## CS4710 HW#2 Write Up

Bryan Chen (bc2vf)

Hans Zhang (kz6ef)

Describe your basic path finding algorithm. Show a brief analysis of how well it works on a few different datasets that you produced. What kinds of data sets are more inefficient? Why is that the case?

Our A\* path finding algorithm has the following steps:

1. Divide evaluated and unevaluated Points (nodes) into to lists
2. Create a map that uses one node as key and its parent node in the path as value
3. Create maps for nodes to store f score and g score
4. Begin to evaluate all unevaluated nodes:
   1. Return the path if current node is our goal
   2. Else, mark it as evaluated and generate all neighbors
   3. For each adjacent node to the current, evaluate the f Scores and g Scores.
5. If there is no path found, return an empty list.

The algorithm performs well on both small and big datasets, with either majority of walls or reachable nodes.

Describe how you adapted your algorithm when dealing with uncertain situations. How did you deal with the fact that the robot sometimes incorrectly viewed a space in the world?

We used heuristic for dealing with uncertainty situations.

Produce data that shows how well your algorithm performs on different inputs. What happens if you slightly tweak or change your algorithm? How do these changes affect the performance and why?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Size (L x W) | Time to generate path (ms) | Total # of moves | Total # of moves | Reached destination or not |
| 2 x 2 | 14.20 | 1 | 3 | Yes |
| 2 x 4 | 12.41 | 3 | 7 | Yes |
| 4 x 8 | 16.49 | 7 | 53 | Yes |
| 10 x 10 | 99.97 | 13 | 144 | Yes |
| 20 x 20 | 57.13 | 38 | 418 | Yes |
| 40 x 40 | 80.73 | 43 | 405 | Yes |
| 100 x 100 | 3975.85 | 198 | 6501 | Yes |