CS7056 – Lab 3 – Pathfinding

The purpose of this lab is to learn about the A* pathfinding algorithm.

You will probably want to add the code you will develop from now on to the code you have from Labs 1 and 2, so you don't have to merge the code bases later on.

As you solve the tasks, take notes about the decisions you made. You will need them for documenting your AI toolkit at the end of the semester.

If you have already implemented A* as part of the Animation Course, I suggest that you plug that A* implementation into your AI toolkit. If you do this, you can skip tasks 1 and 2 on lab 3. I will set replacement tasks in a later lab for those who do this.

Task 1: Implement a Suitable Data Structure

See 'Pathfinding I' course notes.

Design and implement a C# data structure that is suitable for representing the OPEN and CLOSED sets of nodes required for the A* algorithm. It should have the three required operations discussed in class. It is okay to offer a front-end to a data structure from C# or .NET if you like. Also give a complexity estimate of your data structure's time and space requirements using big-O notation.

Task 2: Implement A*

See 'Pathfinding I' course notes.

Design an implement the A* algorithm as shown in the course notes and handout. Use your data structure from task 2 to represent the OPEN and CLOSED set. Keep your A* algorithm generic, for example to operate on any objects that implement a particular interface.

Task 3: Extract Navigation Data from Your Game World

See 'Pathfinding II' course notes.

Add code to your game world representation (from Lab 2) that will process the game world geometry and create a navigation graph on which A* pathfinding to be performed. Depending on how you solved task 3, this could involve creating a vertex object for each tile. You will also need a heuristic, such as one of those covered in class. Add code to create the navigation graph from the random world geometry when the game starts.

Task 4: Make the Agents Move

Add logic to your agents that allows them to move across the landscape using A* to find paths. You may need to add a new state to your agents' state model in which they move towards their target. Think about the best way to do this without introducing too many new states. Use the shading function from Lab 2.4 to show the path that a given agent is taking when it moves.