Game Specialism: Programming

Mirco Brief 1: In Game FPS Counter A screen shot of a black and blue screen

Description automatically generated

A screenshot of a computer program

Description automatically generatedFor the first brief I wanted to select something simple to get me started as well as something useful that I could use import into any game I am developing. I chose the FPS counter as it is a useful tool to use when developing your game, it can help you identify how smooth your game is running and where they might be performance issues in your game.

Testing:

To test the FPS counter, I decided to use something that would dramatically reduce the fps of the scene to see if it could keep up with the quick reduction in frames. To do this I generated a significant number of particles using unity’s build in particle generator, this produced the intended results and reduced fps from average 140 fps all the way down to as low as 55 fps which the counter successfully displayed. By reducing the FPS so dramatically it also helped when testing the average FPS as it saw a decrease which would slowly rise again after.

How could I improve?

If I were to redo this brief, I would want to include multiple new statistics to go along with the standard fps count, this could include the ability to show the 1% lows to help show when a game is struggling in a particular scenario. Other more advanced FPS counters also do more than just show FPS, GPU and CPU utilization could also be something to looking into when redoing this brief.

Mirco Brief 2: Speedometer

A white and blue sky

Description automatically generated

For my second brief I decided to do a speedometer, as I wanted to try designing a character controller in 3D for the first time as well as have a template if I ever wanted to design a racing game.

Method: As this was my first time creating a character controller for a 3d game, and first time working in 3d in general, I didn’t have any previous experience and scripts to work on, so it took a bit of time to get the character moving, this was also the first time I made a character from the first person so I was also learning how to move the camera with the player. To have a speedometer that displayed miles per hour, I had to take the velocity from the Rigibody and times it by the conversion from meters per second to miles per hour.

How would I improve:

Next time I would attempt this brief, I would like to add a digital display to the speedometer, this would bring it closer to being a functional asset to use in a game. I would also like to attempt to design a character controller that feels more like a car and to design a track to properly test it on.

A screen shot of a computer program

Description automatically generated

A road with a white object on it

Description automatically generatedMirco Brief 3: Rolling Road

For my last brief I wanted to try making an infinitely rolling road, a massive portion of the mobile games industry are games that feature an infinitely and procedurally generated rolling road, this would help me gain some experience as to how to basics of those games are made and would help me if I ever decided to make a mobile game myself.

As this was only meant to be a test for the rolling road feature itself, I decided to keep everything else basic and focus only on the mechanic, this was not meant to be a game only a test for to get the road to work properly.

Method:

I decided to stick to one prefab for the test to get the feature working first before adding to it if I had to time. The road worked by spawning a set number of road tiles at the start of the game, the road itself would then move backwards as a set speed until in left the players view, when the game detects that the road is behind the player and out of view, it repositions the road to the back of the group, it does this for every tile infinitely.

Testing:

When testing the rolling road, I made sure to change the values of the number of roads and speed of the road to ensure that everything ran smoothly which it did just fine even going up to 10,000 road titles before encountering performance issues.

A screenshot of a computer program

Description automatically generated

How would I improve:

When doing this brief again, as I have a better understanding of how the core mechanic works, I would have more time to dedicate to adding more features, I would like to add different prefabs and have it chosen randomly from each one, adding changes in direction would also help to create a more fleshed out feature. To do these I would need to change how to infinite road works now, instead of placing the road that went out of view to back of the tiles, I would instead have to destroy that tile and generate a new one.