CG1111 Engineering Principles and Practice I

The A-maze-ing Race Project 2019

Welcome to the grand project of CG1111: The A-maze-ing Race!

In this race, your mBot needs to find its way through a maze in the shortest time. Similar to its namesake TV program, your mBot will be facing a number of challenges at intermediate waypoints while attempting to complete the race. To successfully meet all the requirements, you need to have a good grasp of many of the principles you have learnt in CG1111 and apply them into good practice!

The following are the key requirements:

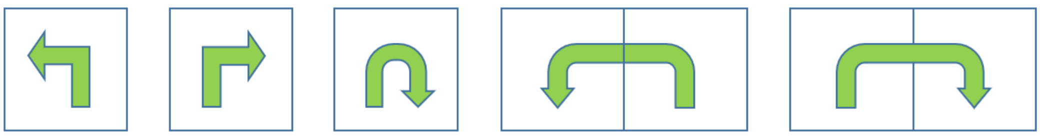
1. The mBot must not bump into any wall. Your mBot shall rely on a front ultrasonic sensor, and two side IR proximity sensors to accomplish this. You need to come up with your own algorithms to meet this requirement. Note that there will be penalty points for bumping into walls, even if your mBot doesn't get stuck.

2. All turns in the maze are dictated by the waypoint challenges. Your mBot must not make any automatic turn without decoding a waypoint challenge.

3. At each waypoint challenge, there will be a black strip (about 4 cm by 21 cm) on the maze floor. Your mBot needs to detect the black strip, stop, solve the waypoint challenge directly above it, and act according to the turn instruction decoded from the waypoint challenge.

4. There are two types of challenges that dictate the turns at the waypoints:

   (i) Colour-sensing Challenge:
   Your mBot already has two RGB LEDs and one LDR built onto its mCore. You need to use them to implement colour-sensing. (Note that you need to remove the top cover of your mCore to accomplish this.) Depending on the colour of the paper that your mBot senses above it, it needs to execute one of the following five types of turns:



Left-turn	Right-turn	180° turn within the same grid	Two successive left-turns in two grids	Two successive right-turns in two grids

   <u>Note:</u>
   For the "two successive left-turns in two grids" and the "two successive right-turns in two grids", there will not be any black strip in the second grid to guide the mBot about where it needs to execute the second turn. Your mBot could rely on another sensor (e.g., ultrasonic sensor) to decide where to execute the second turn.

Table I summarizes how the colours are to be interpreted. Setups with colour samples for each type of turn will be provided in the lab. Note that the colour paper will be suspended at a height of about 14 cm from the maze floor. Hence, your mBot's maximum height (including all the wires and other components) must not be taller than **14 cm**.

Table I: Colour interpretation for the Colour-sensing Challenge

Colour	**Interpretation**
Red	Left-turn
Green	Right turn
Yellow	180° turn within the same grid
Purple	Two successive left-turns in two grids
Light Blue	Two successive right-turns in two grids

(ii) Sound-based Challenge:

The board above the sound-based challenge will be **black** in colour. There is a black audio speaker suspended above the mBot that plays either one of two types of audio:

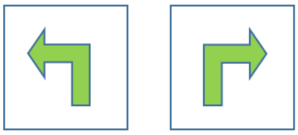
Type 1: The audio consists mainly of low frequency sounds, where most of the sound frequencies are between 100 and 300 Hz.

Type 2: The audio consists mainly of high frequency sounds, where most of the sound frequencies are 3000 Hz and above.

Depending on the type of audio detected, the mBot needs to execute one of the two types of turns, according to the interpretations stated in Table II. Setups with the sound samples for each type of turn will also be provided in the lab.

Table II: Audio Interpretation for the Sound-based Challenge

Type of Audio	**Interpretation**
1	Left-turn
2	Right turn



Left-turn Right-turn

5. End of Maze:

At the end of the maze, there will also be a black line. The colour of the paper above the mBot at this grid will be **black**. Upon decoding that no turn action needs to be taken for both colour-sensing challenge and sound-based challenge, the mBot shall interpret this as the end of the conquest**, stop moving**, and **play a celebratory tune** of your choice (Yay!).

Final Project Evaluation:

The race will be conducted during your **Week 13 Studio 2** timeslot:

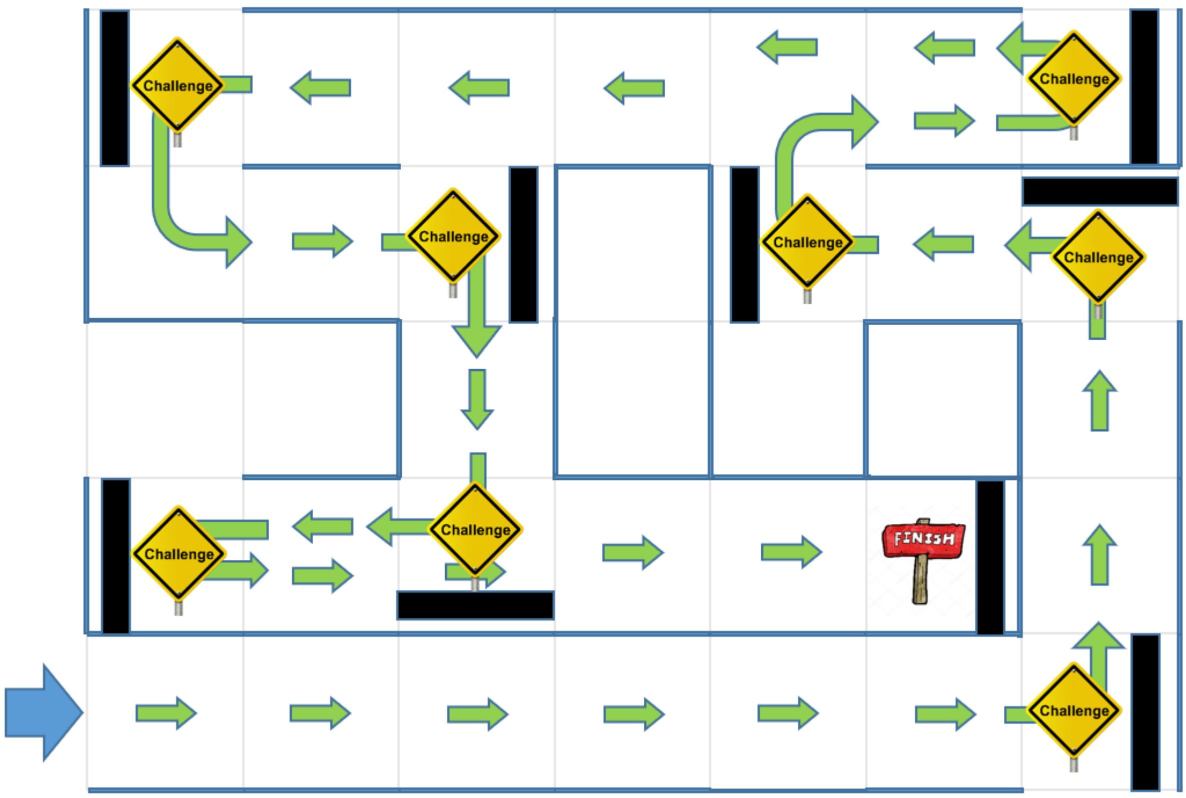
Studio Group 1: 13 Nov (Wed), 9am-12pm
Studio Group 2: 13 Nov (Wed), 2pm-5pm
Studio Group 3: 14 Nov (Thu), 9am-12pm
Studio Group 4: 14 Nov (Thu), 2pm-5pm

Rules:

1. At each challenge, if your mBot turns in the wrong direction, it will be teleported to one grid before the challenge to make a second attempt while the clock continues to run. If it fails in its second attempt, it will be manually turned to the correct direction, and an additional penalty time of <u>30 seconds</u> will be added.

2. You are not allowed to add any commercial-off-the-shelf sensors that are not issued by the DSA Lab.

3. The actual maze layout for the A-maze-ing race will not be revealed beforehand. The figure below shows a **sample** maze layout.



4. You are not allowed to perform any last-minute calibration before the start of your team's turn during the project evaluation. Hence, all calibrations must have been completed using the sample waypoint challenge setups that we have provided in the lab, before you come for the project evaluation.

5. The group that completes the maze in the shortest time wins a delicious prize (through the courtesy of your instructors)!

Grading Criteria:

Criterion	Marks
Meeting the required features	15
Algorithms and coding (e.g., elegance of algorithms, well commented codes, etc.)	5
Short group report	5
Total	**25**

Deliverables (Report and Source Codes):

Deadline: **18 Nov 2019 (Monday), 2359 hrs**

(10% will be deducted for every day it is late)

Zip all files into a single file (one submission per team), and name it according to your Studio Group Number, Section Number, and Team Number, e.g., **Group1_Section1_Team1.zip**, and upload it into the "Project Submission" folder in LumiNUS before the deadline. You must include the following:

1. Program source codes. The codes must be well documented by providing appropriate comments.

2. A concise written report, in PDF format. You may use your own discretion for the number of pages. Your report must describe your design in detail, especially about how the required features are met. Please include:
   - A cover page with your Studio Group Number, Section Number, and Team Number, along with the team members' names.
   - Description of the overall algorithm your mBot uses to solve the maze. You may include pseudocode/flowcharts or other pictorial aids to assist in your explanation.
   - Implementation details of the various subsystems – algorithms for keeping mBot straight, audio processing circuit design and algorithms, colour sensing algorithms, end of maze detection algorithm, etc.
   - Steps taken for calibration.
   - Details about work division within your team – each member's role in the project.
   - Any significant difficulties and the steps taken to overcome them.

Peer Evaluation:

You will submit a confidential peer evaluation about your teammates' contribution in the project. More details will be announced at a later date. If there is any teammate who is not contributing to the project, **please inform us as early as possible**.

Academic Integrity:

Plagiarism will not be tolerated. You are **not** supposed to share any code with other teams. You **may** discuss the project requirements or your solution strategies at a high-level, without sharing details at the code level. We do **not** distinguish between those who copy others' work, and those who allow their work to be copied. If you are involved in plagiarism, you will be given 0 mark for the project, and referred to the University for disciplinary action.

Resources and Tips for your CG1111 project:

1. You can download the latest Makeblock libraries from the following link:
   https://github.com/Makeblock-official/Makeblock-Libraries
   - Follow the instructions on the webpage to add the libraries into your Arduino IDE on your laptop.
   - You can find the source code for the mBot's factory firmware by navigating to /examples/Firmware_For_mBlock/mbot_factory_firmware
   - You can learn a lot about how the mBot works by going through the factory firmware.
   - In addition, there are many code examples inside the subfolder /examples/, from which you can learn how to work with some of the components on your mBot that you would need for your project (e.g., LDR, line sensor, DC motors, speaker, etc.).

2. The schematic for your mCore (i.e., mBot's brain) can be downloaded from this link:
   https://github.com/Makeblock-official/mBot_Firmata/blob/master/hardware/mCore.pdf
   It serves as very good reference when trying to understand how the firmware works.

3. The datasheet for your electret microphone can be downloaded from the "Project Resources" folder in LumiNUS.

4. When mounting your breadboard with the audio processing circuits on top of your mBot, ensure that it does not cover the RGB LEDs and the LDR on your mCore. Otherwise it may affect your mBot's performance in the colour-sensing challenge.

5. You need to ensure that your batteries are properly charged. If their voltage gets too low, your mCore's 5V line may be less than 5 V. This could affect your calibrations.

6. To tackle the sound-based challenge, you need to build two active audio filters with peak detectors. Since your mBot only has 0 V and 5V, you will power your op-amp using just the

positive power rail and 0 V. You may also want (optional) to include a variable resistor in each of the two audio filter circuits, so that you can tune its gain.

7. You may also find the free LTSpice circuit simulation software useful for testing your design for the active audio filters:
   https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html
   It is quite easy to learn. Simulate your audio filter designs and verify their frequency response. You can model your electret microphone as a small AC voltage source with a series output impedance of 2.2 kΩ (note that this is the impedance inside your microphone, according to the datasheet given).