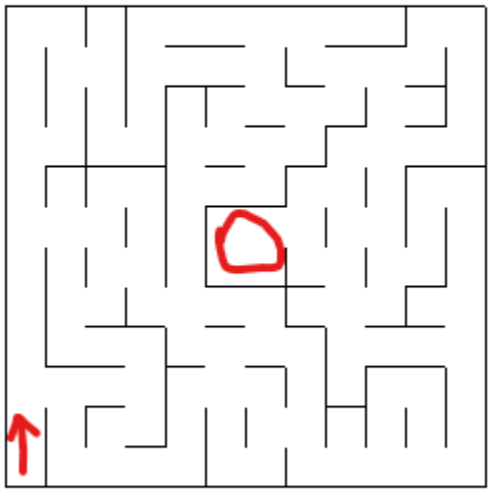
Nanodegree Capstone Project: Plot and Navigate a Virtual Maze project.

By: Mohlomi Pholoana

Data Exploration:

Below is a snippet of how one of the test_maze – 01 looks like and the robot's starting position is always bottom left – 1 = 1*1+0*2+0*4+0*8 and the first only move is to go up, minding all the obstacles ahead of the next move to take and considering the shortest path to the goal all while trying not to exhaust all of moves limit (1000 = max_time). The robot self learn from previous paths/ moves explored so that it find the best and shortest paths to the goal. So the next_move function is very critical to the movement of the robot. It makes use of direction sensors that explores paths available on the next box, and if there are no obstacles and whether rotation to towards an open path will be possible all while move_time/ number of moves still isn't equal to max_time and goal found = false.

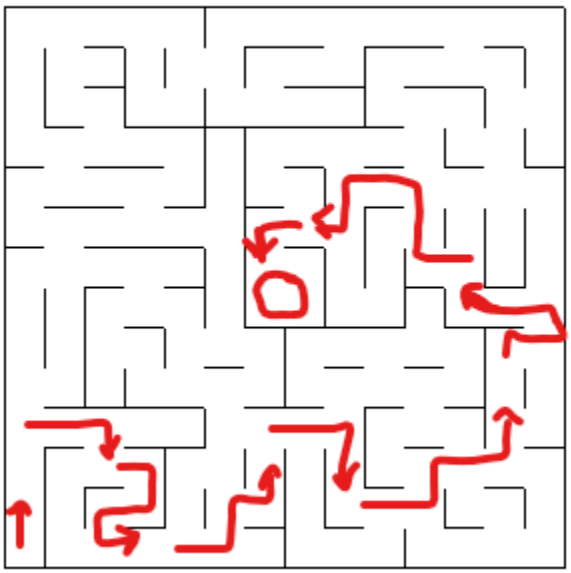


Exploratory Visualization:

The robot uses heuristic exploration algorithm to quicker get to the goal in 25 moves for the second run and also the considering the fact that test_maze_02 is 16X16, the bigger of the 3 mazes. Different algorithm like BFS, Dijkstra's and A* were explored but I found the heuristic to be the best for my purpose since it is faster and efficient, scalable, and practical. A* was slower and I got "allotted time" several times while trying it out. On the other hand, BFS was faster but

need max_time to be be increased in order to explore all optional paths available before it can decide on the shortest route.

Below is the demonstration of the text_maze_02 and textfile which defines the 16X16 maze



16

1,5,5,6,1,3,5,5,5,5,7,5,5,5,5,6

0,0,0,10,2,9,6,3,5,5,13,7,5,5,6,10

0,0,0,11,3,5,14,8,2,3,5,13,5,6,10,10

0,0,0,10,11,5,13,5,12,9,7,6,3,15,13,14

0,0,0,9,12,0,0,2,2,0,10,11,14,11,6,10

0,0,0,0,0,0,2,3,6,0,9,12,11,12,10,10

0,0,0,0,0,3,7,12,9,5,6,3,13,7,12,10

0,0,0,0,1,10,8,2,2,0,10,9,7,13,7,12

0,0,0,0,1,9,7,5,4,3,14,3,13,7,13,6

0,0,0,0,0,1,10,3,6,10,10,10,3,12,5,8

0,0,0,0,0,0,11,14,10,8,10,10,10,8,0,8

0,0,0,0,0,1,10,11,15,6,9,12,14,0,0,0

0,0,0,0,0,0,9,14,10,10,3,5,12,4,0,0

0,0,0,0,0,0,1,11,14,10,10,8,8,0,0,0

0,0,0,0,0,3,5,12,10,9,12,4,0,0,0,0

0,0,0,0,1,9,5,5,12,12,0,0,0,0,0,0

Benchmark

My expectation was because the max number of moves/ runs had been put as 1000 and considering the fact that this was my first challenge Project, I had anticipated to get "Allotted time exceed" on the first run and I will have to update and try different algorithms and fine tuning until now the goal was found at 900 runs.

Data Preprocessing

No data preprocessing was required in this case since the sensor specification and environment design are provided to me.

Improvement

If we are to adapt the robot's code to handle a continuous domain introduces several problems. One of them would be the robot's position which will need to be represented with float coordinates and the movement will have to be in continuous instead of from one cell to the next. We will also have to consider thickness of the walls and size of the robot while demining next move. We would also have to use path planning algorithms like A* that works in a continuous space.

The Robot would also have to consider its dimensions and thickness of the walls around before next move and implement a localization algorithm (e.g., Particle Filter, Kalman Filter) to accurately estimate the robot's position in the continuous domain.

References:

1. pdf-libre.pdf (d1wqtxts1xzle7.cloudfront.net) (Comparative Analysis of Pathfinding Algorithms A *, Dijkstra, and BFS on Maze Runner Game vol 1, by Silvester Dian Handy Permana1 , Ketut Bayu Yogha Bintoro2 , Budi Arifitama3 , Ade Syahputra4)

2. ml-nanodegree-capstone/Plot and Navigate a Virtual Maze.pdf at master · eminnett/ml-nanodegree-capstone (github.com) (6th June 2024)