MV4025 Lab 1

The goal of this lab is to gain experience implementing A* pathfinding in a 3D simulation using a third party A* implementation. We will implement in Unity using the **free version** of the Astar Pathfinding Project, available at https://arongranberg.com/astar/.

1. Determine who will be in your lab team (up to four total members). One of you should email me a list of team members by COB today.

2. The free version of the Astar Pathfinding Project package has been downloaded and imported into the Unity project provided for Lab 1 (available on Sakai). Verify that this is the case.

3. Take a quick look through all of the documentation available online for the Astar Pathfinding Project, and determine what looks most useful. The "Getting Started" and "Movement Script" sections deserve special attention. Do not use the AIPath, RichAI, or AILerp components. These are prepackaged higher-level behaviors, and we want to build our own that we can totally control. You will use the Seeker component, however.

4. Familiarize yourself with applying the Astar code. You may want to create a simple secondary scene in the Unity project for testing purposes.

5. Once you understand how to use the Astar code, generate a square grid of nodes spaced 5m apart over the entire terrain. **Treat the wall as an obstacle, i.e. no edges of the navigation graph should go through the wall.**

6. Using the Astar code and modifying CommandInterface.cs and Entity.cs as necessary, change the behavior so that when the user orders a blue entity to a position, instead of moving directly there in a straight line, an astar path is generated to the nearest graph node (waypoint) to that position, and the entity follows this path to that node. (Hint 1: Set the "goal" of the entity to be each successive point in the computed path. Hint 2: Make your own copy of the path to ensure that the pathfinding library does not change it when you still need it.)

If this is not achievable in the time provided for the lab, then try to implement one of the simpler functionalities described below:

a. Just generate a line indicating the correct path with no motion or else move a simpler mover (i.e. NOT a BlueEntity) using one of the methods described in the Astar package documentation

b. Display movement and/or paths on a simpler test scene, instead of the Cayucos Creek terrain provided.

7. Write a README that includes the names of your team, clearly indicates what parts of lab specification you have succeeded with implementing, and anything else you want me to have in mind when I test your submission.

8. Zip up the Unity project (include the Assets and Project Settings subdirectories only), including README, and upload to Sakai.