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 reaching the goal. Once it is found, the path is built backwards towards the start node.

Dijkstra

Breadth first search, ‘wavefront’ that fills the area as it searches. More costly than A* to run (in most cases), because of the lack of heuristic analysis. Will search a lot more nodes, but gives a more reliable path (as far as shortest path computing is concerned).

A* (AStar)

A* Implements similar behavior to Dijkstra, but includes a heuristic analysis to prioritize specific nodes. This iteration of A* uses the Manhattan Distance heuristic, which is admissible. Since the grid provides equal weights to all of the nodes, I ignore the coefficient multiplying against the sum of the deltaX and deltaY (it would be 1). Code includes link to a reference on the heuristic.

A*’s benefit is that it explores fewer nodes than Dijkstra, but this is a trade-off for accuracy. The path returned by A* is not guaranteed to be the shortest path.

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.