Assignment 4: Pathfinding

Brian Baron

Architecture for this project is built on the base Pathfinding Solution from the EGP-410 repo. The project was already linked, but had a compiler error that was fixed in one of the early commits.

Pathfinding Algorithms

Both Algorithms begin at the start node, and expand through the grid space until the shortest path to the goal node is found.

Dijkstra

Breadth first search, ‘wavefront’ that fills the area as it searches. Returns the shortest path upon completion. More costly than A* to run (in most cases), because of the lack of heuristic analysis.

A* (AStar)

A* Implements similar behavior to Dijkstra, but includes a heuristic analysis to prioritize specific nodes. This iteration of A* checks for the shortest Euclidian distance to the goal node, and puts the node at the end of that connection at the front of the ‘nodes to visit’ list. This way it is the next node evaluated, possibly saving time (by preventing the search of unnecessary nodes).

InputManager/Messaging

The InputManager sends messages that affect the game state upon input. These messages are then handled by the message manager. This avoids direct linking from the InputManager to the Game/GameApp or other such code bases.