Machine Learning Engineer Nanodegree

Capstone Proposal for

Robot Motion Planning

Plot and Navigate a Virtual Maze

By

Halimat Mercy Oseni

February 25th, 2018

DOMAIN BACKGROUND

The inspiration for this project is the Micromouse competitions which are held worldwide and are most popular in the UK, US, Japan, Singapore, India and South Korea. Micromouse is an event where a small autonomous robot mouse solve a 16×16 maze. It began in the late 1970s, although there is some indication of events in 1950. The micromouse is expected to move around the unfamiliar maze and find its way to the designated goal which is commonly the middle of the maze starting from a designated corner of the maze.

The micromouse is allotted two(2) attempts with the maze; the initial attempt is for exploration, where the mouse keeps track of its position, discover walls, map out the maze and detect when it has reached the goal and the final attempt is for optimization, which is mainly for the mouse to find an optimal route and shortest possible time to get to the goal.

I am interested in this project because I'm fascinated by how robots perform mind-blowing activities and I'm more than excited in building mine.

PROBLEM STATEMENT

The goal of this project is for the micromouse to get to the center of the maze in the fastest time possible. The maze is a square with equal rows and columns having $n \times n$ grid, where n is the number of squares along each side and it must be 12, 14 or 16. The outside perimeter of the grid and the edges connecting some of the internal squares are blocked walls which prevents the micromouse from travelling outside the maze. The starting point is in the bottom-left corner of the grid, facing upwards (top) as the bottom, left and right side of the grid are blocked, hence the initial move will always be forward. The goal area of the maze is at the center having a 2×2 square.

As stated earlier, the micromouse will run the first of the two(2) trials which is for exploration. The exploratory goal is for the mouse to understand the structure and shape of the maze and also the shortest possible time to get to the center (goal area) of the maze. An exploratory trial is successful when the mouse gets to the goal area. However, the mouse can also continue to explore the maze. The micromouse is then moved back to the starting position and set for the second trial. In the second trial, the micromouse is expected to use the previously learned knowledge to get to the goal using an optimal route and shortest possible time. The maximum number of steps for each trial is 1000 and on each step of the trial, the micromouse can only rotate clockwise or counterclockwise ninety (90) degrees and can move forwards or backwards a distance of up to three units.

The micromouse performance is calculated by summing:

- One-thirtieth (1/30th) of the total steps taken in the exploratory trial and

- The total number of steps taken in the optimization trial

DATASETS AND INPUTS

The dataset for the maze layout is provided in a text file. The first line of the text file is a number that describes the number of squares on each dimension of the maze. The following lines describe the actual structure of the maze with the edges that are open for movement and the numbers are separated by commas. Each number represents a four-bit number that has a bit value of 0 if an edge is closed and 1 if an edge is open; the 1s register to the upwards-facing side (top), the 2s register the right side, the 4s register the bottom side and the 8s register the left side.

> ***For example: If all edges are open, we'll have***
> ***($1\times 1 + 1 \times 2 + 1 \times 4 + 1 \times 8$) which is equal to 15***

The micromouse movement follows a rotation which can be -90, 90 or 0 and each movement is an integer which ranges from -3 to 3. The micromouse must always have knowledge of its position in the maze, the number of walls surrounding it, its intended action and information from previous trials. The code that constructs the maze is part of the project prerequisites provided by Udacity and can be found in the maze.py file.

SOLUTION STATEMENT

The two(2) trials for this project attempts to solve different problems, the first trial questions the overall structure of the maze and the second trial ensures to reach the goal in an optimized manner.

For the first trial, a reasonable solution will be for the micromouse to have a detailed knowledge of the maze layout and all possible paths that leads to the goal area while for the second trial, it will be to reach the goal in the shortest time possible.

The solution is quantifiable and can be mathematically expressed as the total number of moves and total time taken by the micromouse to travel through the maze is constantly calculated. It is also measurable because at the end of the exploration trial, the micromouse is expected to have recorded knowledge of the maze and the different possible routes to reach the goal and during the optimization trial, it is expected to choose the best and fastest route. The solution is also replicable, the optimization trial can be repeated for more rounds and the same result is expected.

BENCHMARK MODEL

For my benchmark model, I plan to use the Depth-first search and Breadth-first search algorithms to train and test the project data. Depth-first search algorithm performs graph-search by building the search-tree as deep as possible and as fast as possible while Breadth-first search is similar to the depth-first search but it attempts to make the search tree as broad as possible and as quickly as possible. Then I'll compare the results of the algorithms with my final solution.

EVALUATION METRICS

As previously discussed in the Problem Statement and Benchmark Model above, the evaluation metrics to be used to quantify the performance of both the benchmark model and the solution model is:

Performance score = (1/30 of A) + B

Where A is the total number of steps taken in the exploratory trial(first trial) and
B is the total number of steps taken in the optimization trial(second trial)

The evaluation metric stated above is evaluated by both the exploratory and optimization trial, although the optimization trial will have a greater impact on the performance score compared to the exploratory trial as the goal is to minimize the score which is the result of an optimal trial.

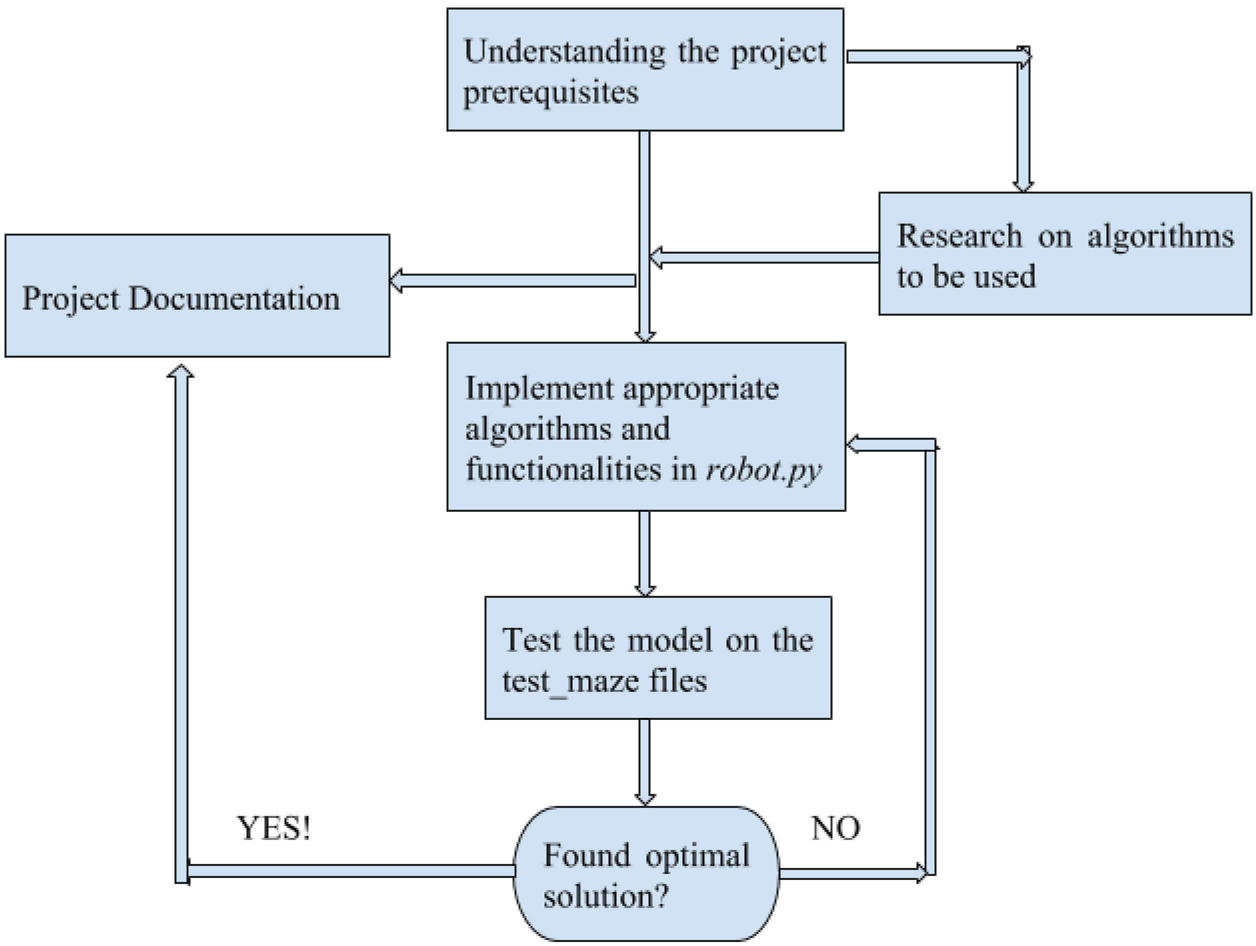
For example: If the micromouse took 660 steps and 25 steps during its exploratory and optimization trial respectively. The performance score would be calculated as -

Performance score = $(1/30 \times 660) + 25 = 47$

Hence, the optimal score for the example is 47.

PROJECT DESIGN

A theoretical workflow for approaching the solution for this project commence with and not limited to understanding the prerequisites provided, researching on the algorithms to be used and also documenting the whole process so as to have a detailed project report. The prerequisites also include starter codes - maze.py, robot.py, showmaze.py, tester.py and test_maze_##.txt. After going through the starter codes, I'll run the project to view the decoded virtual maze. Then I'll start working on the robot.py file (which is where I'm going to have all my code implementation) with the appropriate algorithms and functionalities. The next step is to run the algorithms with the defined benchmark model and keep records of the performance. I'll test my model on the test_maze files using both the tester.py and showmaze.py to view how the project runs in the terminal and the visual representation of the maze respectively. I'll ensure to keep track of the scores and I expect the micromouse not to find an optimal solution with my first code implementation. I'll therefore revise my implementation and test again until the micromouse finds the optimal solution. Below is a simple pictorial representation of my workflow:



Some of the algorithms I consider to use for this project include Flood fill, Follow wall, Block deadend, Breadth-first search, Depth-first search, A* and Dead reckoning.

REFERENCES

Udacity project files
https://docs.google.com/document/d/1ZFCH6jS3A5At7_v5IUM5OpAXJYiutFuSIjTzV_E-vdE/pub

History of Micromouse
https://en.wikipedia.org/wiki/Micromouse

APEC Micromouse competition
https://www.youtube.com/watch?v=0JCsRpcrk3s

Benchmarks in Robotics Research
https://www.staff.science.uu.nl/~gerae101/pdf/benchmarks1.pdf

Robot Competitions - Ideal Benchmarks for Robotics Research
https://www.ais.uni-bonn.de/nimbro/papers/IROS06WS_Benchmarks_Behnke.pdf

Robot Motion Planning
https://books.google.com.ng/books?hl=en&lr=&id=nQ7aBwAAQBAJ&oi=fnd&pg=PR9&dq=Robot+motion+planning&ots=7oh-m1pOd2&sig=fkNvfZe1RNHtFzu4CIf7RW26RcY&redir_esc=y#v=onepage&q&f=false

http://ais.informatik.uni-freiburg.de/teaching/ss11/robotics/slides/18-robot-motion-planning.pdf

Analysis of Micromouse Maze Solving Algorithms

http://www.academia.edu/8226457/Analysis_of_Micromouse_Maze_Solving_Algorithms

Modern Machine Learning Approaches For Robotic Path Planning
http://ijcsit.com/docs/Volume%208/vol8issue2/ijcsit2017080226.pdf

A Survey of Machine Learning Approaches to Robotic Path-Planning
https://www.cs.colorado.edu/~mozer/Teaching/Computational%20Modeling%20Prelim/Otte.pdf

AISB Quarterly issue 39 Dec.1980 on micromice and the first european micromouse competition
http://davidbuckley.net/RS/mmouse/AISBQ39.htm