the design. To allow for additional variation, I typically defined several starting points for looping music segments. This meant that when the low-intensity music started, it would randomly start at approximately five different starting points within the song. So, instead of hearing the same song always start from the same place, it could almost make the one song sound like five, which helped reduce the repetition within the looping music. One additional element was the "on fire mode" for when the player performed enough good moves and scoring to cause the ball to catch on fire. In this case, the game would complete the current bar of music and switch to the "on fire" music segment which looped until the "on fire mode" had ended. When the "on fire mode" was completed, the music would return to the spot it had left off with the music.

EA's adaptive music implementation tool allowed for basic scripting which enabled me to control the variables required to keep track of the states in the music in relationship to the game with a minimum of programmer involvement. However, this also meant that there was additionally a potential for bugs which could, at their worst, cause the music to stop playing. Overall, the game's music had a good balance of changing intensity in response to gameplay but would not oscillate too quickly between the high and low music loops as the levels from the game were quite smooth. I was particularly happy with the "on fire" remixes that I made for the songs that typically contained more dense percussion, additional modulation on synthesized elements and a synthetic fire-sound that would fit between the swells of the music to give the player the feeling as though the music itself was on fire. Overall, it is a basic branching scheme with no layering. The style of branching suits the fast-paced nature of the arcade basketball gameplay and it is commonly able to transition out of a given intensity within a few bars.

The music playback-streaming system and script allowed for ramps on the playback rate of the streams in real-time, so I decided to have it sound as though there were a DJ doing real-time record grabs on the music for certain events. On slam dunks, steals and fouls, the music would do a brief ramp down of the music playback rate and a quick return to the regular rate. This made it sound as though the DJ was grabbing the record and then letting it go, allowing the turntable to start playing the music normally again. I experimented a bit with quantizing and found that quantizing sounded strange and that doing the effect immediately was the most effective in general. The issue being that sometimes there was a great sounding transient that would be scratched and sometimes there was basically no signal which resulted in somewhat of a non-event. When a DJ scratches, they typically have an ear listening to monitor the result and find the transient where, in this case, we simply hoped that a generic scratch event would hit an interesting sound. I had also tried increasing the playback rate of the music to heighten the intensity of the music, similar to a DJ using the pitch control, but found that it compromised the integrity of the music to an unacceptable degree. For big dunks, the music would duck down lower and for longer, which made for a very noticeable music event. When the sound effects were playing, the music basically lowered itself in pitch to get out of the way of the slam-dunk sound effect and vocalization, and became almost a low-end sweetener for the overall dunk sound design. When the game went to testing and the sound effects were turned down, the effect was thought to be a bug, as it made the music almost disappear into a very low-frequency signal for a short period of time. We were able to push it through, which makes for a great overall dunk-effect to differentiate the frequent basket sounds in gameplay. In the end, the game was generally well received and I hope that the design and implementation of the interactive music had a part to play in its positive critical reception.

Interactive music is an exciting new area for creativity and research that has been investigated and explored within interactive entertainment for just a short time and promises new enjoyment and insight as it continues to mature and entertain us in the future.

15 Bibliography

Clint Bajakian (2010): Adaptive Music: The Secret Lies within Music Itself. Presentation at GDC 2010, San Francisco, USA : http://www.gdcvault.com/play/1012601/ Adaptive-Music-The-Secret-Lies [05/05/2012]

Matthew Belinkie (December 15, 1999): Video game music: not just kid stuff, http://www.vgmusic.com/vgpaper.shtml [05/05/2012]

Alexander Brandon (June 2002): Interview with Guy Whitmore, http://www.iasig.org /pubs/interviews/guy_whitmore.shtml [05/05/2012]

Karen Collins (2005): From Bits to Hits: Video Games Music Changes its Tune, Film International. 12, pages 4-19, http://uwaterloo.academia.edu/KarenCollins/Papers/ 198764/_From_Bits_to_Hits_Video_Games_Music_Changes_its_Tune [05/05/2012]

Karen Collins (2008): Game sound: An introduction to the history, theory, and practice of video game music and sound design. Cambridge, Mass: MIT Press.

Chuck Doud, Brad Aldredge, David Murrant & Clint Bajakian (2006) God of War: Sound and Music Production. Roundtable at the Game Developer's Conference 2006, San Francisco, USA.

Jonathan Fish (2003): Interactive and Adaptive Audio for Home Video Game Consoles, http://summit.sfu.ca/system/files/iritems1/8489/b30941970.pdf [Masters of Arts (Communication) Thesis] Simon Fraser University: Vancouver, [05/05/2012]

Rusty James (June 2004): Interview with Gustaf Grefberg, http://web.archive.org/web/ 20041116084639/http://www.music4games.net/f_gustafgrefberg_chronicles_of_riddick.html [05/05/2012]

Damian Kastbauer (August 1, 2011): Limbo – Exclusive Interview with Martin Stig Andersen, http://designingsound.org/2011/08/limbo-exclusive-interview-with-martin-stig-andersen/ [05/05/2012]

James R. Marous (2000): Repairing your Space Invaders or Space Invaders Deluxe Sounds, http://www.brentradio.com/images/SpaceInvaders/si_sound.htm [05/05/ 2012]

Jayson Napolitano (October 7, 20008), Dead Space Sound Design: In Space No One Can Hear Interns Scream. They Are Dead. http://www.originalsoundversion.com/dead-space-sound-design-in-space-no-one-can-hear-interns-scream-they-are-dead-interview/ [05/05/2012]

Jeremy Parish (March 7, 2007): GDC 2007: Mario Maestro Shares His Secrets: Nintendo's Koji Kondo on the interactive approach to music, http://www.1up.com/ news/gdc-2007-mario-maestro-shares [05/05/2012]

Leonard J. Paul (January 31 2010): Droppin' Science : Video Game Audio Breakdown - Interactive Music, http://vimeo.com/10021144 [05/05/2012]

Leonard J. Paul (September 2010): Remix: Composing Structures 4 Play and Performance, http://videogameaudio.com/ECUAD-Sep2010/RemixPresentation-LPaul-AHIS333-ECUAD-Sep2010.pdf [05/05/2012]

Alessandro Scotti (2004): *Space Invaders – Hardware map,* http://www.ascotti.org/ programming/ side/hardware.htm [05/05/2012]

Taito Corporation (1978): Space Invaders Trimline: Manual, http://www.archive.org/ details/ArcadeGameManualSitriml [05/05/2012]

Sophia Tong (Sep 12, 2008): The Music of Braid, http://www.gamespot.com/xbox360/ action/braid/news/6197644/spot-on-the-music-of-braid [05/05/2012]

Various (Thursday, July 19, 2007): 10 Questions for Shigeru Miyamoto, http://www. time.com /time/magazine/article/0,9171,1645158,00.html [05/05/2012]

Guy Whitmore (July 2, 2003): A Spy's Score: A Case Study for No One Lives Forever, http://www.iasig.org/aan/NoOneLivesForever.shtml [05/05/2012]

Guy Whitmore (May 29, 2003): Design With Music In Mind: A Guide to Adaptive Audio for Game Designers, http://www.gamasutra.com/resource_guide/20030528/whitmore_ pfv.htm [05/05/2012]

Tyler Wilde (April 13, 2011): Portal 2's dynamic music - an interview with composer Mike Morasky, and five tracks to listen to now! http://www.gamesradar.com/portal-2s-dynamic-music-an-interview-with-composer-mike-morasky-and-five-tracks-to-listen-to-now/ [05/05/2012]