

GAME AUDIO



Enji Maaroufi
Matrikel-Nr : 36279
Ton Seminar SS22



TABLE OF CONTENTS :

1. Brief history of Game audio.....	3
1.1. Pong (1972).....	3
1.2. Atari 2600 (1977).....	3
1.3. Space Invaders (1978).....	3
1.4. The Golden Age of the Arcade (1980s).....	3
1.5. Evolution of game mechanics.....	3
1.6. Nintendo NES (1983).....	4
1.7. Playstation (1995).....	4
2. Workflow.....	4
2.1. Middleware.....	5
2.2. Events	5
2.2.1. 2D events.....	5
2.2.2. 3D events	5
2.3. The Asset List.....	5
2.4. Things to consider before making game audio.....	5
3. Sound design.....	6
3.1. Diegetic sounds.....	6
3.2. Non Diegetic sounds.....	6
3.3. Looping	6
3.4. Sound effects	6
3.5. Where do sounds come from.....	7
3.6. Editing and Mixing.....	7
3.7. Voice overs.....	7
3.8. Music	7
3.9. Goal of game music.....	7
4. Adaptive music.....	8
4.1. Vertical re-orchestration.....	8
4.2. Horizontal re-sequencing	8
4.2.1. Crossfade.....	9
4.2.2. Phrase Branching.....	9
4.2.3. Bridge transition.....	9
4.2.4. Stingers.....	9
5. References.....	10

1. Brief history of Game Audio

Just like the first movies, the first games were silent and had no sound at all. Moving into the 1970s, we begin to hear the first primitive sounds : the *bleeps and bloops* made by the first arcade games. At this point, these sounds are created with hardware chips and the sounds are programmed using oscillators and basic tone generators. Early video game sound was limited to either sounds from PSG chips (Programmable Sound Generator) or FM synthesis chips (which were introduced in the mid 80s). The sound chips generate sound waves by synthesising multiple basic waveforms and often a noise generator.

1.1. Pong (1972)

The first arcade video game to use sound was *PONG* that was released in 1972. This was Atari's first best-seller. The integration of sound played a big role in the game's huge success. It made the experience of playing much enjoyable and attracted the crowd. Pong was a table tennis themed game and it only had 3 sound effects and no music.

1.2. Atari 2600 (1977)

In 1977, the Atari 2600 was released. It was the first video game console that integrated a sound chip. Atari designed a display and sound chip called the TIA (Television Interface Adaptor). It generates the screen display, sound effects and reads the controllers. The chip was capable of generating different types of pulse and noise output to its two channels.

The 2600 featured 2 channels of 1-bit monaural sound along with a 4-bit volume control.

1.3. Space Invaders (1978)

In 1978, Space Invaders was released. It was the first video game to use a continuous background music. It had 4 simple chromatic descending bass notes repeating in a loop, however it was dynamic and interacted with the player by increasing pace. Each time the enemies approach, the music gets faster to put pressure on the player and induce panic.

1.4. The Golden age of the arcade (1980s)

The 1980's were considered the golden age of the arcade .It was one of the major means of entertainment at that time and there was a lot of major hits and memorable soundtracks like *Pac-Man* for instance. The games music was starting to become more melodic and it was mostly created by 3 synthesised waveforms for melodies and a noise channel for simulating percussive noises. This resulted in a new style of music called *8 bit chiptune* that gained popularity in the eighties and was later used in electronic music and other genres.

1.5. Evolution of game mechanics

Early games were 2D and featured one screen at a time. The earliest games like *Space Invaders* and *Pac-Man* essentially provided exactly the same level. After a while, gamers wanted more than just one screen. One prominent early example of this expansion in arcade games was scrolling. Scrolling is a way for the game to proceed without an obvious level change. The next innovation is contrasting game levels. In these types of games, the player progresses in some way through the level, like a maze or obstacle course until they reach the goal at the end, and then the next level loads; or perhaps it's a character going from room to room searching for specific items. This provided a reason for the music to change, as our character or player is potentially in a different emotional environment than before. As far as sounds are concerned, it might affect ambiances or sound effects as animations change from level to level.

The evolution of game mechanics resulted in the birth of several game genres such as platform, maze chase, fighting, hack n slash, sports, adventure, horror.. and each genre would inquire a certain type of music and emotions.

1.6. Nintendo NES (1983)

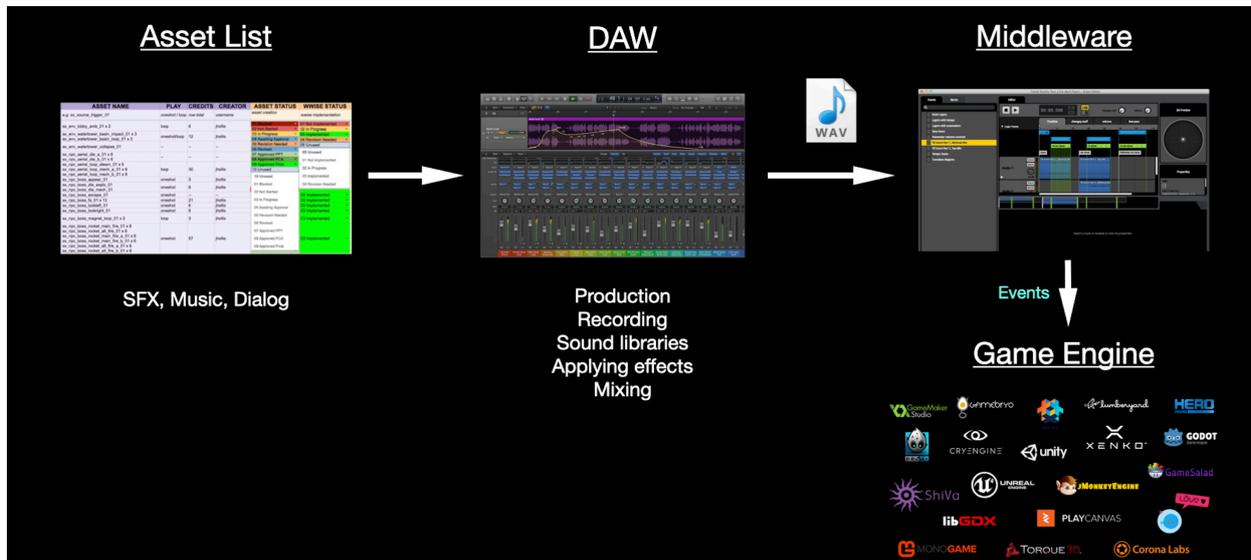
In 1983 the Nintendo NES console was released. It offered significant improvements in hardware over earlier consoles, it featured a 5 channel sound chip ; 3 musical voices (Two of them are pulse waves and one triangle wave which was usually used for bass), one noise channel for sound effects and percussion and one sample channel. Nintendo's biggest hit was Super Mario Bros, released in 1985. The music of this game was quite catchy and was written by a professional composer Koji Kondo. He provided different background music for every area Mario visited.

The NES convinced a significant percentage of arcade gamers to stay home to play games, thus presaging the eventual downfall of the arcade industry.

1.7. Playstation (1995)

The biggest leap came in mid 90s with the adoption of the CD by Sony's Play Station. The 24-channel sound chip in the Playstation provided CD-quality stereo sound plus built-in support for digital effects such as reverb and delay. The added storage, speed and memory certainly gave composers and designers some room to work and enabled them to create high quality soundtracks , add background noises for atmosphere and introduced voice acting to improve storytelling.

2. Workflow



The basic workflow in making game audio is to produce the sounds in the DAW by the sound designer and then the programmer would implement the audio files in the game engine. This approach is used in many indie games and low budget games. However, this workflow isn't preferable in big budget games that require a huge amount of audio files. To organise the implementation in a better way and to reduce the amount of code, a software is created that is called Middleware.

2.1. Middleware

The middleware is a software that acts as a bridge between the DAW and the game engine. It allows the sound designer to gain more control over how, when and where the audio is used in the game and how it interacts with the environment. The software looks similar to the DAW but it comes with extra interactive tools; so here the sound designer isn't actually crafting the sounds but rather defining audio behaviour. It comes with a lot of effects (echo, reverb, delay, compression eq ..) that can be modulated and automated based on game parameters. The software also allows the sound designer to randomise sounds and to make the music adaptive to the game. The most popular middlewares are FMOD and Wwise.

The programmer now does not implement the WAV files created by the sound designer, but the so-called audio events, in which the audio files used are stored.

2.2. Events

An event is a unit of sound content that can be triggered, controlled and stopped from game code. One can integrate more than one sound file in an event. Every situation in the game that produces a sound should have a corresponding event. It contains and is primarily composed of tracks, instruments and parameters. The parameters trigger the tracks. FMOD, for example, offers the possibility to work with 2D or 3D events

2.2.1. 2D events

A 2D sound (or event) is played without any spatialization. For example : the main player's voice; because these sounds are going to be constantly attached to the player character and the camera is going to be constantly following that character, the distance between the sound and the camera/the listener won't change. Other 2D events could also include the music, interface sounds, static ambience etc..

2.2.2. 3D events

An event is 3D if it contains anything that makes its behaviour depends on its position relative to the listener. The audio is linked to moving objects and depends on the player's distance or direction from the object. It can also use built in parameters to automate properties such as volume and EQ filters or reverb to give the impression of spatialisation.

An example of a 3D event is the sound of moving enemies or ambience related sounds like passing by a waterfall or radio or tv .. The farther the sound source, the quieter it is; the closer it gets, the louder it becomes.

2.3. The Asset List

The first step in Game audio production is preparing the asset list. It contains all the information about each sound that will occur in the game. Here, the sound designer can list what category the sound fall under, how it should sound, an indication of whether the sound is looping or a one-shot trigger. .. this implies on all the sounds (SFX, music, dialog).

2.4. Things to consider before making game audio

There are a lot of things the sound designer should consider before starting the production :

- What kind of game is this? Is it a side-scrolling kids' game with goofy cartoon graphics, or is it a big-budget console game with super life-like animations ?
- Who is the target audience for this game? Is it for adults or for children?
- What is the pace of the game? Fast or slow? Energetic or mellow? These considerations affect sound choices.
- What platform is the game being developed for?

- How big can the audio files be?
- What is the hardware space budget of the game?

3. Sound Design

Sounds in games can either be diegetic or non diegetic.

3.1. Diegetic sounds

A Diegetic sound is a sound from sources that are visible on the screen or that are implied to be present because of actions occurring in the game. Examples can include character voices and sounds from objects on the screen.

3.2. Non Diegetic sounds

Non-diegetic sound is sound from sources that are neither visible on the screen, nor present in the action. Examples are music soundtracks, SFX that are added for dramatic effect, the narrator's voice etc..

3.3. Looping

Looping is a huge part of creating sound effects or music, primarily because of the open nature of time in gameplay. It is impossible to write a perfectly timed music cue when you have no idea how long a player will stay in a specific area within the game environment. Looping also saves space in the game. The loop should be seamless and shouldn't have any clicks and pops. This applies on background ambient material, music or of foreground sound effects : example automatic weapon, meaning the weapon will continue to fire as long as the player is pushing the trigger or until the weapon runs out of ammunition. If the music to be looped is too short, it might drive the player crazy, but if it's too long, it may take up more storage space than the developer allotted.

There are several ways to create a perfect loop :

- Zero Crossing : A seamless loop can most likely be created when the end of the loop and the beginning of the loop cross the exact centre of the waveform where the volume of the wave is essentially zero. This is known as a "zero crossing" and it is vital to avoid clicks and pops.
- Crossfading
- Call and Response technique : In Music, one can use the call and response technique. It is a compositional technique that works similarly to a conversation. A "phrase" of music serves as the "call," and is "answered" by a different phrase of music. These phrases can be either vocal, instrumental, or both. We can hear this technique in the legendary soundtrack of Super Mario bros : the music starts off with a response and ends with a call which makes the loop very seamless.
- The use of the same audio material at the beginning and end makes it harder for the ear to detect the loop point.
- In ambience, to avoid a sense of repetition, one technique is to use a few ambient loops of slightly different lengths playing together, so that they overlap each other at different points.

3.4. Sound Effects

Sound effects in games include the Atmosphere, the interface sounds and the foreground sounds.

- **Background ambience** is environmental audio. It creates the setting and mood in which the action will take place. An example of diegetic background ambience might be the sound of water if the character was near a waterfall. A non-diegetic background ambience might be spooky noises in a haunted house.
- **Interface sounds** are sounds that are directly attributable to the game interface, which the player uses to view status or change settings.
- **Foreground sounds** are the individual sounds that occur when a character moves around or encounters objects within the game space. They consist of gunshots, explosions, swooshes, dashes etc..

3.5. Where do sounds come from

Sound designers may create custom libraries by recording their own sounds or they may use pre-made sound libraries. Today, the open market offers libraries of all kinds, from cartoon sounds to ambient backgrounds. Sound designers mix, edit, and recombine sounds from these libraries to produce original creations. Some sounds can be produced by recording Foley such as footsteps or using synthesis (analog or digital) to create brand new, unique never-before-heard noises.

3.6. Editing and Mixing

Once the right sounds are found or recorded, the sound designer edit and layer them in the DAW where he manipulates the sounds using effects like distortion, reverbs, EQ, reversing, pitch shifting etc.. Some sound effects would require some variation to avoid repetition. Examples would be the grunts of the player's character, the footsteps, the swooshes ; it can sound unnatural if the player hears the same sound repeating over and over again. Therefore, the sound designer makes different variations of such sound effects and later randomise them in the audio middleware. The variation can either be done by recording several different takes or by manipulating a single sound with effects.

3.7. Voice overs

The voice overs in games include :

- Dialogue : This is very common in adventure games with plots . Dialogue can also be used between offscreen commentators like in the football game *FIFA*.
- Narrative Voice over : It is often used to give instructions within a game.
- Game Feedback : It has become increasingly common for voice feedback to take the place of regular sound effects or in some cases be used in conjunction with them. Example when the player messes up, a voice would say, "Oh no, that's not right, try again!" Or "Good Job" when the player wins a fight. One common use of voice feedback found in games of all types is calling "Game Over".
- Grunts, Victory lines and taking damage sounds.

3.8. Music

Music is a very challenging and delicate part of games. Game music is non linear (it doesn't follow a timeline because one never knows how long the player would be in a location or status) so it should be able to skip around, jump sections according to cues and be loop-able. The challenge here is to make music that won't drive the player crazy by being too short and won't drive the producer crazy by being too long. It also shouldn't get too much attention when playing the game, to avoid ear fatigue. And in most cases, it should be adaptive to the game.

3.9. Goal of game music

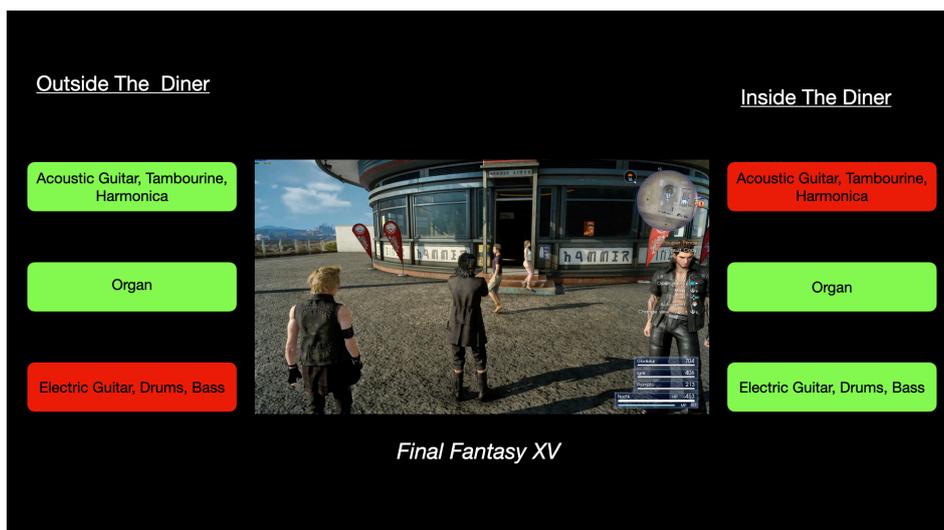
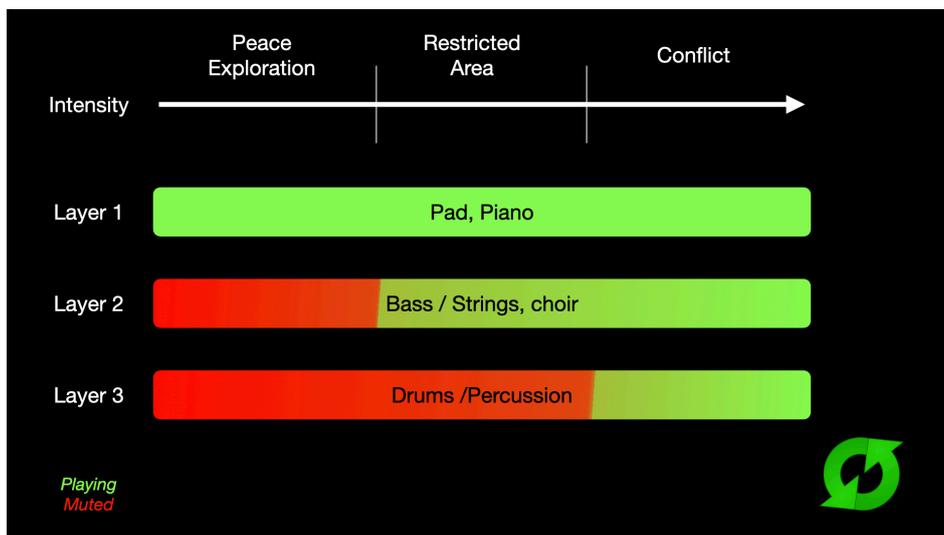
Music helps create an emotional connection to the game that makes it a satisfying experience. It sets the mood and feel of a game, the theme (action, mysterious, horror, happy..) It Identifies the time and place within the game (Medieval England or ancient Egypt..), it determines the location and setting (beach, On the streets, Desert ..) and helps players identify their surroundings. Music also establishes the pace of gameplay and helps the players to immerse in the game.

4. Adaptive Music

Adaptive music means that music changes based on context in the game. It can change when the player enters a new location or a new level or a danger zone, or when the game status changes from peace/exploration to combat mode. Adaptive music requires careful construction during composition stage. It can either be done with vertical re-orchestration technique or with horizontal re-sequencing.

4.1. Vertical re-orchestration

The idea of the vertical re-orchestration technique is to export a piece of music in several stems, which all run and loop simultaneously and can be switched on and off or have a volume automation depending on gameplay intensity or other game parameters like entering a new location. Basically, the way music changes is by adding or subtracting layers of that piece of music. The interchangeable layers usually have the same structure and melody but each layer has different instruments and sometimes different *genre*.



4.2. Horizontal re-sequencing

Horizontal re-sequencing techniques change the musical material by stopping an existing track, and starting a new one. It is composed of many sections and music will jump between the sections based on actions. So unlike the vertical technique, the individual tracks can have a completely different tempo and different key. There are different methods of transitions between the tracks.

4.2.1. Crossfade

One of the common transitions is applying a simple crossfade between the two pieces of music, which is to say, fade one cue out as the other cue fades in.

4.2.2. Phrase Branching

Another method for transitioning is phrase branching. Surely, the middleware allows the sound designer to quantise the transition after being triggered whether after one bar, two bars or half note etc.. But sometimes that won't be the end of the music phrase. So by Phrase branching, the music gets sliced into smaller segments. The beginning of each segment acts as an entry point to the musical track, and the end of each segment acts as an exit point. When triggered, the transition occurs after finishing the music phrase. For example, the middleware would wait for the final percussive phrase to end (final hit) before switching to another track. The smaller these chunks are, the more adaptive the score can be and it can quickly change to the other track.

4.2.3 Bridge Transition

Bridge Transitions are short musical cues used to connect one musical cue with another for more seamless transitions. It can be a full musical segment for a few bars or it could be a simple reversed cymbal or percussive hit for example.

4.2.4. Stingers

Stingers are also transition pieces. They consist of one-shot piece of music or sound effect that is very short and abrupt. It is a form of punctuation and applies a dramatic effect. It can either be percussive hits or vocal shouts etc.. The stinger smoothes over the transition between the slower and faster music.



5. References

Literature

Gina Zdanowicz & Spencer Bambrick (2019). *The Game Audio strategy guide : a practical course*

G.W.Childs (2006). *Creating music and sound for games*

Steve Horowitz & Scott R Looney (2014). *The essential guide to Game Audio : The theory and practice of sound for games*

Internet

<https://theses.whiterose.ac.uk/27365/1/Chapters/Chapter%203/Chapter%203.html>

https://en.wikipedia.org/wiki/Adaptive_music

https://en.wikipedia.org/wiki/PlayStation_technical_specifications

https://en.wikipedia.org/wiki/Video_game_music

<https://www.fmod.com/docs/2.02/studio/fmod-studio-concepts.html#event>

