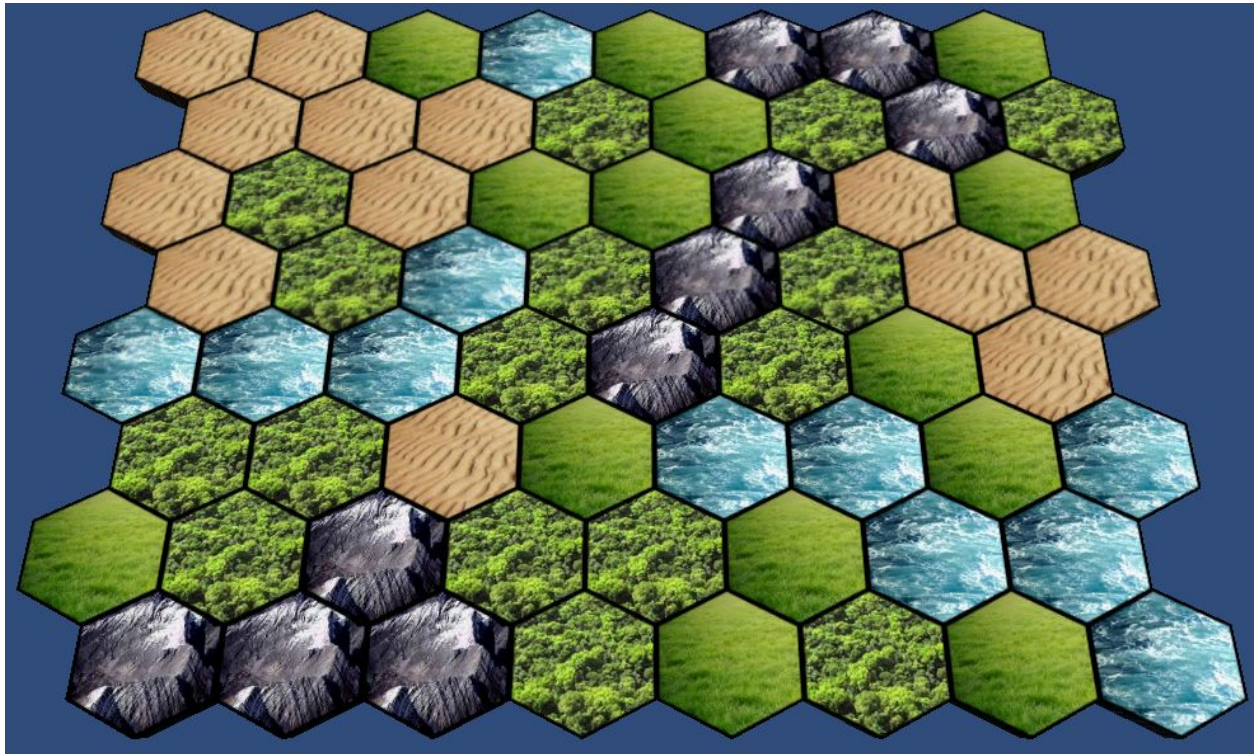


# Using A\* Pathfinding

*Do not share or distribute this test. All work must be your own.*

At Talespin, you will often work with existing code made by other programmers. In this exercise, you are asked to visualize the results of an A\* pathfinding algorithm implemented at Talespin. The implementation of the A\* algorithm is provided in a Unity project that should have been received with this document. That Unity project will also contain a model and several hexagonal textures that will allow you to construct the following map:



Please send us back the Unity project in which you have added a scene that contains a visualization of our A\* algorithm. Playing this scene should do the following:

1. Creates a map that looks like the one in the picture (this means the map tiles are **not** in the scene before pressing play!)
2. Allows the user to select a starting tile by clicking on it
3. Allows the user to select a ending tile by clicking on it
4. Shows the shortest path between the selected start and end tiles
5. From here the user should be able to continue investigating different paths

The route **must** be calculated using the GetPath function of the AStar.cs class. This means you will have to make use of the interface defined in IAStarNode.cs. These two files should **not** be modified in any way.

The map contains 5 different types of hexagonal tiles: forest, grass, water, mountain and desert. Each of these tiles has a different travel cost associated with it. These costs should be taken into account by the A\* algorithm.

Tile	Cost to travel through that tile
Grass	1 day
Desert	5 days
Mountain	10 days
Forest	3 days
Water	you cannot travel through water, ever

The resulting path should be clearly indicated on the map and visible for as long as the user likes. For example as in the following picture, where the start and end nodes are marked green and the path is marked red.



You do not have to match the path visualization of this image, as long as the path is clearly indicated on the map.