Programming Brief Response

For specialism, my goal was to create the four briefs with no bugs to a complete standard. I believe I achieved this very well, and in some briefs, I made them look presentable to a user too.

I found the briefs this year notably easier than the ones from last year. This was due to me attempting the majority of the challenging briefs last years, and the briefs this year had not changed. Of course, I would not do the same brief twice, and so had to do some simpler briefs instead. Despite of this, I still learnt lots from the specialism course this year.

Firstly, I attempted the speedometer brief. This brief asks to create a script that tracks the speed of a moving object. On the surface this is rather simple, and there is an extremely simple method of calculating this using rigidbody.velocity. I did not use this method as it did not seem like I was creating my own code, rather using an in-built unity method. Instead, I took the Vector3 position of the object and calculated how much the object moved since the last frame, repeated this until 1 second had passed, and then calculating MPH and KMPH from this.

The brief I made shows an automatically moving ball with the MPH and KMPH of the ball calculated and displayed on the UI. The camera also follows the ball so that it is always in vision. I made the ball move automatically as it would show that there is no other interference between the player input and the velocity of the ball.

I chose a simple brief as the first to give me motivation for later harder briefs, as I would have already completed 1 to look back on. I enjoyed this brief, and it showed me the importance of using Invoke rather than a custom-made timer when the function must always be run at a set time.

Secondly, I picked a more challenging brief and attempted the extension task for it. This brief was to make a coin calculating system to compare different monetary values on the old British coinage system. This brief was very challenging, not just for coding, but also understanding the old money system itself. I learnt to use a complex system of arrays to hold the data for the old money system, and then use a combination of for loops to search through this to find the coin combination required. The way I did this is by converting the money to pennies, the shared lowest value of all the old monetary system (with exception to half penny an quarter penny). Then I can calculate how to convert the pennies into the correct coin combination through these loops.

Finding which coin combination was worth more was far easier using this method too. This is since I already know the penny value of each combo, I can simply just see which number is bigger than the other. Overall, this was the hardest brief to do, and it is not fully functional. It can calculate all coin combos which require 2 or less coins, however, going over this causes an error. Despite this the code is acceptable in its current state, it is just a shame that it is not fully finished.

This brief challenged me to be creative in my solution, as the old British currency system works nothing like the current system. There is also not much support online for this task and I had to invent my own solutions.

Thirdly, I attempted a brief which was to create an auto-scaling text box. This means the box grows automatically when text is put into it. This is to prevent text getting cut off. The way I did this is by calculating the estimated size of the letters being written, then increasing the x size of the text box by n letters typed * size of each letter. As well as this, I made it so after a certain number of letters the box would then grow downwards in the y direction. This is because monitors do not continue forever horizontally, and many more words can be shown if the line is dropped down.

This brief taught me about sizeDelta which canvas objects use to determine their width and height. This has already been helpful in other programming tasks I have done and has helped me understand how to create a more dynamic UI system.

I can see this brief being reused in other projects that I may work on, especially for an RPG or animal crossing style game where NPCs will talk to the player using text. This brief can be hopefully adapted to support world-space canvas to allow for speech bubbles in the scene to dynamically grow and shrink to support character speech.

Finally, I attempted the fourth brief which was an FPS counter. Originally, this was rather easy to make, and I had the base FPS shown in the console on the first day. I had to lookup how to approach this brief however, and I learnt that people use the Time.deltaTime variable to determine the FPS. This was new to me, however the math for calculating the FPS is rather simple and so did not take up too much time.

After awhile I returned to the brief and attempted to create a GPU graph to display the FPS. I knew I needed to record the FPS in a list to show all the FPS that had past, this bit was simple. However, the graph was much harder to do. I took heavy inspiration from a YouTube video that showed me how to make the graph using custom meshes.

I had to use this convoluted method as I wanted the graph to show on the UI rather than in the scene, and annoyingly the Line Renderer does not work on a canvas. Adding the graph massively improves my brief response, and makes it look much more complete. The graph also automatically scales its top value depending on the max FPS that has occurred.

I am not fully confident with the method I used to complete this graph, however, due to the fact it was an extension task, I was less concerned about taking heavy inspiration from someone else's code. I usually am very hesitant before copying someone's code and adapting it to my program, instead I like to program everything from scratch myself. Perhaps this is not the way to go about programming, and I have learnt that I should use pre-existing code and modules to support me.

This last brief also taught me more about Lists and using a list like a queue to add and remove values from the start and end of the list when they need to be.

I found time organisation somewhat challenging this semester, especially due to the VLE being down. I rushed out my briefs quickly, thinking that they were going to be due very soon. However, the deadline was a long way away. I do not regret doing this as it did allow me to focus more on my other work and allowed me to especially work on my recreation game project. However, it would have been less stressful to spend less time at one go on the work. I am lucky that I did not burnout, perhaps because I did enjoy these briefs that I was working on, and the course in general, that this did not happen.