Reflective Statement– Game Specialism

Before I started picking the three briefs I would do, I looked through them all and choose what I thought would be in my capability as I’ve never programmed before. I chose programming specialism as I wanted to try something new. I kind of thought we would be learning a bit about programming before diving right in to making them, so that made me nervous as I did not have the experience needed to make a good or even functional game.

My first brief I chose was Speedometer as I thought I would be able to tackle this with little to no problems, but when I researched other people making speedometer online, I learnt how much maths would be needed to complete this task or even to make it look like it would work. My plan for this brief was to have a car just go across the screen when pressing a key, and having the speed it would be going at show you. To start with I made a square game object which will represent my car, and this being the easiest part of the game in all. I had to work out what meters per second to miles per hour would transform into, which is 1 meter per second would equal 2.23693629 miles per hour but having the decimal points afterwards would not be helpful to my game, to say I started coding in C# only whilst being at university my knowledge is not that good, so I asked Max who knows more C# then I, with his help I got the code:

float milesPerHour = speedMetersPerSec \* 2.23693629f;

text.text = milesPerHour.ToString("N0") + " MPH";

Diagram

Description automatically generated with medium confidencewhich means that on the text part of the game i.e., the speedometer bar, will show what speed the car will be traveling at. This part gave me loads of decimal points numbers so what I had to do to get rid of this was add “N0” meaning that I wanted no decimal points. And I fully didn’t understand the code.

So, I looked it up what it fully means, and I still didn’t fully understand until I got to the second brief. Looking at this image it helped me understand it better.

This was the hardest part of the whole process. The feedback I got about this brief was that the car was too big to look like it was going 120mph, so it went off the screen faster. With that I changed the size of the car, but what I wanted to add/change was how it looked because I just gave it a basic shape, and now looking back I wish that I added some art assets into the brief as well. But I was just mainly focusing on the coding side. For the extra credit it asked, “Include a display component which converts the speed into a rotary dial”, I tried this but as it was my first-time doing programming I just stuck to the basic slider. So, in future I would attempt this. Overall, it shows the basics and someone with little to no C# skills can do.

For my second brief I chose the FPS counter as I thought this would come handy in other games I attempt to make. To start with I watched videos of others making an FPS counter in unity as I had no idea how to start this. (1) the video by Code Monkey and how easy it is to make FPS counter. Looking at this it gave me no help as I still didn’t know how to program fully yet. I started off by adding a UI text box, which will show the Frames Per Second. I was going for a simple text as I thought it would be the easiest for someone with my skill. As I was making this, I was trying to understand code, but it was hard to fully understand. As I was coding it, I got to a part where I had involved but then realised that I can change it and have a simpler way of doing it.

if (lastTime + 1 <= Time.time)

{

lastTime = Time.time;

textComponent.text = "fps: " + fpsNumber.ToString("N0");

fpsNumber = 0;

else

{

//fps = 1 / Time.deltaTime;

fpsNumber++;

}

When I got it working the feedback I got was if I can add it too a graph, with that I tried but I made it wrong and did something to my first code of this brief, so I just went back to how it first was. I thought it be better to show that this works then have a graph that messed it up. But I will attempt to that in the future as I want to improve my unity skills. Overall, I think this brief was the easiest as it didn’t need that much code to make it work, and it didn’t have art assets.

For my third brief I chose the Radar as this would come in handy when I want to make more games in the future. I was going to do a pulse radar (2) as I thought that would be easier than a sweep radar, but after reading the brief it doesn’t say anything about it being a pulse or sweep, so I went with it being a constant radar. I started with the player being able to move and have two enemies that would show up on radar map. I struggled putting in the made radar UI as I never done that before so I asked Max for help, and he showed me where I could find it and helped me put it into my game. I had it where the enemies would show up on the radar and they would disappear after the player was no close to them, but I got feedback saying that if I have the enemies still on the radar map as the player walks away it would improve it, so I changed it. I like how with this brief you can see the progress made and what I’ve learnt over this semester. What I would change about it is make it look more better, the art asset is just basic. Overall, I think it turned out the best one out of all the briefs.

Overall, I feel that I’ve improved as the briefs went along and next time, I would try a challenging one just to see if I can handle it.

1. www.youtube.com. (n.d.). Simple Framerate FPS Counter in 30 SECONDS! [online] Available at: <https://www.youtube.com/watch?v=I2r97r9h074> [Accessed 02 Mar. 2023].
2. www.youtube.com. (n.d.). *Awesome Radar Effect in Unity!* [online] Available at: <https://www.youtube.com/watch?v=J0gmrgpx6gk> [Accessed 20 Apr. 2023].