Sean Carnahan 915227211 February 2nd, 2020 ECS 170

1 Proof for AStarDiv and AStarExp

let mC represent the minimumCost = the lowest possible cost between two point

Run A* with Dixtra as heuristic.

start with mC = 1, if the cost comparison between any two points is less than mC than this becomes your new mC.

Run this against all possible seeds and then take the minimum of the minimums and this becomes your new mC.

double minCost = 0.8505747126436781. // for AStarDiv double minCost = 7.62939453125E-6. //for AStarExp

For the the Divison pathCost = return (double) (getTile(p1)) / ((double) getTile(p2)+1.0);

For the the Exponential pathCost = Math.pow(2.0, (getTile(p2) - getTile(p1)));

By this account the actual PathCost will be very close to a straight line.

Therefore, we would want to use Euclidian Distance. However Euclidian distance is not admissible because its under the assumption that the path cost between each tile will be 1. Since the PathCost of our program fluctuates and is ¡ 1. Euclidian Distance is not an admissible heuristic.

The calculation for Euclidian Distance:

Math.sqrt(Math.pow(p1.x - p2.x, 2) + Math.pow(p1.y - p2.y, 2));

Now if you multiply the Euclidian distance by the minCost you will have an admissible heuristic because there will never for be an overestimate

Here is the Mathematical proof by induction that proves this:

Prove f(n) >= h(n)

f(n) = sum of all costs along the optimal path

$$= \sum_{i}^{j} k_i$$

where optimal path goes from i to j and

$$cost(k_i)$$

is the cost to move to point

$$k_i$$

h(n) = min (path.length) * min(cost(k))

$$\sum_{i}^{j} cost(k) >= min(path.length) * min(cost(k))$$

$$\sum_{i}^{j} cost(k) = \sum_{i}^{j} min(cost(k))$$

INDUCTION: for all

$$cost(k_i) >= minlength$$

= optimalPath.length * min(cost(n))
= min(path.length) * min(cost(n)) - INDUCTION:

$$optimal >= minpathlength$$

Therefore

$$f(n) >= h(n)$$