Reflection Form

Since this is my first year going into coding with Unity, and C#, I personally am still developing my skills alongside of doing work or even my personal projects which further enhance my knowledge of Unity and its complex components additionally, the C# functions and what to manipulate within a script to change a Unity component. However, being tasked to work on three briefs has further bolstered my insight in Unity and C# and for a challenge, I decided to choose one beginner, intermediate and advanced brief to explore the range of difficulties within Unity and in terms of programming, whilst also increasing my knowledge on the topic itself and to push the boundaries of what I can and cannot do to gradually gain confidence in myself when coding future projects – not only will this be helpful in upcoming projects, since I have the habit of commenting the scripts, I can look back at them for references in newer scripts making programming more easier.

The first brief I tackled was the ‘Rolling Road’ beginners brief, which was to create an endless runner. Already having experience, I remembered to create platforms which had to be cloned for the player, however the way I would do them in my previous own projects was to clone an already existing platform that will be duplicated below the player to create those platforms – yet the duplication will also occur above the player which is a very inefficient method of instantiating objects hence why I created a newer script called ‘NewPlatformSpawner’ which outperforms the old script in every way possible as it introduces more control over the spawning platforms as well as the possibilities of turns in the game during runtime. And naturally, the platforms are destroyed once out of camera view to keep the performance efficient for the player as well as meeting my brief. This brief was able to find gaps in my knowledge such as using the random value which was completely new to me and after some research, I was able to use it comfortably during this brief to create those turns you would see in games such as temple run. An insight on what I should improve on during this brief is to create better naming conventions which is what ill be working on in upcoming plans to keep the code clear and concise for readability and reusability.

For the Second brief, I decided to follow the ‘Shuffle’ brief which meant I had to create the Shuffle feature for a playlist so that random songs are picked when a track is finished playing or skipped. Having a crack at this brief, I firstly had to brush up on my for loop implementation and devise a way to iterate a list which will check whether the element in the list is checked for whether it had been played or is currently playing and to separate them from the tracks that haven’t played yet in order to differentiate between what’s played, playing and hasn’t played. With better naming conventions, I was able to understand what is happening within the script as the tracks had to be split into many different variables to split them up from what’s playing and what isn’t, however, to catch errors within the script during runtime, I created an external script called ‘ShuffleTester’ which checks for duplicates in the audio tracks played by the ‘ShufflePlaylist’ script. Using an event handler, I was able to invoke every time the ‘ShufflePlaylist’ script changes the currently playing track. With that in mind, I created two scenes whereby one will play the songs within the playlist without errors and the other scene with errors using a simple Boolean to stress assess my shuffle function and to find errors when it shuffles the tracks. I can gladly say that this project has led me into using events for the first time which broadened my knowledge of Unity’s call-back function which I can use in the future, additionally being the best brief I personally have finished because of the complexity, readability and reusability of the code in the future.

Lastly, my third brief was the most complex one yet, as I had not ever explored spectral data and the calculation side of Unity, however challenging myself, I wanted to give this a go and see the results. Grasping the knowledge of how spectral data is split and divided into smaller segments was hard to understand but besides the difficulty – I was able to calculate the Hertz for each band the game objects should manipulate, additionally, using ‘FFTWindow’ to reduce leakage of spectral data had given me insight on what types to use when handling audio data. Although this was mostly a learning curve for me, I can definitely reuse this code in the future for games that may require sound to be displayed and not only did I learn along the way whilst creating this brief -but I have also picked up a better understanding on maths in Unity and the many math properties Unity has. I can conclude that this brief was the most challenging - yet the most fun to create since the results of the project worked accordingly and is aesthetically pleasing when ran whilst also being the most helpful in terms of maths as it had expanded my insight in Unity maths.

Finally, what is to be concluded?

Doing the three briefs has given me the opportunity to gain experience in the sector I want to flourish in, in addition to introducing me to new concepts and giving me a better understanding on Unity and its components with the help of tutorials and forums, my programming knowledge is much more enhanced from when I didn’t do the briefs - yet after doing them, I can see the progress I’m making in my programming journey. Of course, there will be difficulties during this period, but the more practise I do, the more familiarity and progress will increase as well as filling in those gaps left in my knowledge. This was a fun experience and is certainly something I would do again!