

Development

ADAD7005_LON

Element D2b - Instrument Essay

Ashford Graham-Francis

Word count 3702

Introduction.

For this assignment I will be creating an Interactive audio Instrument using the audio middleware software

FMOD. I had never heard of this software prior to this academic module nor had any previous experience in creating such an Instrument. In truth I found this module very challenging in the beginning and it took me a few weeks to get my head into gear and in the right frame of mind to scale this technical mountain (thats how my mind perceive'd it at the time).

To start with I knew that I would need a reference point or some kind of go to manual to help guide me through this foreign Creative process and I chose to utilise the book Game Audio with FMOD and Unity (2019) created by my current tutor and mentor Mr Ciaran Robinson as my primary weapon of choice.

Before reading Mr Robinsons book (2019) and progressively working my way through the Subsequent chapters, it became quite apparent that my mentor had a vast understanding of this subject area which was a major boost to my confidence and self belief enabling me to try and grab the bull by the horns (The assignment brief).

I knew that if i was going to have any chance of building this instrument and creating something of substance that I must first understand the core principles behind the creation of computer game music and how it differentiates from Popular music and Film composition.

Game Engines.

The first principle or should I say the first concept that is spoken about in Mr Robinsons book (Game Audio with FMOD and Unity 2019 p.2) Is about game engines and what creative and technical attributes they bring to

the table. From my understanding based on the book a game engine is a creative fabricator and tool that a computer game developer can use to create a game from scratch something similar to what the digital audio work station (DAW) is to a music producer.

It is possible through coding for developers to create their own tools and apply physics to objects within the game engine. The advantage in creating your own game engine is a software that provides the developer with the opportunity to achieve optimal performance but in return will put a huge strain on the developers time and resources. However I soon learned that there is another way to create a game which is cost effective and more efficient. That way was to use a pre made engine and provided me and any other new comer who had an interest in the development of games the chance to learn the ropes whilst spending more time on creating the game visually instead of coding gravity and logistics.

It is common practice today for developers to use third party game engines like the Unreal engine created by Epic games and the Frostbite engine developed by EA DICE, but this wasn't always the case. It was during the 1990's when the third party game engine started to take centre stage throughout the computer games industry and First person shooter(FPS) games Doom and Quake were central to this transition. Due to supply and demand and the success of the games the engine developers ID decided to started licensing out their software to other games developers and branded the technology as the Doom and Quake engines.

These days a third party game engine would Normally include the following components:

- **An Audio Engine** (To provide audio control and processing for the game).
- **A Game Editor** (Which allows you to arrange and configure the game world).
- **A physics Engine** (Which provides gravity and applies physics to the game world).

If successful and you plan to release and sell your game once its finished I learned that you would still have to pay the engine developer a licence fee and then send the completed game to the engine developer to achieve optimal performance and playability before going on to negotiating revenue splits.

For this Project I opted to use the third party game engine Unity.

This engine is one of the best known third party engines alongside the Unreal game engine and is a free download from the Unity website. I found this ideal as it allowed me the time and the head space to learn the craft without occurring any financial costs or penalties. I know it sounds too good to be true that the game developers would be giving away access for free but the old cliché of “supply and demand” certainly applies here.

Once again I give the utmost credit to Mr Robinson (2019, p.2) because without his book I would have

been none the wiser and oblivious to the method behind the madness. Engine developers knew that once game developers became accustomed their software it was highly likely that would continue to use their software once they graduate into making higher budget games.

This ("try before you buy") method also opens up the possibility of the game developers becoming a consumer of assets that are accessible to the makers via the integrated shop fronts that are present inside the engines. This feature also allows consumers to buy and sells assets as they please including graphics, plugins and game templates which I opted for.

Coding and Languages.

Now this is the part Where I really struggled with Unity and due to that fact I wasn't able to deliver the interactive instrument that I had first envisaged. Never the less I will try my best to explain one of my weaknesses on the subject.

Unity speaks one language and one language only. This language is known as "Code" Mr Robinson (2019, p.3) quoted that "it is a fairly easy language to learn" but as first timer using such a format I would have to disagree but I will elaborate on this matter later on in the assignment. The Input Format used by Unity starts with the command of C sharp (C Hashtag) on the surface which is then converted to the C plus plus (C++) format in the background via the Unity processing engine for optimal performance. Mr Robinson (2019, p.3)

quotes that by Unity using both Coding formats it is able to provide consumers with “the best of both worlds”.

The Creative Process.

When I was first told about the task at hand I didn't have any clear idea of what I was going to do or where I was going to start. After doing some research On computer game engines and code languages I started to watch a few tutorial Videos made by Mr Robinson via the Tileyard and Vimeo media platforms. Ironically the video was called “Introduction to game Audio”(2023) and was the perfect reference point for me to start building a vision in my minds eye of what what could be created using FMOD and Unity together.

From the get go Mr Robinson mentions that “the way Game sound works is Via Events, triggers and Parameters” and likens this to the same way Midi works when sequencing music via a DAW (Digital audio workstation). Robinson then goes on to mention Code and Pascal casing as the correct Alphabet for spelling and writing code. An “IDE” (Integrated development Environment) is a useful application to use when writing code. It helps the user correct any grammar mistakes and autofills coding sentences. Visual studio, Monodevelop, Visual studio code and JetBrains rider are a few IDE's that are Compatible with Unity. Once again it may seem like I am harping on a lot about coding but I cannot stress enough

how important it is when working with computer game music. If I was to liken this to the human body Unity would be the Heart, FMOD would be the Lungs and the coding would be the circulation system carrying the correct information in between the two.

Computer game sound is described by Mr Robinson (2019 p.17) as been Non linear and unpredictable but with the use of Audio-middleware tools like FMOD I was able to create various events and triggers to make the audio seem more predictable and interactive.

In FMOD Events can be created in either 2D or 3D actions or 2D and 3D timelines. When re creating my instrument I wasn't quite sure how to differentiate them all from each other until Mr Robinson explained (2023, 05:10) that one is for creating a one shot sound effect (Actions) and the other (Timeline) is for creating multiple sound effects. He also explains that the 3D format is more suitable for multiplayer games which was variable I didn't need for my instrument.

When creating assets for my events it was essential to put the various samples into the correct folders using the correct grammar with Pascal casing and underscoring as these folders will be become part of the code used in Unity. Mr Robinson also states that it is very common for the code to break (not work correctly) if it is not articulated in the professional way(2023, 04:00). At first I found this new language a little bit tricky but once I started building a few events and building my banks in FMOD it started become second nature.

Learning from successes.

Whilst creating my first Interactive instrument I found the whole process quite rewarding. The concept of non diegetic music and diegetic music was very clear to see as my first instrument was made up of various footstep samples working in coordination which is known as a multi instrument. At first I had the samples on a "shuffle loop" which would play the four different footsteps in a sequence one after the other before starting back at the beginning of a cycle. This sounded quite reasonable but for some extra realism and effect I opted for the randomiser setting to play through the various footsteps at random. Mr Robinson stated in his book that "repetition destroys Immersion for a gamer" (2019 p.14) which was an ever present factor in the back of my mind. To add even more realism and depth to my Instrument I added some cloth samples and created another multi instrument to compliment and alternate in between the footstep assets giving my instrument a new and improved loop with a potential of sixteen different outcomes as opposed to the eight I had previously.

When Creating my 2nd Interactive event I was still unsure how i was going to elaborate on some of Mr Robinson's tutorials (2023) and make an instrument with my own sound and personality. In truth I had a lot of difficulties with Unity and it was starting to get the better of me like it did when I first approached this task, but after having an online personal tutorial with Mr Robinson and a few wise words of wisdom to pick me up I was able to

regain my will, focus and composure as I was advised to just play to my strengths as a creative music producer focusing more on creating in FMOD.

Following on from My personal tutorial I created my third interactive instrument and started to incorporate my own samples. I chose to build my soundscape around the Dragon Ball Z Character Cell using his signature leitmotif and vindictive sound bites. In order to achieve this I had to upload the audio samples into Logic pro X and separate them into Individual regions. Once again as mentioned before by Mr Robinson (2019 p.25) I had to name each individual region correctly with the use of underscoring and Pascal casing so all the samples would be Compatible for coding down the line. The instrument was designed to be a Game over Scene reacting the scenario if the player had lost all his health and allowing the music to adjust accordingly. Unfortunately due to my on going struggles with Unity I was unable to Implement the level of interactivity that I wanted so instead I had to settle for an eight bar loop that entailed two multi instruments and two single instrument all working in tandem.

To achieve this I had both multi instruments layered on top of each other one on a guitar loop track and the other on a bass loop track with the playback settings set to a ratio of 75% to 25%. On the second instrument I had the playback settings the same as the first multi instrument but this time utilised the Async setting. Moving on I placed the first of two single instruments in order. The first

underneath labelled as "CellGameOverFlourish" with a playback percentage of 20% similar to the example set by Mr Robinson (2019 p210). The second was placed in a four bar loop straight after the first multi instrument and had a loop region placed around it. I also added two other loop regions, one that covered the first four bars/multi instruments then another that covered both regions from start to finish.

To add some trigger parameters and effects I created another tab and called it "SceneTime". The playback range was between 0-180. Once selected I placed another sample that I had created featuring Cell and placed it at the end of the SceneTime tab. The Idea behind this Instrument was to apply automation to all three Audio tracks and then use the AHDSR modulation in conjunction with the SceneTime parameter to change the dynamics of the sound whilst the loop is playing back. Once the SceneTime parameter became full it would trigger the second four bar region and the game over quote from Cell filtering out all the other instruments to simulate that the player had died.

Learning from the challenges.

Although I did enjoy a moderate to basic level of success whilst working in FMOD I was still unsatisfied with some of my instruments and their level of interaction. For example when creating the GameOverMusic loop pictured above I would have like to have created my own theme music within Logic X to provide myself my control sonically and musically when layering the different assets into

multi instruments. I also didn't know how to re trigger the instrument back to its original position without pressing the stop button I would have liked it to be fully dependent on the scene time parameters settings.

Another challenging concept that I came across whilst working on my third instrument was FMOD's vertical hierarchy system and the importance of using Logic Tracks. In his book Mr Robinson (2019 p.211) quotes that " the vertical order of items in Logic Tracks affects how they will behave" before then going on to mention that "the higher they are placed the higher their priority". Without knowing or understanding this principle it is virtually impossible to be effective let alone creative in FMOD as the majority of its functions are based on around this principle.

Taking these new found principles on board I was a lot more confident and assured when creating my fourth interactive instrument. The idea behind it was to create a loop consisting of three different single instruments that would complement and co operate with each other whilst the loop was playing. The idea was similar to the third instrument but was equipped with a lot more interactive triggers and event parameters.

The scenario was initially designed to react to players picking up items whilst playing the game, but in my case I struggled with coding and setting up the standard assets in Unity. To improvise and avoid this setback I created this Instrument with the idea of it being a proximity radar and hazard warning, to prepare the gamer for an eventual

encounter with Cell.

Utilising the game score parameter tab as a trigger I was able to create a range of motion that was full of interaction and variety. The introduction of tempo markers, transition regions and transition timelines were important mechanics throughout this process. Their unique features and capabilities allowed the playback cursor to jump between the three different single instruments depending on their level of success on the game score parameter. The higher the game score the closer you were to facing Cell almost like two polarity poles on a collision course.

Once the desired encounter would happen the instrument would skip to the third single instrument and trigger a vocal response from Cell quoting "This time there is no one to interfere". To acknowledge and reacted to the imminent danger the tempo marker above was set to 120bpm 20bpm more than the other two instruments to let the gamer know they are about to do battle.

The transition timeline was a very useful mechanic as it allowed me to cover the second instrument region with multiple transition markers and measure their distance in music theory notation. Once I discovered this and learned how to use destination markers I felt even more comfortable working with FMOD and started to experiment with nested instruments (2019 p.225).

Visually I found FMOD much easier to understand and

learn compared to Unity with its interface reminding of Ableton Live. However with that being said its heavy reliance on the looping of audio regions, probability settings and tempo marking make it more than something that is made by "Fisher Price" a quote often used by Mr Robinson(2023).

In order to improve and develop with FMOD I feel a basic understanding and further studying of its core principles and fundamentals are essential to improving my creativity, workflow and productivity. Trial and error matched with patience and perseverance will be at the heart of my on going progress.

The Journey

Whilst working on this project I found it very challenging especially from an emotional and mental point of view. In the beginning it felt like the task was "bigger than me" and slightly beyond my capabilities as I had never attempted to create such an instrument before. It almost felt like I was a freshman starting out at high school again with no friends and having to establish a name for myself. There was a lot of new information to take on board and process, which was not helped by the fact I work 48 hours a week and still had some outstanding work to finish from the previous semester.

As you can imagine the day to day work commitments and on going life problems were making it difficult for me to focus and study. My work life balance suddenly shifted completely onto work and was leaving me emotionally drained and devoid of

creativity. When It came to attending my online lectures I was either half asleep or catching up on sleep from the previous night shift and started to feel like things were getting out of my control. At one point I was working a shift pattern of 6 days on and 1 night off which totalled at over 62 hours per week. The craziest thing about it was I did not feel like I was making any progress or moving forwards despite working all of these hours and soon realised it was down to my mind avoiding the task at hand.

Fortunately for me I have been in this situation before and failed. Buckling under the pressure and opting for the easy way out something that I promised myself I would not do again.

Due to my past experience and improved level of maturity, I decided to start attending my online lectures again with one of my course tutors. There were no other students in the lecture so It turned into a personal tutorial. She reminded me about the importance of completing this course and suggested that I put everything else on the back burner. She also gave me a small step by step plan of how I should tackle the various modules and tasks that remained. This strategy reminded me of the traffic light model I had downloaded from the bebop app (Tileyard education hub) at the beginning of this course.

Once implemented I began to grow in confidence and started ticking the boxes off one thing at a time. I found this method very refreshing and intrinsically rewarding as I was no longer stuck in my head or avoiding "the elephant in the room."

Looking forward

Looking back on the whole process from start to finish I feel have grown considerably throughout this module not only as an artist but more as an aspiring industry professional. I have developed an inner skillset and frame of mind that will hold its self accountable when things are not going according to plan. On top of that I am no longer afraid to ask my tutors or support system for help when I am falling behind or no longer in control of my situation. I have also learned that it is important to isolate yourself from the crowd when it comes to focusing on deadlines and handing in assignments. Friends can sometimes become your worst enemy by way of unwanted distractions or friendly conversations. This can become a hazard to your success especially when you are in the middle of learning something new or struggling to find a productive mindset.

For future projects I looking forward to seeing how far my potential can take me by attacking the tasks as soon as possible instead of burying my head in the sand and responding when it is nearly too late.

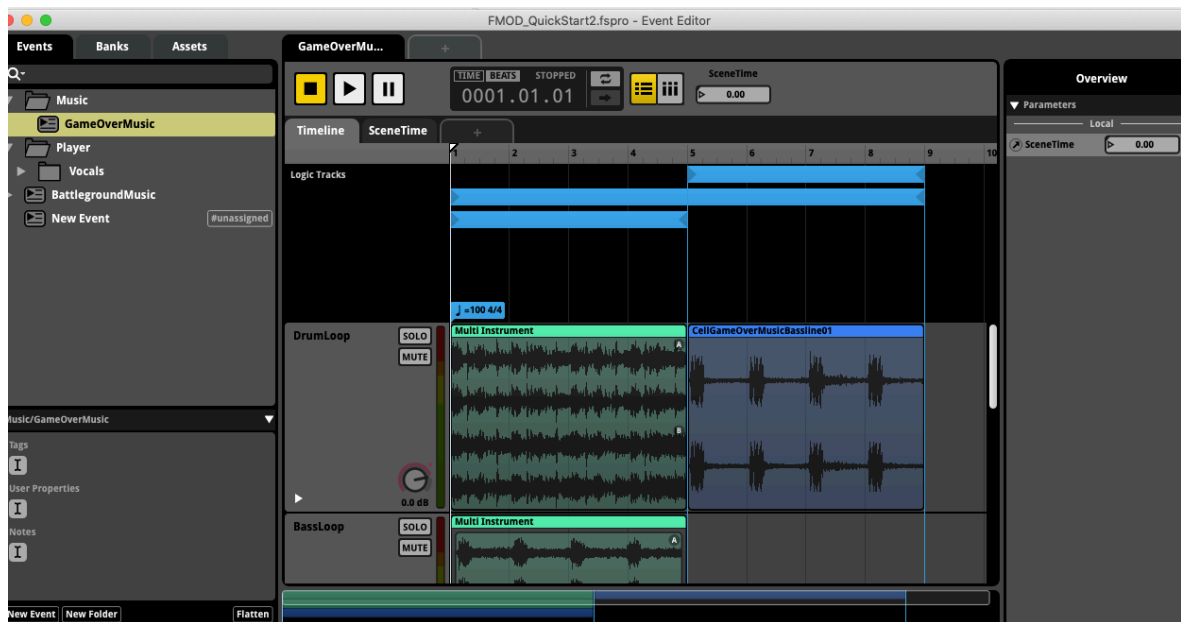
Before attempting this module I did not consider game audio as a career or a lucrative path I could go down. But moving forward and looking towards the future it is definitely something I could see myself doing as I have been a massive gamer through out my childhood years.

CONCLUSION

To conclude I am not totally satisfied with the interactive Instruments I have created. I wanted to create my own Instrumental and record some of my own audio samples then link them up to Unity with an IDE (2019 p6). In truth I really struggled with Unity and its interface drove me up the wall at times. That coupled with my work commitments and concentration issues made me feel like I was not up to the task. Thankfully with the help of my various Tutors and family support system I was able to put something together and complete to the job at hand.

This module has shown me that with the right focus and attitude anything is possible and not to “throw your toys out the pram” when things are not going your way. It is important to manage your time accordingly and plan for success otherwise you are planning to fail.

Appendix



Event Instrument Three.



Event Instrument Four.

Bibliography

Robinson. C (2019). *Game Audio with FMOD and Unity*: New York, Routledge [Online] Available from https://tileyard.beebop.app/resources/modules/content?class_group_id=1954&content_type=class_view

Robinson. C (2023) *Game Audio Introduction*: Tileyard Education [Online video]. Available from https://tileyard.beebop.app/resources/modules/content?class_group_id=1954&content_type=class_view [Accessed July 1st 2023].

Robinson.C (2023) *FMOD and Unity Integration walkthrough*: TileYard Education [Online] Available from https://tileyard.beebop.app/resources/modules/content?class_group_id=1954&content_type=class_view [Accessed July 1st 2023]

