Robot Motion Planning Capstone Project

Plot and Navigate a Virtual Maze

# Capstone Proposal

James Tooles  
February 19, 2018

## I. Definition

### Project Overview

This project’s [core outline](https://docs.google.com/document/d/1ZFCH6jS3A5At7_v5IUM5OpAXJYiutFuSIjTzV_E-vdE/pub)1 was created by Udacity and the idea for this project is mainly derived from [Micromouse](https://en.wikipedia.org/wiki/Micromouse)2 competitions. In the competitions, a robot mouse agent is given the task of planning and plotting the best path from a corner of the environment (a maze) to its center. The purpose of the competition was for the agent to discover the environment during the first run and to use the best planned path to reach the center of the environment in subsequent runs. The goal of this project is to emulate these constraints of the completion and achieve the goal of the agent obtaining the fastest times possible in a series of test environments. The competition places the same restrictions on the agent as this project’s restrictions, which will be covered in the following sections.

### Problem Statement

This project is tasked with solving the problem of a robot mouse agent plotting a path from the bottom left corner of a maze to its center. The agent is allowed to embark on two runs of any given maze. The first run of the maze, the agent will explore, map, and analyze the environment to determine the best path plans to reach the center of the maze. The second run of the same maze, the agent will attempt to navigate the environment as quickly and efficiently as possible in order to reach the center of the maze.

### Metrics

The agent will explore multiple environments and must complete two runs on each. During the first run of the environment, the agent will be allowed to freely explore in order to build a map of the environment. At someone point during the exploration, the agent must enter the goal area but, the agent is free to continue exploring the environment after entering the goal area.

The second run of the environment, the agent will be returned to the starting position and orientation. The agent’s goal is to then navigate to the goal area in fastest time possible, minimizing actions (time steps) taken by the agent.

The agent’s score is the number of time steps required to execute the second run, plus one thirtieth the number of actions taken during the first run. The maximum actions allowed for a completion of both runs is one thousand actions for both runs for a single environment.

Score = #actions2 + (1/13) \* #actions1

where lower Score is better

## II. Analysis

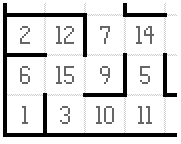
### Data Exploration

The dataset and inputs for this project consists of the environment, the maze. The maze is an environment of an n x n grid of squares, where n is even. The environment can have an n value range of twelve to sixteen. The state space for this problem is equal to n \* n \* 4, where n is the size of the environment and 4 is the number of actions the agent can take in any given location within the environment. The state space is between, 576 and 1024, for environment sizes of 12 x 12 and 16 x 16 respectively.

The environments are accessed through text files. The first line of the text file is a number which describes the number of n squares within the environment. The first data row in the text file is the leftmost column of the environment, with the first element being the bottom-left corner of the environment. The remaining n lines will be n comma-delimited numbers which describe the passible (open) edges of the environment. Each number represents a four-bit number that has a bit value of 0 if an edge is impassible (closed) and 1 if an edge is passible (open). The 20 is the upward side, 21 is the right side, 22 is the bottom side, 23 is the left side.

Ex. 1010 ⬄ (23 + 0 + 21 + 0) ⬄ (8 + 0 + 2 + 0) ⬄ 10

{Left: Open, Bottom: Closed, Right: Open, Upward: Closed}



The environment will always have impassible walls on the outer board of the grid, which blocks all movement. Walls will be placed throughout the environment to block movement of the agent, creating a maze. The robot agent will always start in the bottom-left corner square of the environment grid, and be oriented upwards. The starting square will always have walls on the left, right, and bottom, which will allow for the first action to always be upwards. The environment will have a center within the environment grid consisting of a 2 x 2 square, this is the goal for the agent to position itself within the center 2 x 2 square.

### Exploratory Visualization

In this section, you will need to provide some form of visualization that summarizes or extracts a relevant characteristic or feature about the data. The visualization should adequately support the data being used. Discuss why this visualization was chosen and how it is relevant. Questions to ask yourself when writing this section:

* *Have you visualized a relevant characteristic or feature about the dataset or input data?*
* *Is the visualization thoroughly analyzed and discussed?*
* *If a plot is provided, are the axes, title, and datum clearly defined?*

### Algorithms and Techniques

The agent is given the problem of solving a randomized maze environment in the least amount of actions. The agent is given access to the environment dimensions initially, then allowed to explore the environment in the first run. The second run, the agent should utilize the optimal path in order to take the minimum actions necessary to reach the goal. Thus, receiving the best (lowest) score possible during the second run. The following theoretical high level workflow will be taken:

1. Dimension of environment will be taken in on the initialization of the agent
2. The goal location area will be derived from the environment dimension
3. First run of the environment:
   1. The agent will explore the environment
      1. This could be done with random actions, actions that prefer to move towards the middle until encountering a wall, or another means of selecting actions
   2. The agent will record the environment during exploration, i.e. where there are walls
   3. The agent will not attempt to run into walls discovered
   4. The agent will reach the goal
      1. The agent can stop exploration
      2. The agent can continue exploration until the environment is completely discovered
4. The path planning algorithm will take in the recorded environment data and decide the optimal path for the agent
   1. [Decision Tree](https://en.wikipedia.org/wiki/Decision_tree_learning)3
   2. [Neural Network](https://en.wikipedia.org/wiki/Artificial_neural_network)4
   3. [Q Learning](https://en.wikipedia.org/wiki/Q-learning)5
5. Second run of the environment:
   1. The agent will follow the path planned by the algorithm
   2. The agent will stop once the goal is reached

### Benchmark

The Benchmark Model for the agent directly relates to how the evaluation metrics are done for scoring the agent’s path planning regarding steps taken within the environment. The agent is restricted to a total of one thousand time steps total for both runs within the environment. One baseline model would be that of taking five hundred actions for each run, maximizing the allowed one thousand time steps. These actions could be random actions to create a true baseline model, which would be random guessing path planning. If the solution for the agent does better than this baseline model, it would be a successful model. The following score would result for the baseline model:

Score = #actions2 + (1/13) \* #actions1

ScoreBaseline = 500 + (1/13) \* 500

ScoreBaseline = 538.46

## III. Methodology

*(approx. 3-5 pages)*

### Data Preprocessing

In this section, all of your preprocessing steps will need to be clearly documented, if any were necessary. From the previous section, any of the abnormalities or characteristics that you identified about the dataset will be addressed and corrected here. Questions to ask yourself when writing this section:

* *If the algorithms chosen require preprocessing steps like feature selection or feature transformations, have they been properly documented?*
* *Based on the****Data Exploration****section, if there were abnormalities or characteristics that needed to be addressed, have they been properly corrected?*
* *If no preprocessing is needed, has it been made clear why?*

### Implementation

In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section:

* *Is it made clear how the algorithms and techniques were implemented with the given datasets or input data?*
* *Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution?*
* *Was there any part of the coding process (e.g., writing complicated functions) that should be documented?*

### Refinement

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section:

* *Has an initial solution been found and clearly reported?*
* *Is the process of improvement clearly documented, such as what techniques were used?*
* *Are intermediate and final solutions clearly reported as the process is improved?*

## IV. Results

*(approx. 2-3 pages)*

### Model Evaluation and Validation

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model’s solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section:

* *Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate?*
* *Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data?*
* *Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results?*
* *Can results found from the model be trusted?*

### Justification

In this section, your model’s final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section:

* *Are the final results found stronger than the benchmark result reported earlier?*
* *Have you thoroughly analyzed and discussed the final solution?*
* *Is the final solution significant enough to have solved the problem?*

## V. Conclusion

*(approx. 1-2 pages)*

### Free-Form Visualization

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section:

* *Have you visualized a relevant or important quality about the problem, dataset, input data, or results?*
* *Is the visualization thoroughly analyzed and discussed?*
* *If a plot is provided, are the axes, title, and datum clearly defined?*

### Reflection

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section:

* *Have you thoroughly summarized the entire process you used for this project?*
* *Were there any interesting aspects of the project?*
* *Were there any difficult aspects of the project?*
* *Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?*

### Improvement

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section:

* *Are there further improvements that could be made on the algorithms or techniques you used in this project?*
* *Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how?*
* *If you used your final solution as the new benchmark, do you think an even better solution exists?*

### References

1. <https://docs.google.com/document/d/1ZFCH6jS3A5At7_v5IUM5OpAXJYiutFuSIjTzV_E-vdE/pub>
2. <https://en.wikipedia.org/wiki/Micromouse>
3. <https://en.wikipedia.org/wiki/Decision_tree_learning>
4. <https://en.wikipedia.org/wiki/Artificial_neural_network>
5. <https://en.wikipedia.org/wiki/Q-learning>