Group Assignment - Micromouse

MTRN3100 - UNSW School of Mechanical and Manufacturing Engineering

Learning Outcomes

- Apply relevant theoretical knowledge about mobile robots including locomotion, perception, and localisation utilising onboard sensors, navigation, and path planning, for complex problem-solving.
- Apply general computer vision techniques for feature/object detection and tracking, for complex problem-solving.
- Demonstrate hands-on skills in mechatronics design, fabrication, and implementation by completing practical activities.
- Collaborate effectively within a team via participation in a problem-solving competition.

1 Objective

The objective of the micromouse challenge is to build a robot that navigates through a maze in the shortest time. Each team's run will be recorded and ranked to decide an overall winner.

1.1 Robot Specification

- The robot must be powered by the provided 2s Li-Ion battery pack, with the appropriate fuse.
- All software processing must be done with the onboard Arduino nano microcontroller unless otherwise specified (computer vision for path generation).
- The robot must fit within a cylinder 150 mm wide and 200 mm tall.
- The robot must not use permanent fastening techniques e.g. epoxy and superglue.

Sufficient robot components are supplied however you are allowed to include your own components within reason, If you are unsure if additional components are allowed please contact course staff. Replacement parts will not be provided unless you can prove your component was damaged through no fault of your own.

1.2 Maze representation

You are free to represent the maze however you choose, However, A source and destination location will be provided in the following format. The maze is made up of cells in a 9 x 5 configuration. Each cell is 250mm x 250mm and the grid is enclosed (Figure. 1).

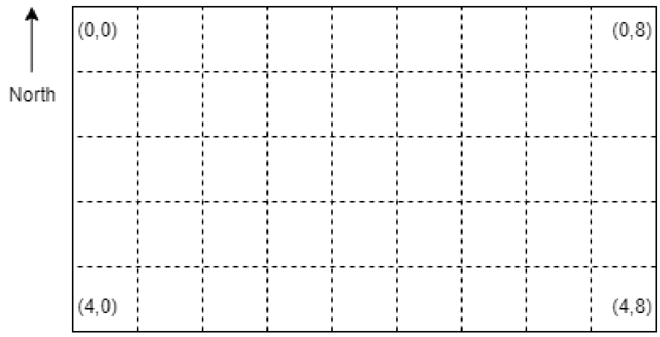

Figure 1: Representation of the maze grid with coordinates.

The start position is given in the configuration (row, column, direction) which indicates the robot's initial row and column as well as direction (N, S, E, W). If the start position is (2,7,N) the robot will be starting in the 2nd row, 7th column pointing north. The goal position is given in the format (row, column) which indicates the target row and column.

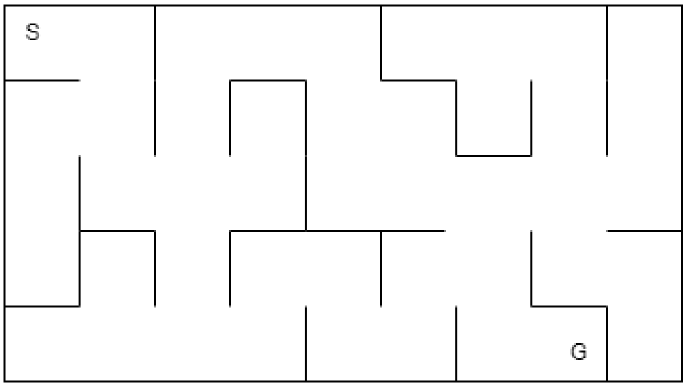

Figure 2: Example maze layout for the final competition. S represents the start location at cell (0,0). G represents the goal location at cell (4,7).

2 Simple driving

The simple driving task involves 3 different trials. You may compile and upload code between trials, however no significant changes will be allowed. A significant change is one requiring more than 5 minutes to implement. Figure. 3 shows an example of what the maze may look like for the simple driving assessment. This assessment will be marked during your week 8 lab time, and worth 10% of your final course mark.

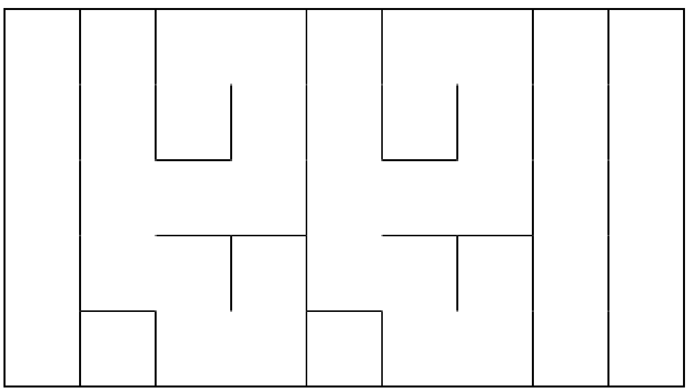

Figure 3: Example maze configuration for the simple driving assessment. Showing the rotation cells, straight lanes, and partial maze.

2.1 Drive Straight - 3%

For the straight driving task, you must be able to demonstrate your ability to drive straight for a longer period. The test will take place on a straight map of 5 cells with walls on either side. There will be a wall at the end of the straight to assist in robot alignment.

0.5 marks will be awarded for each square reached by the robot (For a maximum of 2.5 %). If the robot reaches the final cell and the turning point of the robot is within 25mm of the center of the cell 0.5 marks will be awarded. A collision will terminate your run. Each team receives a maximum of 2 runs for this task, with your best attempt used for marking.

2.2 Turning - 3%

For the turning driving task, you must be able to demonstrate your ability to accurately turn around in a cell. This test takes place within a single cell with walls on all 4 sides. Your robot must turn 90 degrees 4 times in a direction given to you by the demonstrator (clockwise, counter-clockwise).

You receive full marks if your robot's final position is within +/- 5 degrees of the correct direction, 2 marks if you are within +/- 10 degrees, and 1 mark if you are within +/- 20 degrees. Each team receives a maximum of 2 runs for this task, with your best attempt used for marking.

2.3 Chaining Movements - 4%

You will be provided a string of 8 movements which your robot must be able to execute in a standard maze environment. For the maze shown in Figure. 3, the robot may start at (1,1,S) and be provided the string "fflfrflf". This string would be the command sequence "Foward, Foward, left, Foward, Right, Foward, Left, Foward". In which Foward indicates moving one cell, Left is a turn 90 degrees counter-clockwise, and Right is a turn 90 degrees clockwise.

You receive a 0.5 mark for each of the actions successfully taken by the robot. A collision will terminate your run. Each team receives a maximum of 2 runs for this task, with your best attempt used for marking.

2.4 Submission

Marking will occur during your Week 8 Lab. Teams will be called to the front to be marked. It is expected that you turn up to the lab ready to get marked. Marks will be awarded at the discretion of the demonstrator. At minimum one team member must be present, If no team members are available please contact the course staff. A team evaluation task will be available on moodle after the competition of this task.

All the code used for the simple driving task must be pushed to your team's GitHub repository in a branch called "simple driving".

3 Micromouse

The micromouse task involves 3 different tasks. You may compile and upload code between trials, however no significant changes will be allowed. A significant change is one requiring more than 5 minutes to implement. Figure. 2 shows an example of what the maze may look like for the simple driving assessment. This assessment will be marked during your week 12 lab time, and worth 20% of your final course mark.

3.1 Path Generation - 2%

Teams will take an image of the maze and use it to generate a series of commands that solve the maze. This series of commands can be copied to the robot's source code and uploaded. Teams may choose to use the command format laid out previously or a custom solution. The initial and goal locations will be provided to you in the format outlined in 1.2.

2 marks will be awarded for showing a demonstrator your code executes. You must show the image input and the output to verify your solution is not hard coded.

3.2 Maze Completition - 8%

Maze Completion %	Mark
[0%-20%)	0
[20%-40%)	1
[40%-60%)	2
[60%-80%]	3
[80%-99%]	4
[100%]	5

Table 1: Marks awarded for maze completion.

Speed	Mark
Top 50% of teams which completed the maze	1
Top 25% of teams which completed the maze	2
Top 10% of teams which completed the maze	3

Table 2: Marks awarded for competition speed.

To be eligible for the speed marks you must complete the maze. Marks for maze competition progress will be at the discretion of the Marker. Each team will get a total of two runs. The first minor collision (hit but continues without assistance) will be allowed. The second minor collision or the first major collision (requires human intervention) will cause the run to terminate.

3.3 Continous Planning - 5%

For this task, a 3*5 cell section of the maze will be replaced by an obstacle course that the robot must navigate (Figure. 4). All the obstacles will be circular with a diamter of 100mm, however, they will be placed randomly within the designated area. Computer vision should be used to generate an occupancy map from the image and solve the path. You may hardcode the position of cells in which the obstacle course is in. The exact formatting used to transfer commands from the computer vision code to the robot is up to you. It may be best to generate a list of waypoints relative to each other which the robot can move to in series. There will only ever be one entrance and exit to the obstacle course. You are free to hard-code this knowledge into your generation algorithm.

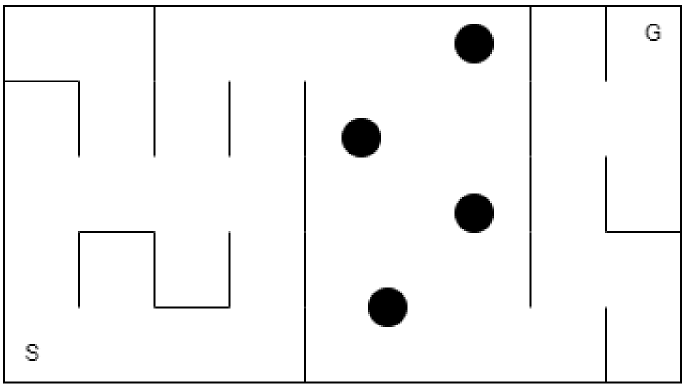

Figure 4: Example of a continuous maze your team may be presented with

1 mark will be awarded for generating an occupancy map and solving it. To receive this mark you must show an image with the trajectory to be taken by the robot to the demonstrator. 2 marks will be awarded for showing your robot can complete the obstacle course by following the trajectory shown in your occupancy map image. For this your robot may start at the first cell in the obstacle course and finish in the last cell of the obstacle course. 2 marks will be awarded if the robot starts from the start of the maze and makes it to the end of the maze moving through the obstacle course autonomously.

Each team will get a total of two runs. The first minor collision (hit but continues without assistance) will be allowed. The second minor collision or the first major collision (requires human intervention) will cause the run to terminate. Partial marks will be awarded at the discretion of the marker based on progress through the obstacle course if the run fails.

3.4 Autonomous Mapping - 5%

This is the hardest task and is meant to challenge teams who attempt it. For this task, teams will use the robot to autonomously search and generate the map of the standard maze. It will then navigate to the start position, solve the maze, and complete the shortest path run of the maze. During the mapping of the maze, a visualization of the map must be shown on the screen with a % completion score, derived from the number of cells the robot has visited.

The only information allowed to be hard-coded into your solution is the start coordinates (including orientation) and the goal coordinates. You must prove to your demonstrator that your solution is not hard coded to get marks for this task. Marks for maze completion progress will be at the discretion of the marker. Each team will get a total of two runs. The first minor collision (hit but continues without assistance) will be allowed. The second minor collision or the first major collision (requires human intervention) will cause the run to terminate.

Maze Completion %	Mark
[0%-20%)	0
[20%-40%)	1
[40%-60%)	2
[60%-80%)	3
[80%-100%)	4

Table 3: Maze Completion Marks

1 mark will then be awarded if the robot solves the maze in the shortest path after mapping it. It is possible to explore and solve the shortest path maze without fully mapping the maze. In this case, teams will be awarded the full 5 marks.

NOTE: Attempting and succeding in this task does not award full marks for the computer vision path planning task.

3.5 Submission

Marking will occur during your Week 12 Lab. Teams will be called to the front to be marked. It is expected that you turn up to the lab ready to get marked. You may compile and upload code between trials, however no significant changes will be allowed. A significant change is one requiring more than 5 minutes to implement. At minimum one team member must be present, If no team members are available please contact the course staff. A team evaluation task will be available on moodle after the competition of this task.

Code used for the micromouse Challenge must be pushed to your team's GitHub repository in a branch called "Micromouse".

4 Plagiarism

If you are unclear about the definition of plagiarism, please refer to What is Plagiarism? — UNSW Current Students. You could get zero marks for the assignment if you were found:

- Knowingly providing your work to anyone and it was subsequently submitted (by anyone), or
- Copying or submitting any other person's work, including code from previous students of this course (except general public open-source libraries/code). Please cite the source if you refer to open-source code.

You will be notified and allowed to justify your case before such a penalty is applied.

4.1 Generative AI

Code written with a generative AI (ChatGPT) is allowed but must be clearly labeled with in-line comments and mentioned in the file header.