Spring 2023

CDA 4621: Control of Mobile Robots
CDA 6626: Autonomous Mobile Robots

Micromouse Competition

The following sections provide a general description of the Micromouse Competition.

A. Competition Description

Micromouse is an international competition that started in the late 1970s[1,2]. The objective of the competition is for an autonomous robot (mouse) to learn a maze configuration through multiple navigation trials in order find the shortest path from start to goal. The maze is made up of 16×16 grid cells each, as shown in Figure 1, (left) physical robots, and (right) and simulated robots. The As it navigates, the robot needs to keep track of where it is, discover walls as it explores and map out the maze, and detect when it has reached the goal. Having reached the goal, the robot will typically perform additional searches of the maze until it has found an optimal route from start to the finish. Once finished, the robot will run that route in the shortest possible time.

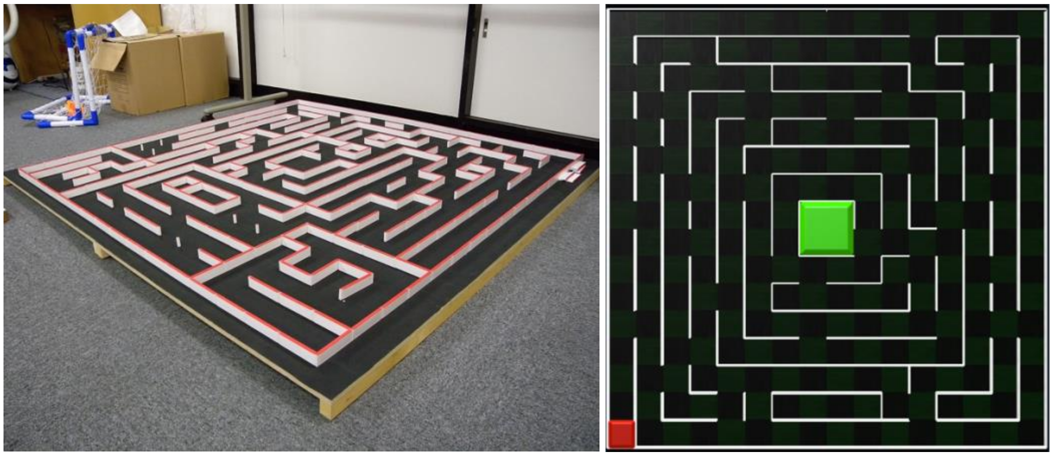

Figure 1. Sample competition maze: (a) physical maze, and (b) simulated maze (start location in red and goal location in green).

B. Tasks

The competition requires two separate tasks: (a) map the maze, and (b) reach the goal in the shortest time.

C. Rules

1. Rules for Each Task
    a. The contestants are assigned the (i) mapping algorithm task, and (ii) the fastest run task. Participants need to submit the two python controller scripts and any additional files needed, such as a pickle file of data structures or a python library containing custom functions to move the robot or update the data structure.

[1] https://en.wikipedia.org/wiki/Micromouse
[2] http://micromouseusa.com/

i. MappingController.py
   1. This script should navigate the maze and build a data structure that represents the maze.
   2. This script can be run up to 10 times, however, each participant only has 10 minutes of simulated time for mapping. Note that if the script is reset then the timer stops until the mapping script resumes.
   3. This script should save the maze data structure using some known method (such as pickle) so that the “FastestRun.py” script will be able to utilize the map created during mapping.
   4. You should assume that the starting location of the robot is known in advance, however, no additional information should be added in between mapping runs.

ii. FastestRun.py
   1. This script should use a data structure that represents the maze and find the fastest path from the starting location to the goal location. Participants can use any searching algorithm they wish.
   2. The robot can only start from the predefined staring locations, which will always be in one of the four corners.
   3. To be considered as successful, the robot must start in the staring location and stop once a goal location is reached. If the robot does not stop at the goal the clock does not stop.
   4. A run time is considered from when the robot starts navigation until the robot stops at the goal location.
   5. This script can be run up to 10 times, however, each student only has 10 minutes of simulated time for fastest run trials. Note that if the script is reset then the timer stops until the script resumes.

2. Rules for Micromouse
   a. A Mictromouse is not allowed to jump over, fly over, climb, cut, or mark the walls of the maze.
   b. Any violation of these rules will result in immediate disqualification.
   c. A Micromouse should compute the time from its start to its end on its own in both the mapping algorithm task and the fastest run task and print it to the console once the algorithm is finished. This will be compared to the simulator time.
   d. If there is too much difference between calculated time and simulator time run will be disqualified.
   e. A Micromouse has two wheels. Each wheel has a diameter of 1.6 inches (40.64mm), and an axis length is 2.28 inches (57.912mm)
   f. The Micromouse provided includes:
      i. 4 x distance sensors positioned in the front, right, rear, and left of the robot
      ii. 1 x lidar sensor with a 360 degree field of view and a resolution of 1 layer by 360 readings
      iii. 1 x camera with color recognition enabled facing the front of the robot

3. Rules for Maze
   a. The maze is composed of multiples of 18 cm x 18 cm unit cells. The entire maze is composed of

16 x 16 unit cells.

b. The walls of the maze are 5 cm high and 1.2 cm thick. The maze is enclosed by the outside walls.
c. The sides of the walls are white.
d. The start of the maze is located at one of the four corners. The start cell is bounded on three sides by walls. The destination goal is the four cells at the center of the maze. The goal area has only one entrance.
e. Each cell has at least one wall at each corner of a cell.
f. Multiple paths to the destination cell are guaranteed. The destination will be positioned so that a wall-following mouse will not be able to find it.

4. The maze where the Micromouse will run is only seen at evaluation time.
5. Must use e-puck provided. Robot may only be modified by including additional sensors at cost of 10 points per sensor.

C. GitHub Server

GitHub server containing all the project files is: https://github.com/biorobaw/CDA4621_MicroMouse

D. Evaluation Criteria

Scoring Criteria: Evaluate the Fastest Score.

1. Mapping algorithm task: max 60% score (less than 10 minutes)

The following equation will be used for the mapping algorithm task.

$$y = \frac{90c}{256}\sqrt{-\frac{(t-600)}{600}} + 10(\frac{c}{256})$$

y: score weight
t: how long the robot takes to finish each task
c: how many cells the robot maps
Mapping algorithm task score = 0.6 × y - (additional sensor deductions)

2. Fastest run task: max 40% (less than 10 minutes)

The following equation will be used for the fastest run task.

$$z = -\frac{1}{6}t + 100$$

z: score weight
t: how long the robot takes to finish each task
Fastest run task score = 0.4 × z - (additional sensor deductions)

F. Competition Schedule

Competition will take place during finals, May 2, 12:30-2:30pm.