12-2018

How Sound Effects Affect the Player in Video Games

Wendell Orteza
California State University, Monterey Bay

Wendell Orteza

Senior Capstone

Fall 2018

How Sound Effects Affect the Player in Video Games

Video game sound effects have become an integral part of the experience for players playing a video game. The majority of this paper will compare the concepts introduced in Karen Collins' *Playing with Sound* and ideas introduced in Rob Bridgett's *From the Shadows of Film Sound*. Collins has researched on game sound extensively, writing many books about music and sound design in video games. Bridgett has worked on video game sound in games such as *50 Cent: Blood on the Sand* and *Shadow of the Tomb Raider* while occasionally posting articles for the site *Gamasutra*. Both are veterans in the field of video game audio. As this paper details the concepts presented in both books, the paper will also draw examples of video games that follow the concepts presented. This paper will describe various methods to create sound effects, how they are implemented in a game, and how they affect the player.

The first concept that Collins introduced in *Playing with Sound* is the idea of synchresis. Collins describes synchresis, coined by film sound theorist Michel Chion, as "the fusion of sound and image [that] leads to new meanings that may alter or add to the original meanings of the sound and the image" (20). An example that Collins gave is a stalk of celery and a bone breaking. When the recording of a celery being snapped is associated with a visual of a bone breaking, we may think that sound is actually a bone breaking (20).

To add on to this idea, when we see and hear this bone snap, we may physically react to this. Collins explains that there are special neurons in our brain called mirror neurons. Mirror

neurons allows us to “mirror the action of what we witness as if we are performing that action ourselves… our emotional and neurophysiological states can be directly affected by what we see or hear” (Collins 39). These neurons are important because without these, we would not be able to squirm or react physically and mentally during an action sequence or a sound we hear in a game. It is also the case that we have a stronger reaction to an action that we have experienced in the past (Collins 40). When this happens, it hits those who have experience the action harder than for other people that have not experienced the action before.

Mortal Kombat X does an excellent job at showcasing this idea. When a player inputs and lands a special command called an “x-ray attack,” they are greeted with a short cinematic of their character performing bone-crushing moves to their opponent.

(Unbreakable)

In the picture above, the player sees an x-ray of their opponent’s skull and hears the skull snap from the player’s palm. For players seeing this cinematic for the first time, this can lead them to squirm and anguish over what they have witnessed. The scene would not have much of an

impact if there was no sound that associated with the skull snap. This is because “action-related sounds are associated with an image and an action in our minds” (Collins 40). The sound that the player hears during this scene is a crunchy, snappy sound. When this sound is applied to the skull snap image, then we get a reaction as a result of our mirror neurons activating in our brain. One side note about the sound effect used for the bone snap in this game is that it sounds like a celery being snapped. If this is the case, then it seems that a celery being snapped is the standard for a bone-crushing sound.

Similarly, Bridgett somewhat discusses synchresis in his book, but talks more directly about the concept of ambiguity. Bridgett states, “On a simplistic level, ambiguity can be described as the idea that works of art, particularly sound works, are open to multiple meanings or interpretation” (73). In essence, the definitions of synchresis and ambiguity are interrelated. However, the most important idea for the concept of ambiguity is aural imagery. Bridgett explains:

> [T]he brain also has a deep desire to supply “aural imagery,” finding internal dreamlike associations and semiotic connections, that may not be the cause of the sound, but that are images supplied by the mind of the listener. When the source of a sound cannot be identified, the listener is overcome with the need to imagine causes or physical and visual explanations for that sound. (75)

The idea here is that when a person hears an unknown sound without a visual representation in sight, they tend to imagine images in their mind of what is creating that sound. These images could be things from past experiences that make a similar sound to what they have heard.

However, the things people imagine to associate with that unknown sound vary from person to person. One's own interpretation of a sound can be different from another person.

Aural imagery comes with two classes: objective aural image and ambiguous aural image. Objective aural images are "sounds which form part of a readily recognizable semiotic fabric, those with which an audience seeped in relevant cultural references will have no problem identifying: a typewriter, dog barking, passing cars, gun shots" (Bridgett 76-77). Basically, if we hear a dog bark offscreen, we automatically assume that it was a dog that barked because we recognize the sonic make-up of a dog's bark thanks to past experiences and cultural notions.

On the other hand, ambiguous aural images are "deliberately ambiguous and strange sounds… [that] short-circuit this concept of the culturally dependent notions of semiotics and the objects they represent" (Bridgett 77). Let's say that the dog's bark has been altered in some way using effects like a phaser, chorus, pitch shift, or other audio processing. We may hear a bark, but it has been altered in a way where we cannot recognize that it was a dog that barked. Instead, we hear a noise unfamiliar to us that begs the listener to find out what that sound is and what caused it.

An example for ambiguous aural image is the unknown shriek of a monster in the first level of *Bloodborne*. When a player begins their first playthrough of *Bloodborne*, they are greeted with a one-time event in the first level of the game. The player needs to climb up a long ladder in order to advance through the level. As the player climbs up the ladder, they hear a terrifying shriek of an unknown monster. What is clever about this event is that the shriek is heard to the left. This event serves two purposes. One is that for many people hearing this sound for the first time, it begs the question of who or what caused that noise. Secondly, if the player

decides to rotate the camera to the left, they see a bridge in the distance. This is a way for the game developer to tell the player that the shriek might have come from the bridge. If the player is able to traverse to the area and reach the bridge, they are greeted with an optional boss in the game. As a result, the question that has been on the player's mind -"what caused that sound?" - is answered. They now know that the shriek came from an optional boss.

In a video titled *The Sound of Uncharted 4: A Thief's End*, Senior Sound Designer for the game studio Naughty Dog Jeremy Rogers describes creating sounds for the grappling hook. Roger explains:

> What's cool is I actually went and bought a David Morgan bullwhip, which is the same bullwhip maker that did the Indiana Jones whip that Ben Burtt, big sound designer, used for all the whip cracks and whooshes and everything. So I went out and bought a David Morgan whip because the grappling hook is sort of, like, a "whip" in this game. We did a whole series of recordings with that, and because of that, I think it's an homage to *Indiana Jones*, and I think it becomes one of the iconic sounds of this *Uncharted* game. (10:02 - 10:47)

This is an interesting, creative choice to have a grappling hook sound like a bullwhip. In another interview, Rogers states, "we decided that a bullwhip was the best and most diverse sound for this particular tool" (Andersen). In real life, a grappling hook does not sound like a bullwhip. However, the intention for the grappling hook to sound like a bullwhip is to pay homage to *Indiana Jones* and make the player feel like they are Indiana Jones when using it. The grappling hook in the game serves as a tool to traverse through areas, and to latch and pull crates from unreachable areas. Much like the movie, Indiana Jones uses his bullwhip to traverse through

obstacles by latching onto something like a tree and swing across, and also used to capture or disarm enemies. Although a grappling hook and a bullwhip are two different things sound-wise, the main point in using them within their respective context interrelate. With this, it is easy to understand the inspiration why Rogers used a bullwhip as the sound effect for the grappling hook.

The concepts of synchresis and ambiguity are not only used for objects like the grappling hook in *Uncharted 4*. A video titled *The Sound and Music of The Last of Us* shows how the sound effect for the enemy, the Clicker, is created. The Clicker is a blind enemy that uses echolocation in order to navigate its surroundings. To create its sound, a voice actress produced a tight, clicking noise using the back of her throat reminiscent to a pig squeal. To Audio Lead Phillip Kovats and Senior Sound Designer Derrick Espino, this peculiar sound effect captured what they wanted the Clicker to sound like. Neil Druckmann, the Creative Director in *The Last of Us*, comments how this sound effect is used in-game:

> Here's this benign, clicking sound, and on its own isn't very creepy, but put in this other context and you hear this sound, and it has this different symbolism in the game, and it becomes this fear factor. And again, using audio, you don't know where this is, and you hear this clicking sound. It's echoing down the hallway and people get very scared when they hear this audio cue. (8:21 - 8:41)

From this comment alone, we see the concept of synchresis/ambiguity in full effect and how a sound effect affects a player. The clicking sound, as described by Druckmann, is harmless and not creepy. However, when this clicking sound is associated with an enemy like the Clicker, not only does this give new meaning to both the clicking noise and the Clicker, but now the player

knows that this clicking sound belongs to the Clicker. When a player navigates an area and hears the Clicker in the background, it changes the way they play. Now, they have to be extra cautious because they do not want to make sudden movements and must think of creative ways to navigate past them. Also, with 3D audio positioning, the player can use that to their advantage if they cannot see where the Clicker is.

These types of sound cues are very typical for players to identify enemies in games. Another game that does this well is *Left 4 Dead*. *Left 4 Dead* puts the player in a zombie apocalypse, and the goal in the game is to escape. Aside from the typical zombie sound effects, there are five special infected zombies that have their own special sound effects and music themes. They are the Hunter, the Smoker, the Boomer, the Tank, and the Witch. The Hunter makes a growling noise; the Smoker makes a coughing noise; the Boomer makes a sort of grumble, vomit noise; the Tank makes a sort of roaring noise; and the Witch makes a crying noise (marksmanman96). Since each special infected have their own unique sound effect, it allows players to easily distinguish and be able to prepare to fight back.

The second concept that Collins introduced is kinesonic synchresis. This idea focuses on how sounds are fused with action instead of image. There are two ideas that Collins lists for kinesonic synchresis: player-generated event and game-generated event. A player-generated event is “an event that the player initiates (for instance, by clicking a mouse or by pressing a controller button)” (Collins 32). There are many examples for player-generated events. An example that Collins gives is if a player presses the jump button, they may hear Mario jumping (32).

Another example of a player-generated event is going through the menu of a game. When the player goes through the menu, the player will hear different confirmation noises depending on what they choose. When a player goes up and down the menu screen for the game *Chime*, they hear music notes (Menus). What is interesting is about these notes is that they play into the composition of the main menu theme. These music notes essentially become an interactive part that the player can partake in. As a result, players can create a wide array of renditions for the menu theme. The main point here is that player-generated events allows the game to give feedback to the player for their actions, thus making these sound effects interactive.

On the other hand, a game-generated event is "an event initiated by the game's algorithms, such as the control of a nonplaying character, a timer-related action, and so on" (Collins 32). One of many examples of this would be NPC (non-playable character) fights in *Bloodborne*. By doing a specific questline for a specific character named Eileen, the player has to take down an NPC that has essentially lost their mind. Once the player takes out a certain amount of health from the hostile NPC, Eileen will come and help the player. This is a game-generated event because this only happens by completing the prerequisites that lead to this event, and Eileen will appear to help once the hostile NPC is at a percentage of health.

Collins states, "player-generated events are always interactive, but game-generated events can be interactive or noninteractive" (32). However, Collins did not give an example of game-generated events with sound effects that can be interactive. On the other hand, Bridgett does talk about game-generated events that can be interactive in the form of ambiance. Typically, ambiance in games helps paint the world around the player. In the first level of *Bloodborne*, the player can hear screams, coughs, and maniacal laughs behind closed doors. These sound effects

should tell the player that something obviously wrong is happening that is affecting the people in the city. However, these sound effects are simply loops. The player cannot directly interact with these loops. When the player interacts with the doors, the loop pauses and the player receives dialogue from whoever is inside. After the dialogue ends, the loop resumes again.

An example that Bridgett gave in which the player can interact with ambiance is in sports games. In many sports games like *Madden NFL 17*, the crowd reacts to what the players does. If the player scores a touchdown, then the crowd would start to cheer loudly. However, if the player commits a foul, then the crowd would start to boo. Although these crowd reactions may or may not encourage players to play a certain way, one other importance to this is to recreate what you would normally hear in a real life sports game on TV. Not only do you hear the crowd reactions, but you also hear play-by-play commentary and the football player's callouts in-game. With these ambiances, it makes the player feel like they are watching a real life football game on TV. On top of that, if someone watches or passes by, they might be fooled to think that a real football game is televised due to how realistic the ambiances are to the real thing. All in all, these reactions from the crowd in sports games are game-generated event sound effects.

Finally, the third concept is kinesonic congruity. Kinesonic congruity is the idea where there is congruence between the player's gesture and the sound generated. In the video *The Sound in Uncharted 4: A Thief's End*, the sound design team shares that they went through great lengths to include what Senior Programmer Jonathan Lanier calls a "dynamic foliage system." Lanier comments:

> In previous games, like if the player walked on dirt, we play dirt. If the player walked on grass, we play grass, and that was the beginning and end of it. In this game, we actually

> tie in to the artwork for the background foliage. So, if you're walking through a bush, we play the sound of a bush and that sound is completely dynamic and variable depending on how fast you walk through the bush, what kind of bush it is. And depending on how fast you move through it, you get a slightly different sound. And it just adds a little thin layer of detail that's almost too easy to miss, but it's just amazing how much it connects the player with the space. (9:16 - 9:56)

The sound design team for this game wanted to go with a hyper-realistic approach when it comes to sound. With this dynamic foliage system, it makes the player be more immersed in the game. Unfortunately, the video does not showcase the sound design team recording multiple dynamic takes of moving through a bush for example. However, the video does show how they recorded the sound effects for the Jeep. The video shows that the team recorded a Jeep as it climbs up through rocky terrain and on a pad used for dyno testing cars to record the engine loops. It also shows the team recording various sounds when a Jeep crashes or gets stuck. After this stage, they moved on to Foley to record different surfaces for the tires.

Senior Sound Designer Jeremy Rogers says that *Uncharted 4* "is the biggest Naughty Dog game to date, and the sound grew in both size and demands… the biggest challenge was figuring out what we could do to push audio in games to the next level, but also come out with a product in a finite amount of time" (Andersen). Time and memory are some of the big issues that a team faces when creating a game. As much as a team wants to add to their game, they have to reach a deadline. Also, how much memory they have to work with is also a factor because there is only a finite amount of resources that a team can put in a game before it starts to slow down in

frame rate. Unfortunately, the sound design team for *Uncharted 4* did not go into detail on how they manage to fit everything into their game in terms of sound.

Uncharted 4's take on kinesonic congruity is on a much larger scale with its dynamic foliage system. Since many games do not have a dynamic foliage system, they are content with if a player walks on grass, they hear a grass footstep and so forth. However, Collins talks about kinesonic incongruity where there is no relation between the player's gesture and the sound generated. An example of kinesonic incongruity is a forest path leading to a village in *Bloodborne*. There is a specific root of a tree where if the player walks on it, they would hear the same sound effect used when walking on a wooden floor in the game. Not only is this clearly not the right sound effect to use, but it could potentially throw the player off. If the player hears this misplaced wooden floor sound effect for that specific root of the tree, they might think there is some sort of hatch on the floor that leads to another area.

In conclusion, Collins' book presents concepts such as synchresis, kinesonic synchresis, and kinesonic congruity that show how these three things affect a player. Bridgett's book describes ambiguity and aural imagery that interrelate with synchresis and how ambiance can be an interactive game-generated event. However, Bridgett's book after that point becomes a how-to guide on how to deal with ear fatigue and things of that nature. Although the how-to guide is important, it does not answer the question on how sound effects affect a player. Collins' book, especially the first chapter, explains in great detail how sound effects affect a player with concepts that this paper took. That being the case, one thing that both Collins and Bridgett share is that their ideas are seen across a large selection of games.

For my creative project, I made my own game in *RPG Maker MV* from Steam. *RPG Maker MV* is a design tool to easily create games for people who have no experience in coding. Although there is no built-in tutorial to how to properly use the tool, there are guides on *YouTube* and forums that can easily assist people. The sounds I used to implement into my game came from a site called *Freesound*. This site is essentially a huge public domain sound library that allows users from all over the world to share sounds, and people, such as myself, can download and use them to their liking. The DAW, or digital audio workstation program, I used to edit these sound effects is FL Studio 12.

My main sound design goal for my creative project was to add realistic footstep sounds and background ambiances. I was curious to see how far I could go in terms of adding sound to this game-making tool. Something that I really wanted to see for myself was if this software allows adding positional audio like, hearing a waterfall gradually get loud as the player approaches it. I thought about creating my own sound effects, but adding positional audio was more important to me. So many games today have positional audio; and positional audio and ambiance is what got me to write about this topic in the first place.

As I worked on this creative project, one thing I realized is how limited *RPG Maker MV* is in terms of sound. By default, you can add background music and ambiance very easily. However, if you want to add things such as footstep sounds or positional audio, you have to download plugins in the form of javascripts and place them in your game folder. With these plugins, they essentially open up doors for endless creativity.

The image above is the beginning level of my game. One of the first things I did after I created my map was to add footstep sounds. I downloaded grass, dirt, wood, and stone footsteps from freesound.org. After I downloaded, I used FL Studio 12 to convert the files to .ogg and placed them into the audio folder. To be able to add footstep sounds, I had to install a plugin called RegionEvents by YEP that allows me to add footsteps. As you can see in the picture above, there are different shaded tiles representing the grass, dirt, and bridge. The olive green tile represents a grass footstep; the light red tile represents a dirt footstep; and the mahogany tile represents a wood footstep. Whenever the player steps on these tiles, they would hear a footstep corresponding to the surface. One thing about this plugin is that I am able to change the pitch of each footstep but cannot change the attack rate. For each footstep, I decided to have two different pitches. This allows for variation of footstep pitches, making it a little more realistic instead of playing the same pitched footstep sound over and over again.

However, with these different pitched footsteps comes a problem. For the most part, these footstep noises sound good when you only walk. If you decide to dash, then the footsteps sound unnatural especially since each individual tile you step on triggers a footstep. Due to how fast

you dash in *RPG Maker MV*, you are going to step on these tiles really fast. After some research, it is the game's speed that is causing this issue. There were answers to solve this, but they were usually vague or in some forums I saw, were left unsolved. Another thing to point out was there was no video tutorial online either, so that made it worse. At the end of the day, I wanted to move on what I wanted to do, which is to add ambiances.

After I dealt with that issue, I moved on to creating sound effects when the player enters a house. First thing I did was add doors to all the houses. One cool thing I found out is that the door I used came with a built-in event that opens when the player interacts with it. Not only does it already have a built-in event, but it even had sound effects that came with it. The sound effects that came with it was a sound of a door opening and footstep sounds. I like how there was already sound effects in place, but I did not like the sound effects used, especially the default footstep. The default footstep sounded very artificial, which is something that goes against my goal for this creative project.

There are three houses and an armory shop. The three houses have a regular wooden door and wooden floor, while the armory has a heavy door and stone floor. For houses that have a regular wooden door, I downloaded a light wooden door sound to use when opened. For the sound effect when walking inside, I took the wooden floor sound effect that I used earlier, and spliced it in a way where the footsteps sound faster. For the armory, I downloaded a heavy wooden door sound when opened, and applied the same method with the wood footstep sound to the stone footstep. After I got comfortable adding these footstep transitional sounds, I added a dirt footstep transitional sound effect when the player leaves the house and goes outside. After this, I added a river ambiance that I downloaded and background music provided by the

software. One thing I would like to point out for the river ambiance is I made sure it is not too loud. Rather, I want the river ambiance to be soft to signify that it is a nice, gentle river streaming through this small town I created. To me, it gives a sense of peacefulness to the town.

After I added footsteps and river ambiance to the first map, I started to work on the second map. After many revisions to this map, the picture above is the end result. For this map, I wanted enemies to spawn so that player can fight. However, this also came with another problem. The problem I faced is that there were no enemy encounters after I added footsteps. After some googling, I found out that the tiles where I placed footstep sounds disabled enemy encounters. There was a fix in the main site of where I downloaded the RegionEvents plugin that I had to add a script. After I added the script to each of the footstep tiles I used, I was able to have enemy encounter.

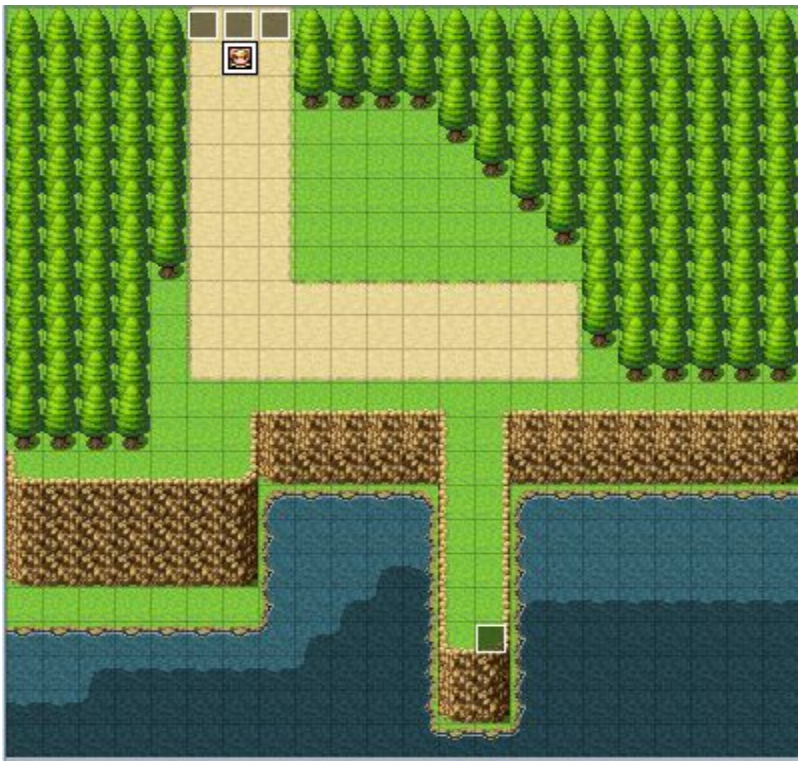
Once I fixed the enemy encounter issue, I added ambiance. At first, I thought about adding wind ambiance to the map. I wanted to go with a peaceful, gentle breeze approach for this map. However, a couple problems rose. One problem was that the wind ambiance that came with

the software was extremely bad. Second, I tried finding some wind ambiance, but there were not that great either. Finally, since this map had enemy encounters, I felt it would go against my peaceful concept. Instead, I downloaded birds chirping in a forest ambiance. With this ambiance, it makes the map give a sense of wildlife in the background and to make the map sound alive. I really liked this design choice and the ambiance I downloaded, so I kept it. After this, I added background music provided by the software.

At this point, I wanted to see if I could add some positional audio. After some googling, I found a plugin called AudioManager Extension by Javahut which allowed me to do so. For the second map, I made a waterfall specifically for this. At first, I had some trouble understanding how to properly use the plugin. Luckily, there was a demo for this plugin provided by Javahut to download and see how he used the plugin (Javahut). After reviewing the demo and checking the events used, I was able to understand how to use the plugin.

As you can see, there is a tile with white outlines between the waterfall and the bridge. This tile is an event where the waterfall sound effect plays. With the special tag <dynamic> (see image below) placed in the notes, the player will hear the waterfall to the right when they reach where I have the character model situated in the map.

It sounds soft at first, but when the player slowly approaches to the waterfall, it gradually gets louder. The player will fully hear the waterfall on both ears when they reach to where I have the event set up in between the waterfall and the bridge.



Finally, the last thing I wanted to add was an ocean ambiance. Since I understand how to use the plugin, I created the last map for this project. As you can see, there is an event in white outlines near the bottom. What is special about this event are what is written on the notes.

This special note essentially makes it so the player will always hear the ocean ambiance no matter where they are in the map. In the last map, you have to be in a specific range to be able to hear the waterfall. With this tag, the player is always in range. The ocean ambiance gets louder once they reach down to the map to where the event is located. The ocean ambiance I used for this map is the one the software provided. I actually tried finding some ocean ambiance in freesound, but they were not the great compared to what was available in the software.

I also want to point out that I did not add music to this map compared to the others. Although this is a proof of concept, the approach I wanted here is to make this area important. This is something that I learned when I played *Pokémon Diamond*. Near the middle to the end of the main story in *Pokémon Diamond*, the player has to reach to the top of a mountain to fight against a legendary Pokémon. When they reach to the top, there is no music; only the wind blowing, signifying a great deal of important in this area. If I ever decide to work on this project in the future, I would certainly add an NPC or a special cutscene of great importance in this map.

After I added all the ambiances and footsteps I wanted, I changed the sound effects used for scrolling up and down the menu, the “OK” and “Cancel” sounds, and the sound effect used to escape in a battle. I already achieved what I wanted to add into my game with ambiances, so this

was mainly just for fun. I downloaded sound effects for each of the things I listed, and replaced the default sound effects to the ones I downloaded. When I tested the game out, I was surprised how well the downloaded sound effects worked. As a result, it definitely got me curious how the creators of the sound effects were able to achieve the sound effect they created.

That is basically the step-by-step process for making this proof of concept. As I worked on this creative project, it gave me a good understanding on how to implement sound effects. It also gave me a better understanding on how to use the plugins I used. While I did not create any of my own sound effects, it is not to say that I will not make sound effects in the future. Background ambiances and realistic footsteps were important in this creative project. Of course, there are many other ambiances out there that I was not able include like how a chest or footstep would sound like inside a cave. Even though this might be the case, I was able to achieve my main goal of adding background ambiances and realistic footsteps.

References

Andersen, Asbjoern. "Designing the Sound for Critically-Acclaimed UNCHARTED 4 - with Senior Sound Designer Jeremy Rogers:" *A Sound Effect*, 7 June 2016, www.asoundeffect.com/uncharted-4-sound/.

Bloodborne. Playstation 4 version, FromSoftware, 2015.

Bridgett, Rob. *From the Shadows of Film Sound: Cinematic Production & Creative Process in Video Game Audio: Collected Publications 2000-2010*. R. Bridgett, 2010.

Collins, Karen. *Playing with Sound : A Theory of Interacting with Sound and Music in Video Games, MIT Press, 2013*. ProQuest Ebook Central, https://ebookcentral.proquest.com/lib/csumb/detail.action?docID=3339561.

Javahut. "MV 1.3 - JavaHut's AudioManager Extension (1st Plugin)." *RPG Maker MV*, 24 Dec. 2018, rpgmakermv.co/resources/javahuts-audiomanager-extension-1st-plugin.1070/.

Madden NFL 17. Playstation 4 version, EA Tiburon, 2016.

marksmanman96. *YouTube*, YouTube, 8 Sept. 2009, www.youtube.com/watch?v=_npS_-NfxWk.

"Menus." *Menus - SoundInGames.com - Sound Design in Games*, www.soundingames.com/index.php?title=Menus.

Pokémon Diamond Version. Nintendo DS version, Game Freak, 2007.

Super Mario Bros. NES version, Nintendo Creative Department, 1985.

Super Mario Odyssey. Nintendo Switch version, Nintendo EPD, 2017.

"The Sound and Music of The Last of Us." *SoundWorks Collection*, soundworkscollection.com/videos/the-sound-and-music-of-the-last-of-us.

"The Sound of Uncharted 4: A Thief's End." *SoundWorks Collection*, soundworkscollection.com/videos/the-sound-of-uncharted-4-a-thief-s-end.

Unbreakable. "Mortal Kombat XL - All X-Ray Attacks." *YouTube*, YouTube, 21 Nov. 2016, www.youtube.com/watch?v=AMk5veiJbGU.