MV4025 Lab 2

The goal of this lab is to implement pathfinding that is based on the time to traverse a path (rather than its length). Additionally, you should implement tactical paths that, as much as possible, avoid observation by hostile forces (the one red entity in the scene).

0. Study the code that in AnalyticPlannerEditor.cs. You will add all of your code here. It is found in Assets\Editor. Also, study the TerrainGridGenerator.cs file in AstarPathfindingProject\Generators that generates the navigation graph. This is a custom generator that I wrote based on PolarGraphGenerator.cs, which is described in the "Writing Graph Generators" tutorial on the Astar Pathfinding Project web site.

1. Rewrite the code for the "Recompute Costs". This code replaces all the edge costs in the navigation graph. Your new costs should represent the time to transit the edge, which should be much slower on a grade. Since costs are represented by unsigned integers, we will scale the time up by a factor. You should also penalize edges that are in view of a hostile observer. You should base your penalty on the average number of observers of the start and end nodes of the edge.

2. Test your work as follows:
    a) Scan the navigation graph
    b) Press the "Compute Observer Data" button
    c) Press the "Recompute Costs" button
    d) In the A* object, cache the navigation graph (do NOT allow it to rescan, or the cost data will be overwritten) with the revised costs
    e) Now press the play button and order the blue unit to the far side of the hill. He should go around rather than over the top.
    f) Now order the blue unit to pass in view of the red unit. He should go the long way around instead.

3. Keep adjusting your code and costs to get the expected behavior described above

4. (Optional) Write code for the "Visualize Edge Costs" button. The visualization should clearly show that the cost to go up hills or in front of red are higher than the alternatives.

5. 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.

6. Zip up the README as well as the Assets and Project Settings directories of your Unity project, and upload to Sakai.