Princeton University
Virtual Micromouse Competition

Rules Last Updated: June 20, 2020.

Registration deadline: TBD

Program submission deadline: TBD

Competition date: TBD, in October or November 2020

The following rules are for the Princeton University Robotics Club’s first Micromouse Competition. The rules are based on the Brown IEEE Robotics Competition rules and the 2019 Princeton Micromouse Competition, with major adaptations for the virtual format.

Objective

Individuals or teams of students will develop an algorithm for navigating and solving a maze, which will be simulated in a computer program. Teams will be scored based on how fast the simulated robot can solve the maze, according to the Scoring mechanism outlined below.

Competition Eligibility

1. Teams consist of at least 1 and at maximum 9 members.
2. All team members must be enrolled in an undergraduate program at the time of the Princeton University Virtual Micromouse Competition. Students who graduate up to nine months before the competition date are also eligible to participate in the competition.
3. If a team wishes to register after the Registration Deadline, they may contact the Princeton University Robotics Club leadership via email (see “Further Information”). Late registrations are not guaranteed.
4. Multiple teams from the same college are allowed to register for the competition, and teams may comprise members from more than one college.

5. Participants cannot join more than one team.

**Rules for the Mazes**

Each maze of the competition will meet these criteria.

1. The maze consists of a 16x16 grid of tiles.
2. Posts are located at the corners of the tiles.
3. Each post has at least 1 wall attached to it, except for the center post which attaches to no walls. A wall spans 2 adjacent posts.
4. The start tile is the bottom-left tile on the screen. It is bounded on three sides by walls, two of which are the bottom and left sides.
5. The robot in the simulation begins facing up, toward the top of the screen.
6. The goal consists of the four center tiles. There is only one entrance to the goal.
7. The maze may have several solutions, but the maze will not be solvable by a wall-hugging algorithm.

Sample mazes will be provided. Competitors may also develop their own mazes for testing purposes. The competition organizers will not release the competition mazes until after the competition has passed. However, the competition organizers will release basic information about the competition mazes, such as the approximate number of walls, so that competitors may test their programs on mazes with similar parameters.

**Rules for the Algorithm**

1. Each team submits one maze-solving computer program. The program communicates with the simulator via stdin and stdout. The interface is detailed in the readme (which will be provided by August 1st).
2. The program may be written in C++ or Python 3. If a team uses Python with libraries that need to be installed, the team must install those libraries using a Pipenv virtual environment and submit the Pipfile specifying the libraries.
3. Programs may span multiple files.
4. The submitted code shall not exceed 2 megabytes.
5. Programs cannot communicate with the internet.
6. The program should terminate itself at the completion of a maze-solving attempt.
7. Any egregious code is grounds for disqualification at the competition organizers’
discretion.
8. Teams must submit their program by the submission deadline. The competition organizers will not share any team's code externally.
9. Here is a brief overview of the interface. More details will be provided when the simulator is released:
   a. The robot has short range sensors and can only detect the absence or presence of a wall immediately to its left, right, or front. It cannot detect walls that are farther away.
   b. The robot cannot move diagonally. It can only make 90 degree turns.
   c. All movements are executed perfectly by the simulator, so there is no need to worry about alignment or state estimation.

Scoring

Each team’s algorithm will be run on three mazes, and the score will be the sum of the final scores on each maze. A lower score is better, and the team with the lowest score will be declared the winner. The score will be based both on the fastest start-to-finish run and the overall time in the maze, so the winning team will seek a balance of finding the fastest path without exploring excessively. Scoring is based on a few basic principles of physical micromouse competitions. Turning and driving consume time, and a robot can go faster if it drives straight for a longer distance. Micromouse robots also have a time limit. The score will be computed as follows:

1. Moving one tile forward adds 1 point to the total effective distance.
2. A 90-degree turn adds 1 point to the total turns.
3. If a robot moves forward more than 2 tiles at once, each tile after the second tile will add only half a point to the total effective distance. This simulates a robot driving faster during an optimized run if it finds a long straight path.
4. A run begins when the robot leaves the starting tile and ends when it reaches the goal.
5. The robot may reset its position to the starting tile by issuing a reset command, but a 15-point penalty will be added to the next run's effective distance. There is no penalty if the robot navigates itself back to the start tile.
6. A crash occurs if the robot attempts to move forward but is obstructed by a wall. The simulator will tell the robot that it has crashed, and any further movement commands will be ignored until the robot resets to the starting tile (and incurs the reset penalty). A command that causes a crash will not move the robot in the
simulation and will not affect the total distance.

7. A robot may make several runs. Its **best run** is the run with the smallest sum of turns plus effective distance.

8. **Final score** = best run turns + best run effective distance + 0.1 * (total turns + total effective distance)

9. If the goal is never reached, the final score will be 2000.

10. A maze-solving attempt will end when the program terminates or when the sum of total effective distance plus total turns exceeds 2000, whichever comes first. This imposes a time limit; however, the highest-scoring teams will probably consume much less time.

**Resources**

The simulator will be released by August 1st and will be compatible with major operating systems. It will visualize a micromouse robot in a maze and display all relevant information on the robot’s score. Sample programs will be released, demonstrating how to interface with the simulator. Participants may seek external resources for algorithms and code as outlined in the **Fairness** section.

**Fairness**

Teams are expected to exercise common sense when it comes to fairness. In general, a team may not seek an advantage that is not available to all teams. The following guidelines apply.

1. As a starting point, teams may use code that is publicly available online and free to use, such as code found in public GitHub repositories. Proper attribution is required, which can take the form of a comment in the code or a separate file listing sources. If teams use code they find online, it is expected that they make substantial modifications to achieve a more optimal maze-solving algorithm that suits the specifics of this competition.

2. Teams may have advisors or mentors who provide high-level guidance to the team. However, these advisors or mentors may not provide code, or detailed pseudocode, unless it is publicly available online. For example, advisors may describe the flood-fill algorithm and point members towards articles and public example code, but advisors cannot provide their own private example code for the algorithm.
3. Teams may not share any algorithm details or code with other teams.

Further Information

Competition Website:
https://robotics.princeton.edu/2020-virtual-micromouse-competition

Run and Hosted By: The Princeton University Robotics Club

If you have any further questions or concerns, please reach out to the Princeton University Robotics Club at robot@princeton.edu.