5-2020

Memory Management for Game Audio Development

Rahmin Tehrani
California State University, Monterey Bay

Memory Management for Game Audio Development

by Rahmin Tehrani

April 15th, 2020

Music & Performing Arts Departments
@
California State University, Monterey Bay

Introduction

The amount of audio files used in a video game can range from a few to thousands. One of the most important aspects of video game audio implementation is the designer's ability to manage these files using the least amount of space as possible. A sound designer's ability to efficiently deliver audio assets into a game by using a range of strategies has developed overtime in a way that still allows them to create a vast world of sound. Some of these strategies include using one sound (file) for multiple effects, using variations of multiple sounds together, hiding sound from a particular level or section in which that sound is not being used, and converting audio file formats and specs. Learning to implement these strategies requires knowledge in audio production, video games, and the technology related to these topics.

I have studied these areas for the last few years, and will explore memory management for game audio over the coming months on a project being developed titled "Kiko." I will be using all of the strategies listed to help reduce the amount of (audio) memory used in the game. Some of these strategies will be more effective than others. A sound designer can take away many important ideas learning about game audio memory management, allowing them to create sonically effective video games while using less space. The majority of memory in video games is given to larger, more consciously noticeable aspects to the average video game player such as graphics and programmable content like character movements. Therefore, audio is given the least amount of space out of all major areas of a game.

Technological Considerations

Developing sound for video games is an intricate process that allows a game to blossom from a simple first-person shooter into a vast, immersive role-playing game developed by many talented specialists. There are a wide variety of tools and techniques used by music producers and sound designers that help them create (sonically) detailed video games with beautifully designed worlds. Programs such as Audiokinetic's *Wwise* or Firelight Technologies' *FMOD* are dedicated audio middleware, which designers use to implement music and other audio into game engines such as *Unity* or Epic Games' *Unreal*. Sound designers who use programs like *Wwise* can also effectively edit and mix music and audio in ways meant specifically for video games. This can occur after they record and edit music and audio in their conventional digital audio workstation (or *DAW*) such as Avid's *Pro Tools*. Sound is then implemented directly into a game engine, making it easier than ever for a sound designer to focus on creative aspects.

Video game music and audio production requires different strategies than traditional standalone music recordings or music and audio for film and television, although all are similar in many ways (Collins 88). Since video game music and audio are interactive, sound can be triggered, for example, by a player pressing a button, or by a character entering a specific area or level. This means that music and audio in games is not always linear, so it does not follow a strict timeline. Playback of specific sound needs to be randomized and triggered at specific times according to the

circumstances of the game (and how the developers design the game). This can make things intriguing and immersive for a player, but it makes the sound designer's job require a unique approach to working with sound.

The Role of Sound in Video Games

Sound in video games has an important role which adds a lot to the user experience. Audio cues give a player hints which allow them to progress in a game, the specifics of which are up to the developers. Cues can range from a single sound effect to longer musical segments. Sound also provides instant feedback to players, helping them understand what they have done right or wrong. Additionally, music is a strong emotional motivator in video games, much like it is in film and television (Lovato).

Production Process

Game music and audio goes through a few stages of production, starting with pre-production which can involve creating an audio design document and collecting assets. An audio design document details the design and implementation of a game's audio. This can include things like the general game type and/or genre, music to reference, and spotting specific audio cue point entrances and exits (when and where sound begins and ends). Assets can include music, dialogue, recorded sound effects and other audio samples (Collins 88). The next stage, the production stage, involves

recording and synthesizing music and audio. Producers may record dialogue, music, sound effects, and any other sound needed for the game (Collins 95). Finally, the post-production stage involves editing, mixing, mastering, and any other finishing touches (Collins 102).

Throughout a game's development, the audio team must always be aware of memory usage. They need to pay close attention to the maximum data usage limit prescribed by the entire development team at the beginning of the pre-production process. In *Aaron Marks' Complete Guide to Game Audio*, Marks says "Find out ahead of time what physical memory restrictions your audio will face and plan accordingly." He then explains that "If 60 minutes of direct audio has to squeeze into 10 MB, you'll probably have to down-sample, adjust the resolution, reduce to a single channel, and save them in a compressed format." Marks says this is an ugly example, but gives an extreme idea of what a sound designer needs to keep in mind. Designers will focus on and implement the previously described techniques throughout the entire production process to ensure all music and audio remains present and essential to the game while still remaining under the maximum data limit.

Using One Sound for Multiple Effects

Sound has many different effects on a player in a game. One sound effect is often used for multiple applications (Audiokinetic.com). A knock on a door can also be

used for something like a thrown rock against wood. The ability to use one sound for multiple applications saves tons of memory in a game's development. Additionally, this technique allows a sound designer to create and experiment with colorful new sounds. This form of audio data management starts early in the production process and provides the best opportunity to reduce memory usage in game audio development.

There are a few reasons a sound designer is able to use one sound for multiple applications. Sound effects that already sound similar are the easiest example of this technique. The sound of rain can also be used for a waterfall, or a sink. A footstep sound can be used for other clicking or tapping effects. Some sound effects are often manipulated to create a new sound, or slightly different versions of an original sound effect. In *Wwise's* 101 lesson, one can find that just a single shotgun shell sound can be used to sound like a few with manipulation.

The reason a sound designer can use one audio file for different effects is because the game engine will only save one instance of an audio file, even if it is copied, manipulated, or edited and used for a different event. If every single bullet shell, footstep, or breath used multiple audio files, there would not be enough space for other important aspects of a game, such as graphics or movements.

Mixing and Editing

Mixing and editing sound is a common practice in all forms of music and audio production. In music, a producer can use editing and mixing techniques to create unique

versions of their original recordings. These techniques are reproduced in game sound to add interesting textures, which simultaneously saves space. Editing refers to things like cutting, copying, and redistributing sound to create noticeable and unnoticeable changes. Editing does not typically refer to any sonic changes to the original recorded sound. Mixing refers to the sonic alteration of a recorded sound, such as filtering to change pitch, reverberation to add room sound, or distortion to create a louder, more aggressive sound effect. The terms mixing and editing are often used interchangeably, and rightfully so. These two practices are often done at the same time, and go hand-in-hand when working with sound.

Editing happens often in video game audio development. Just a simple cut off the end of a short, one second sound effect can create a world of difference. The beginning of a sound can be cut as well, creating an even more interesting texture. As well as cutting the beginning and end of a sound, a designer can fade a sound in or out. In fact, an entire sound can be manipulated just using volume changes throughout the duration of the sound. All of these techniques can be used together as well, giving a sound designer the ability to change many aspects of a sound with nothing but simple tweaks.

Mixing techniques range from quick and simple adjustments to long and complicated adjustments. One of the simplest forms of mixing a sound effect is to change its pitch. If a sound designer used the same instance of a footstep sound effect, a player would notice very quickly, and become easily distracted and irritated. If one footstep sound was used, but with multiple instances of the sound at higher and lower pitches, the repetitiveness would not be as noticeable. *Wwise* and other audio

middleware allow real-time randomization of aspects like pitch, and give the sound designer the ability to set a minimum and maximum value for them. This quick and simple adjustment creates much more opportunity for variation when trying to save space.

Changing the volume level of a sound can add or take away a lot from it. For example, quiet gunshots sound further away than loud gunshots. This eliminates the need for importing an entirely new, quieter gunshot sound. In fact, any sound at a lower volume appears to be further away. To get the most out of volume changes, a sound designer will adjust the volume level of a sound in real time using automation.

Reverberation, or reverb, occurs when a sound hits a surface which reflects back to a listener at varying times and amplitudes, creating a complex echo (Brooks). This effect is frequently used when mixing music. Reverb adds life to video game audio as well, as it can simulate what sounds like being in specific buildings, rooms, or areas. Adding a “reverb zone” to a specific level or area of a game sends all diegetic (in-game) audio through a reverb effect set up by a sound designer. This can save tons of memory, as re-recording dozens or more of the same individual sounds just to simulate a space could potentially use up twice as much memory. Individual sound effects are also adjusted with reverb. This effect allows sound designers to add room to things like walking in caves, water drops in sewers, or dialogue in large rooms.

There are many other digital signal processors like reverberation, or reverb, which are used to change a sound. Delay takes an audio signal and keeps it in a temporary memory buffer, and plays it back at a later time as set by a sound designer

(Raman). Using delay effects provide much more opportunity to play with sound, and gives a mixer many parameters which they can use to be creative. One possible function of delay in game audio is adding variety to sounds of monsters or other fictional characters. Some other digital signal processors include distortion, phaser, and echo. All of these tools can be used to manipulate sound, giving a sound designer the ability to create unique versions of one original sound effect.

Another common technique for mixing music and audio in any industry is automation. Automating sound changes certain aspects of it as prescribed by the mixer, in real time. Instead of lowering the overall pitch or volume of multiple footsteps to simulate a person walking away in a game, the same footstep sound effect can be automated to slowly lower the pitch or volume over time as the person walks away. Just about any parameter of sound can be automated, whether its pitch, volume, reverb, or any other digital signal processor. Programs like *Ableton* or *Wwise* make automation quick and easy as well, making it another great form of audio data management.

Anyone in music or audio production can do a lot with a single sound. Mixers are constantly coming up with creative new ways to manipulate sound, and software companies are always developing new tools that allow sound designers to focus solely on the creative aspects of production. When one sound is used for so many functions in a game, the sound designer is ensuring a large chunk of memory will be saved for other aspects of the game. Although this process restricts sound designers in many ways, it allows them to be even more creative than they originally would’ve been as they figure out how to get more variation from just a single sound.

Variations of Multiple Sounds

Using one audio file for multiple sound effects is naturally the most efficient way to save memory in game development. There are also other techniques a sound designer uses to manage memory in a creative way during the production stage. If one sound isn't quite enough to get the job done, bits and pieces of multiple sounds put together in various ways is the next best option. Compiling and layering sounds are a couple of ways to approach this idea.

Compiling Sound

Similar to cutting and fading sound effects, a sound designer will also compile different sounds in an effort to create new and/or extended sounds. In *Wwise's* 101 course, an example of a shotgun shell bouncing on the ground a few times is used to teach this technique. A designer will first cut audio files in half, separating the initial hit of a shotgun shell and the few bounces afterwards. Next, the initial hits are randomized to be played after each time a player shoots their shotgun, followed by a random selection of the few bounces afterwards. Using audio middleware like *Wwise* or *FMOD*, a designer has the ability to use randomization tools to further expand what limited audio files can do. Furthermore, pieces of different sounds can be compiled to start and end whenever a designer chooses, making it easy to use limited audio files to create lengthier sound effects.

Layering Sound

Layering sound is a beneficial aspect of data management as well. In some forms of music production, especially electronic music, artists will layer drums or other instruments to create a unique sound. Two snare drums layered on top of one another provide much more color than a single snare drum. When designing sound effects for video games, one can use a variety of sounds they have already used in the game to create new, unique sounds. This can work especially well when creating original sound for things like mythical creatures or other fictional beings and aspects of a game. Instead of adding an entirely new audio file to a game, this layering technique can prove to save storage space while also providing even more interesting textures.

Randomization in Game Environments

Using previously mentioned randomization tools in programs such as *Wwise,* sound designers can randomize virtually any sound. In war games, bomb sound effects are randomized all over a map. In games that involve a lot of dialogue, phrases can be randomized in the background to simulate a real public place. During a scene with a lot of water, such as rain or waterfalls, sounds are randomized to resemble actual water sounds in reality. By doing this, sound designers create colorful soundscapes, but use minimum assets. The alternative would be creating a long clip of audio and looping it,

which would take up a lot of space, and possibly sound repetitive. Otherwise, creating multiple clips is effective sound-wise, but could take up even more space.

Randomization in Music

Similar to randomization tools for sound effects in *Wwise,* the program also provides randomization tools specifically for music. Music is organized sound, in time. If game developers were to place a long five minute (or more) recording of a music piece, on a loop, too much space would be used. Since almost all popular, and even game music, includes repeated parts in different sections, it's common to randomize different segments of music in a game. *Wwise* allows multiple music segments to be triggered as per the sound designer's request, during a specific part of a game. In *Wwise's* 201 course, "Interactive Music," one can learn this technique by triggering any one of several different eight-measure recordings of an instrument at the end of each segment. Additionally, the designer has the option to choose when specific segments are played, so if they want the piano harmony to play every other measure, they can do that easily. This means less audio assets than a long recording, which already has the same parts repeated multiple times. Using a longer recording would be an unnecessary waste of space, especially in games where the only music is background music, and not attempting to evoke or alter a player's decisions.

These are just some examples of how to save space using multiple sounds or audio files. Not only are these practices efficient in programs like *Wwise*, but they still

allow creativity and unique approaches to be implemented. Working on sound in audio middleware gives sound designers many useful ways to reduce memory usage.

Hiding Unused Sound

Mixing and editing techniques provide the most creative opportunity to manage memory, but there are other ways that involve more technical work. Audiokinetic's *Wwise* 101 Course explains that many video games are based on different levels, and many of the sounds in those levels are unique to them. Since these sounds are only needed in specific parts of a game, they do not need to be loaded into memory, as they will use up too much. Since video games are interactive, this process is much easier than it would be with other media.

Depending on how a game is designed, sound can be hidden throughout many different parts of a game. Cutscenes are an example of sound that only needs to be loaded in at a specific time. Any time a loading screen pops up, an opportunity to hide previous sound, and bring in new sound appears. In our technologically evolving world, many games are able to do this in real time when gameplay is in an open-world, meaning no loading screens, and no gaps in between cutscenes. Audio middleware and game engines give users the ability to program all of this while in a user-friendly environment.

In open-world games, one can imagine how this technique may be difficult. Developers use technology to ensure a game runs smoothly by hiding sounds based on time of day, geographical location, and distance between any other sound that is playing, all in real time. In a game such as Rockstar's *Red Dead Redemption 2,* it is unlikely a game that large could possibly have thousands of audio assets and music samples present the entire time and still run smoothly.

Hiding sound is not quite as effective as other forms of memory management, however it is simple and doesn't affect any gameplay. Since audio files are still present in physical memory, this only allows for better processing from moment to moment during gameplay. This means space may still need to be allocated after this step.

File Conversion

File conversion is (sensibly) the last step in memory management for game audio development. This is because sound loses quality during this step, making it a less desirable technique. Although converting audio files saves tons of memory, it should not be a process that is taken lightly, and it needs to be carefully thought out beforehand. File conversion can exist in a few different forms, ranging from simple to extreme changes.

Format Conversion

One of the simplest forms of file conversion is converting lossless audio files to a compressed format, such as changing WAV files to MP3 files. This alone reduces file sizes by up to 80%, sometimes more. When converting to MP3, the quality of audio is significantly reduced, as compressing sound to this extreme removes data. There are other forms of lossless audio files such as AIFF, FLAC, and ALAC. Converting any of these to MP3 or any other lossy format would result in the same extreme data loss. Many folks do not hear any change in quality between these formats, as the noticeability is entirely dependent on the speakers being used, and even a person's hearing. Regardless, game sound is designed so players can hear the most accurate representation of what the developers originally intended, so having to compress files is sad, however, saves tremendous amounts of memory.

Other Conversion Options

In addition to completely converting an audio file to another format, sound designers will use some less extreme measures to change files in an effort to save space. Reducing the sample rate is one of these measures, which has a similar disadvantage as converting to compressed files. Sample rates determine the amount of times an original sound is sampled per second (Collins 186). Using a sample rate of 44.1 kHz is standard for CD quality audio. The highest that most digital audio

workstations allow is 96 kHz. 48 kHz is common for current music streaming platforms. Down-sampling from any higher sample rate to lower sample rate is going to reduce the quality of audio, especially higher frequencies, but it will save even more space (Marks 254).

There are a few other ways to reduce audio data used in game development. If stereo channels are unnecessary, a designer should switch those channels to mono. Also, most audio programs allow users to change the overall resolution of the audio, which also results in quality loss when reduced. All of these types of file conversion approaches will work extremely well to reduce memory in a game, but they should always be the last steps in memory management.

In Conclusion

Streaming

As technology advances, new forms of media develop, as well as new platforms for playing video games. PlayStation allows users to sign up for a service called *PlayStation Now,* which is a video game streaming service. Up until 2018, users had to stream full games to enjoy the service. Streaming all assets of a video game, such as audio, graphics, and programmable content, is a lot for an internet connection to handle in real time. Unfortunately, streaming games is difficult, especially on audio. When a

game engine looks for sound, there will usually be a delay while the audio file is located. Additionally, audio would be competing with graphics and other game aspects for bandwidth.

Google launched a product in November 2019 called *Stadia,* which is a streaming platform for video games. The *Stadia* comes with a controller and a *Chromecast* which plugs into a television or monitor, allowing players to play games without consoles, for a much lower price. There is not much information about the quality of gaming on the *Stadia* yet, however some folks on internet blogs seem to have some issues, while Google states their device supports “gorgeous high resolution gameplay.” If video game streaming becomes the norm, memory management will be affected in some way, but it's hard to tell how. One would think smaller games would still stream faster and run better than larger games, although there is little to no evidence or research at this point.

Physical Memory

Physical memory limitations very much exist in consoles and computers, limiting the amount of games a player can keep on their storage device. With the introduction of external storage devices, gamers can expand the amount of games they have installed. The disadvantage of an external drive is the simple fact that a USB connection is not as solidified as an internal drive which is directly integrated into a motherboard. Additionally, external drives are relatively expensive in addition to purchasing a console,

and are still limited in how much space they have available. Consoles do not currently come with much internal storage space, which is why streaming could have been a great workaround for gamers who like to play many games, if the quality was better. PlayStation 4 and Xbox One require games to be installed directly onto their internal hard drives, even if a user has purchased a physical compact disc. This is to provide better overall quality in gaming, but eats up the limited memory on those internal drives. With rapid advances in technology, it will only be a matter of time before the next big discovery in computer memory. When this happens, game audio developers will be able to focus more on creative aspects, and less on memory management.

New Consoles and the Future

The PlayStation 5 (PS5) and Xbox Series X, which have promised to launch in the 2020 holiday season, already announced they will have almost twice as much available memory as their predecessors currently have. Not only that, but these next generation consoles will use Solid State Drives, or SSDs, which are much quicker than hard drives, provide a better gaming experience all around. Console games have been increasing in size since the release of the PlayStation 4 and Xbox One, which tells us that game developers are taking advantage of the fact that gamers can now purchase external storage devices, and/or they are getting a headstart on how much memory they will get to work with for the new consoles. PS5 and Xbox Series X also announced they

will have an additional slot to add an extra SSD, which can allow at least an additional terabyte (TB) of memory, giving game developers a little more space to work with.

Memory management for technology in any form is always going to affect how something is created, as well as how it is used. Over time, video game sound designers have adapted these techniques in an effort to allow games to focus on graphics and programmable content such as character movements and computer-controlled bots. While limiting the sonic aspects of what a sound designer can do, managing audio memory also adds additional time to development. Film and television sound designers use similar processes for production, however, have almost no need to practice memory management as graphics, programmable content, and randomization are not present (with the exception of animation in graphics). Many of the previously mentioned techniques are unique to video game development and have been implemented by countless creative minds for the past few dozen years. As time goes by, and new technologies arise, sound designers will continue to adapt to their memory limitations while providing vast, creative worlds of sound.

Bibliography

Audiokinetic Inc, and Conservatory of Recording Arts and Technology. “Courses.” *Audiokinetic*, https://www.audiokinetic.com/learn/certification/

Axon, Samuel. “Unity at 10: For Better-or Worse-Game Development Has Never Been Easier.” *Ars Technica*, 27 Sept. 2016, https://arstechnica.com/gaming/2016/09/unity-at-10-for-better-or-worse-game-development-has-never-been-easier/.

Brooks, Daniel. “What Does a Reverb Effect Do?: The Basics.” *Reverb.com*, 2 Mar. 2018, http://www.reverb.com/news/what-does-a-reverb-effect-do-the-basics.

Cheng, William J. *Sound Play*. Oxford University Press, 2014.

Chikhani, Riad. “The History Of Gaming: An Evolving Community.” *TechCrunch*, TechCrunch, 31 Oct. 2015, techcrunch.com/2015/10/31/the-history-of-gaming-an-evolving-community/.

Collins, Karen. *Game Sound*. MIT Press, 2008.

Google. “Google Stadia - One Place for All the Ways We Play.” Google Store, Google, store.google.com/product/stadia_learn

Hall, Charlie. “PS5 Won't Require a Proprietary Solution for Storage Upgrades.” *Polygon*, Polygon, 18 Mar. 2020, www.polygon.com/2020/3/18/21185298/ps5-ssd-storage-expansion-xbox-series-x.

Ingraham, Christopher. “It's Not Just Young Men - Everyone's Playing a Lot More Video Games.” *The Washington Post*, WP Company, 11 July 2017, www.washingtonpost.com/news/wonk/wp/2017/07/11/its-not-just-young-men-everyones-playing-a-lot-more-video-games/?noredirect=on&utm_term=.8b45e248d07e.

Lovato, Nathan, and Nathan. “9 Sound Design Tips to Improve Your Game's Audio.” *GameAnalytics*, 24 Sept. 2018, https://gameanalytics.com/blog/9-sound-design-tips-to-improve-your-games-audio.html.

Marks, Aaron. *Aaron Marks' Complete Guide to Game Audio: for Composers, Sound Designers, Musicians and Game Developers*. CRC Press, 2017.

Okahara, Dwight. “Game Sound Design Tips from Dwight Okahara: Waves.” *Waves.com*, 19 Oct. 2017, https://www.waves.com/game-sound-design-tips-dwight-okahara.

Raman, Reuben. “Effects 101: Delays Explained.” *Splice*, 21 Mar. 2016, www.splice.com/blog/effects-101-delays-explained/.

Twells, John. “Game Over: 7 Synths and VSTs for Making 8-Bit Video Game Sounds.” *FACT Magazine: Transmissions from the Underground*, 15 Aug. 2017, https://www.factmag.com/2017/08/05/8-bit-video-game-synths-vsts/.

Unity. “Audio Mixing.” *Unity Learn*, https://learn.unity.com/tutorial/audio-mixing/?tab=overview#

White, Chris. “Composing Far Cry Primal: an Interview with Jason Graves.” *GodisaGeek.com*, 1 Apr. 2016, https://www.godisageek.com/2016/03/composing-far-cry-primal-an-interview-with-jason-graves/

Zabowski, Jason. “Converting WAV to MP3 Is a Hideous Thing to Do, and Here's Why.” *VOX*, Coppertino Inc., 7 June 2017, https://vox.rocks/resources/wav-to-mp3