Reflective Statement

What have I achieved?

Developing my Rolling Road brief, I have produced a system that generates an infinite road with turns, rolling towards the camera giving off the sense of movement, including adjustable attributes to customize the road and control its behaviour, keeping track of my progress in a production diary which allows monitoring my progress and reflect on previous work.

I started by setting up a 3D Unity project, piecing together road section prefabs, a straight road, and two turn sections for later advancement. using 3D game objects and empty game objects as anchor points to connect the roads, a runner character with a collider and Rigidbody set to kinematic to keep it still, tilting the camera down towards the runner, and placing an empty game object at the centre base of the runner to attach the main rolling road script and use as a pivot point for the roads path. Next, I coded a script for handling the rolling road system; starting off simple by generating a straight road, with properties to manage how many roads to instantiate at the start, and how far back they should start spawning to prevent the road's starting edge appearing in camera view; right beneath the runner. Following on from this, I added triggers to the road prefabs and created scripts to detect when the runner has left the section, destroying itself after a tweakable delay and spawning a new section to the end of the road, following by adding movement with alterable speed properties to the rolling road system.

For extra credits I incorporated functionality to generate turn sections to the rolling road system. This required producing further scripts; accessories to be added to the road prefabs and additional behaviour in the rolling road system. To achieve this, I started by keeping the road prefabs in an array; using their index to randomise which would spawn next, along with a variable to manage their new spawn rotation after a turn, integrating a game object with a trigger attached into turn area, attaching a script to administer when to rotate the generated road path using the trigger, before I then inserted another empty game object a little past the end of the road turns; linking a script responsible for detecting obstructions in various paths to find a possible path for the road to follow, this prevents the road path overlapping itself when spawning turn sections. Furthermore, I embedded attributes to control the frequency of which turn sections spawn to increase or decrease road path 'jaggedness'; this could be used to govern difficulty if the system were to be used in a game. Penultimately, I upgraded the road spawn behaviour to generate new roads if the end of the path was visible to the camera, instead of only appending a new one every time the runner leaves a section, keeping the road short and the path ending visible to the camera view, also incorporating variables in the inspector to restrict how many road prefabs can be generated into the path at a time to aid performance.

Finally, I wrote a visually appealing document with images to explain how to use and set up the rolling road system, what properties can be used to manage its behaviours, and how to assemble custom road section prefabs to be used by the system. The nature of this document is that it can be used by a 3rd party, as a concise walk through if this was to be used in another, separate project.

What have I learnt?

Throughout the process of building my rolling road brief I have gained valuable experience in Unity using C# to procedurally generate a rolling road path. Diving deeper I have grasped a better understanding on how to properly set up a prefab in a specific fashion; making use of empty game objects as anchor points to connect additional prefabs, function of the instantiate and various transform methods; for instance, needing to update the rotation of newly instantiated roads after a direction change or translating objects relative to the world axis to keep them heading in the same direction after they have been rotated. Discovered making use of loops to go through child game objects to synchronize their behaviours, for example their movement and speed, and unearthed the 'OnBecameVisible()' and 'OnBecameInvisible()' methods to detect when a mesh is in camera view.

What would I aim to do differently next time?

Reproducing the rolling road brief, I would aim to make the system more customizable and universal, and less limited, starting off with adding capabilities to use more than 3 road section prefabs providing room for variation in the generated path. Enhanced customization with path obstruction detection; embedding the ability to tweak directions it should scan; currently limited to only test left, right and ahead of the path end, thus unrestricting the turn angle of the turn prefabs. Further control over where and when the path starts, stopping the path movement, extended choice of how to generate the path; spawning turns by calling a function or after a delay, instead of just doing so randomly. Improving turn behaviour to keep the runner in the same horizontal position it was at previously before the turn, incorporating smooth turning and regulating its actions with the move speed, extra preferences to handle when to destroy a road section, with options to dispose of it after the runner has left the section and is out of the camera view, or and with a delay. And finally next time I would incorporate input controls to manipulate the runner's movement and build a mini game using the system to demonstrate its capabilities.

How has what I have created assisted me through my studies?

To conclude, the experience I have absorbed assembling my rolling road system will help me better plan and develop future Unity projects used in my studies, particularly those involving 3D scenes, managing; and synchronously manipulating masses of objects, and/or procedural generation.