Path Finding Documentation

[student name]   Advanced Diploma of Professional Games Development
Assessment 3 (Game Artificial Intelligence)

Overview

For this assessment we are required to demonstrate …..

Structure of Graph Data

- Provide 3 methods of storing the data for nodes and edges.
- Show code snippets for each structure used
- for each method, specify what type of data containers you will be using for nodes and edges and why. (List, Vector, HasMap, Deck, Queue, 2D array etc)

Graph Data Algorithms

From the above methods listed, choose one, and define how the following functions can be implemented.

- **Finding a node in the graph based on a condition.** *(provide code and written explanation)*
  This should be a generic condition that can be changed at runtime, therefore, you could use lambda functions or function pointers for customising the condition. How will you be returning the results.

- **Connecting 2 nodes within the graph** *(provide code and written explanation)*
  What are some potential errors that could occur when connecting nodes? Will the pathfinding algorithms you implement work as expected if the same 2 nodes have duplicate edges? or, could duplicate edges work to your advantage? Why?

- **Adding Nodes to the graph.***(provide code and written explanation)*
  How is the Nodes updated, what are some potential errors? eg 2 nodes placed on top of each other? is this a problem, or could this be used to your advantage? How/Why?
  Will edges be automatically generated with other nodes? What is the criteria for connecting nodes automatically?

- **Finding Neighbors of a given node** *(provide code and written explanation)*
  How are neighbouring nodes stored within your graph, and how will you return a connected nodes. What return method will you use? custom iterator, return via reference or return val. Maybe you provide an alternative method.

- **How can this data be saved and loaded from file.** *(provide code and written explanation)* what file types could you save as? XML, JSON, Custom Text document, Custom Binary data. Saving and Loading methods.

PathFinding

- How have you implemented the Dijkstra's Path finding algorithm, how is the the path calculated? (*Provide pseudocode and supporting C++ code Snippets)*

-
- How is the path stored, what data structures are you using, and what functions are available for manipulating / getting information from your path? (*Provide pseudocode and supporting C++ code Snippets)*

  - Path Smoothing, what are some methods you could use?

- How have you modified the Dijkstra's pathfinding Algorithm to perform the AStar pathfinding algorithm? (*show C++ code Snippets)*

- What are some scenarios you would use the AStar Pathfinding algorithm?

- What are some scenarios you would use the Dijkstra's Pathfinding Algorithm.

- What are the Pro's and Cons for both Dijkstra's and AStar pathfinding?

Implementation

Show a small code snippet for how you will find a path using your Pathfinding Algorithm. Here is an example implementation: This implementation will allow for multiple paths to be calculated simultaneously on the same graph.

```
// Member variables
Path *m_path                   = nullptr;
PathFinder *m_pathUpdater      = nullptr;

// called when you want to start finding a path
//---------------------------------------------------------
m_pathUpdater = m_graph->FindPath( StartNode, EndNode );
      // for Dijkstra's  the end node could be a condition
      // if a node meets the condition, than a path
      // should be returned to that node.
//---------------------------------------------------------

// during your update loop
if( m_pathUpdater != nullptr )
{
      pathUpdater->Update();
      if( pathUpdater->IsPathFound() )
            path = pathUpdater->CalculatePath();
}

if( m_path != null )
{
      // TODO: make an entity object follow the path
}
```

References

List all webpages, textbooks, videos you have used as part of your reasearch for these pathfinding algorithms.