Phase 1 Report

Matthew Ya & Aleece Randall

February 5, 2017

1 Quick How-to for Interface and Images

Colors:

Blocked: Black
Regular unblocked: White
Hard to traverse: Orange
Regular unblocked with a highway: Cyan
Hard to traverse with a highway: Blue
Start: Lime Green
Path: Purple
End: Red

2 Implementation of Algorithms

3 Optimization

The data structures used to implement the algorithms were chosen to achieve the best time and space complexities. The fringe or open list is a priority queue that is implemented using a binary heap. This data structure provides worst case complexities of O(log N) for search, inserts, and delete and O(1) for removal of the cell with the lowest value. The successors of each node are stored in a HashSet because of its O(1) insert. A HashSet was also chosen for the closed list after time/space complexity tradeoff. The HashSet has an average case search time of O(1) but worst case time O(n). The HashSet was chosen over a 2D boolean array due to the array's higher space requirement. Although, the Boolean array would yield a better worst case search time of O(1) , the space complexity is always the total number of nodes in the graph. On average, the HashSet is a much smaller value. The HashSet and Priority Queue were initialized to a higher but reasonable initial capacity in an attempt to reduce possible rehashing.

4 Heuritistics

There are several possible ways to calculate the distance between the start node and goal node. Let dx represent the horizontal distance and let dy represent the vertical distance between the start node and goal node. Let D represent the vertical/horizontal cost and D2 represent diagonal cost.

4.1 Euclidean Distance Formula

$$Distance = D2 * \sqrt{dx * dx + dy * dy} \tag{1}$$

Pro: considers diagonals
Con: computationally expensive, only considers a straight line

4.2 Manhattan Distance Formula

$$Distance = D * (dx + dy) \tag{2}$$

Pro: computationally inexpensive
Con: doesnt consider diagonal movement

4.3 Diagonal Distance Formula

$$Distance = D * (dx + dy) + (D2 - 2 * D2) * min(dx, dy) \tag{3}$$

Discussion about picking D and D2 (average cost, lowest cost)

4.4 The Chebyshev Distance Formula

$$Distance = (dx + dy)2 * min(dx, dy) \tag{4}$$

4.5 The Octile Distance Formula

$$Distance = (dx + dy) + (\sqrt{2} - 2) * min(dx, dy) \tag{5}$$

5 Experimental Results from 50 benchmarks

6 Discussion