Lab 5: Maze Challenge

January 6th, 2020

Introduction

Bonus points In this lab, you can earn up to **6 bonus points**. You have time to solve the maze challenge until Monday, January 27th. On this day, you will have the opportunity to present your solution. Bonus points will not only be granted for speed and for successfully navigating through the maze; but you can also earn points if your robot fulfills basic requirements, i.e., driving, turning, and sensing, reliably. We will also ask you to explain the behavior of your robot orally and show the software architecture with a UML diagram.

In summary, the distribution of bonus points is as follows:

- basic functionality: **up to 3 points**
- solving the challenge within the time limit: **up to 1 point**
- explanations, code quality, and UML diagram: **up to 2 points**

Please note that both team members have to contribute equally to the explanation of the behavior. We reserve the right to adjust the points of a member according to the level of program knowledge. Additionally, all points given during the challenge event are subject to a plagiarism check afterwards. Again, we will use your solution uploaded to your Git repository *before* the final challenge.

Award & Ceremony The three best groups get the chance to present their solution to the audience and compete again on a different course for the trophy. We will hold the final competition and award ceremony after the exam Q&A session on January 31st (a detailed schedule will be announced mid January). The winner will be decided by a popularity voting of all participants, and the trophy will be engraved with your names and displayed permanently at our institute.

Extra working hours During the usual lab hours, we will provide a maze for testing your solutions. In addition to the lab times, each group may borrow a robot for a whole day once a week. We offer additional hours between January 7th and 24th, 2020; each Tuesday, Wednesday, and Thursday; always from 9.00 – 16.00 h. These hours are intended to give you room for self-reliant work with the robots. Thus, we will not be around to provide support for you during these days. If you encounter serious technical issues, however, don't hesitate to approach us immediately. For other problems, please ask and chat with your surrounding students, or ask your questions during the official lab times.

We can offer 20 robots per day; meaning that every group is able to work once a week. However, registration of a specific day is 'first-come-first-serve'. To enable a smooth experience for all students, we have to apply certain rules:

- Register in beforehand via StudIP (will be opened shortly, announced via StudIP).
- Robots can be picked up in Building Q, Room 1.006.
- Each team is allowed to register *once per week*.
- Robots have to be picked up between 9:00 and 9:30 h and returned between 15:30 and 16:00 h.

- Stay in building Q with the robots.
- Check the health of batteries and mention any malfunction directly when returning.

You can also download a loan form from StudIP, in which you confirm that you have received a robot with accessories. Please print and fill this form beforehand and bring it when you pick up your robot.

We reserve the right to exclude groups from extra working hours, if they violate the rules mentioned above. We are working on providing you a dedicated room for testing, but we cannot guarantee this at the moment. In the ground floor of building Q, you find several learning rooms which may be available.

The Challenge

In the final challenge, we will place your robot in an undisclosed maze. The goal is to find a target wall with a specified color within the maze. When your program starts, it should show a menu in which the target color can be selected. Once the user selects a color and presses enter, the robot should begin to search the maze for the correspondingly colored wall. The moment the robot starts moving, it should produce a beep, and once the robot has found the target wall, it should produce another beep. We will measure the time between the beeps to enforce the time limit (5 minutes, tentative).

Since there may be more than one colored wall, you must make sure that you detect the right color. If the robot beeps in front of any other wall, this will result in a penalty. It is sufficient to use the color IDs that the color sensor of the robots provides.

Figure 1 shows an example of a maze. The final maze will not be the same, but you can make the following assumptions.

- It will be built in a grid system, where each tile is 35 cm $\times$ 35 cm in dimension.
- It consists of 3×4 tiles.
- It will be closed, so that the robot cannot leave the maze at any location.
- The maze does not contain loops.
- The target wall will be (one of the three walls) in a dead end.
- There may be other colored walls in the maze, but only one that has the specified color (of the target wall).
- The start position is not revealed before the final date.

Hints During the previous lab sessions, you have used and implemented the basic functionality of the robot, that you can reuse to solve the final challenge. However, to ensure that you can explore the whole maze and don't miss any branches, you have to think of a suitable strategy.

You may have noticed that sometimes, when your robot starts to move forward or stops, one wheel increases its speed faster than the other. That leads to an undesired change in orientation. To mitigate the effects, you can synchronize the wheel movements. Have a look at the `startSynchronization()` and `stopSynchronization()` methods of the `EV3LargeRegulatedMotor` class. To further improve, you can also use the gyro sensor while driving straight to keep the desired orientation.

Maze Simulator

We want to enable you to work on and improve your solutions more flexibly. Therefore, we have developed a simulator which you can run on your computer. The simulation provides a graphical 2D visualization of the robot's movement, the LCD output, and allows you to simulate button presses.

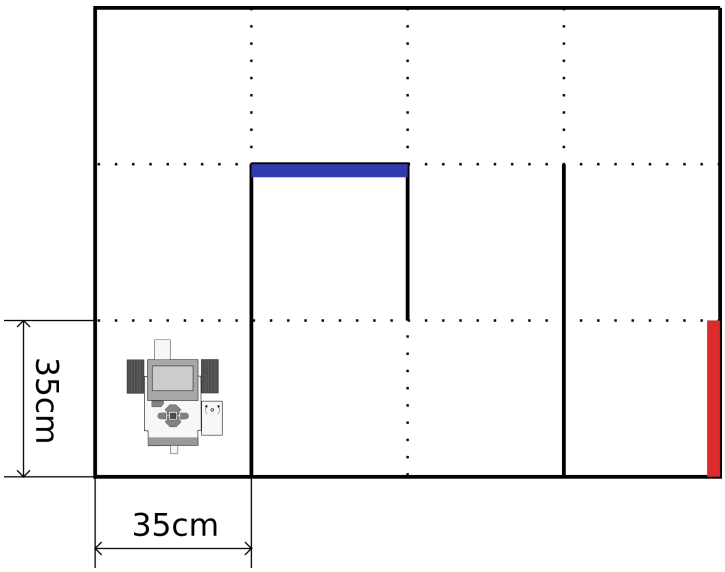
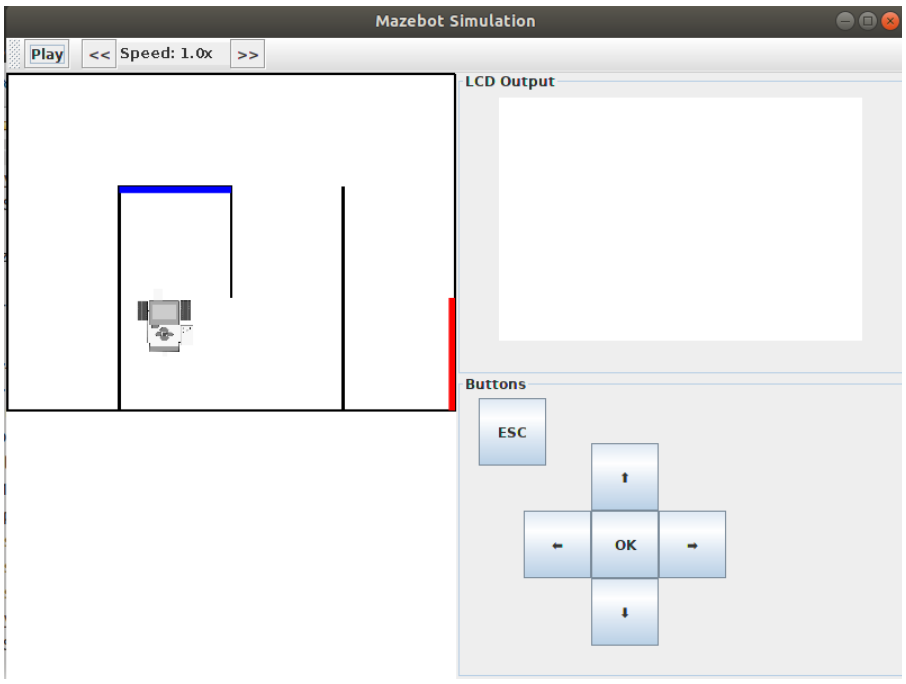


Figure 1: The final maze will have the same size and number of tiles, but the wall configuration and starting location will be different.

Keep in mind that the simulation does not replace testing on the real robot, as it cannot accurately represent the behavior of real hardware and physical effects such as inertia and friction. However, it can reveal logical problems in your maze exploring programs and allows you to test your solutions faster.



This year is the first time this simulator is released. In case you encounter any problems or find bugs, do not hesitate to contact us. To do so, submit an issue to our issue tracker in Git. We will try to fix them timely.

You can find the latest release and the installation and usage instructions in the `wiki`.