***Reflective statement on the programming development process***

So, this reflective statement will provide insight into the development process undertaken for the three programming briefs assigned as part of my coursework. However, I attempted to do 4 with three of the projects succeeding. It brings in the creation of a speedometer (1 succeeded), Radar project (failed attempt), implementation of an instanced scrolling material(succeeded), and development of a Stay system name generator(succeeded). This document aims to journal my experiences, challenges, and learnings I encountered throughout the development journey for each brief. As I reflect on the iterative process and problem-solving strategies employed it should convey programming skills I acquired during the completion of these tasks.

Firstly, the speedometer tasked me with creating a functional speedometer within unity to display the velocity of a moving object. Initially I approached this task with a focus on understanding the requirements and determining the most efficient and effective way to achieve the desired result. Upon reviewing the specifications, I realized I needed to track the velocity of an object and display it in MPH on a console. To accomplish this, I researched various unity components and scripting techniques, considering factors such as performance. One of the primary challenges I encountered during development was accurately capturing the velocity of the object and translating it into MPH from the standard unity scales. Though experimentation and iteration I used the “Private Const float MPS\_TO\_MPH=2.23694f” conversion factor and used the update speed method to finish off the conversion and print off the speed results. Throughout the development process I maintained a proactive approach to problem solving leveraging online resources and unity documentation. I used a slider to make the speedometer more interesting in the sense of you can turn on the slider and then the speedometer will work as intended.

Next, the instanced scrolling material brief was the easiest out of all. It involved dynamically creating and manipulating materials to achieve the desired visual effect of scrolling textures across the game object. At the outset I conducted research into unity instancing capabilities online and the unity documentation and explored various approaches to efficiently scroll textures across the game object. Through experimentation and prototyping I devised a solution that leveraged unity material class and shoder properties to dynamically adjust texture offsets in real time.

The implementation was focused on the script called “Scroll texture” which was attached to the Game Object containing the image component which is water. One of the challenges was when I first tried to play the game, it had a strange occurrence with the sprite and it glitched and stretched. This is because sprites share the same material instance among themselves. To fix this I had to reset the texture to the original state I set the main to vector2.zero. I also made a clone of the original material in the start method and then I assigned it to the object. Also, the scrolling speed was very high and it was stretching so I change the speed to a lower speed. As well as I changed the wrap mode on the text from clamp to repeat. All these things fixed the issue that I was having on the scrolling texture. Overall, the instanced scrolling material brief provided good insights into shader programming, material instantiation and performance optimization with the unity engine.

Finally, I decided to challenge myself big time with the name generator brief tasked me with developing a dynamic system for generating pronounceable stay system names within the unity engine. This included implementing algorithms to come randomly selected adjectives and nouns while also ensuring the generated names were free from duplicates and obscenities. This was the one of the hardest programs I have ever written. To tackle this challenge, I began by researching methods for generating pronounceable words and exploring techniques for text manipulation within unity. I needed to learn new libraries for example TMPRO through online resources to enable me to do this. After gaining knowledge of the library and requirements I devised a solution that utilized text assets containing adjectives and nouns which were randomly selected and connected to form different star system names. I had to carefully use the box. Local position transforms with the correct values of where my box was in the game scene like 0, -475f + (value \*100,)) otherwise it will not work, what this would do is to make sure the boxes are formatted one on top of each other this is because the boxes are 100 in height and if \* this by the value each one will be on top of each other. Moreover, to address the logic for checking duplicates and obscenities I integrated external scripts for duplicate checking and obscenity filtering into the name generation process. Throughout the iterative development cycle, I continued to refine the name generation algorithm, fine tuning parameters and optimizing performance to deliver a seamless user a seamless user experience. Additionally, I conduced extensive testing and validation to verify the effectiveness of the duplicate and obscenity checks ensuring the generated names met the criteria. Overall, the name generator brief provided a unique opportunity to apply algorithmic thinking and data validation techniques in a practical setting. Specifically, the name generator brief I gained valuable insights into the importance of through testing and validation in software development.

In conclusion the completion of the three-programming brief Speedometer, instanced scrolling Material and Name generator has been an interesting experience. Through these projects I have learn and gained insight into problem solving and algorithmic thinking. While also gaining a deeper understanding of C# and the unity engine library.