CmpE 434 Introduction to Robotics Term Project: NBA EV3

Deadline: January 9, 2018

1 Introduction

In this project, a basketball player robot is expected to show its skills to everyone. The eternal rival of our basket player robot kidnapped our robot and put it to in a prison. The only way to escape from the prison is winning against its rival in a basketball match. In the match, our robot will be teleported into the maze and scoring two baskets will be sufficient for winning. However, there are two different colored balls and baskets, so our robot should score with the correct ball also the rival put some traps in the maze and our robot should avoid these traps while wining against its rival.

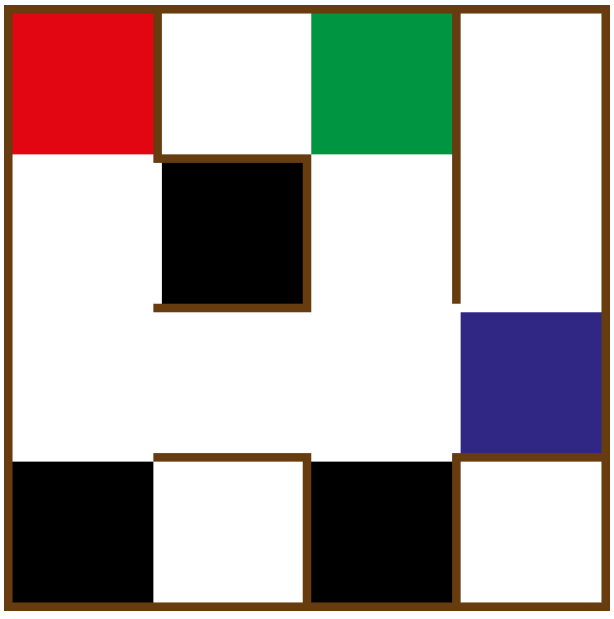

Figure 1: The environment specification of the project

In order to escape from the maze and scoring, you will need to implement a mapping and localization algorithm in a 132 cm $\times$ 132 cm grid maze given in Figure **??**. The grid cell sizes will be 33cm $\times$ 33cm. The red and blue cells are the location of the baskets and black cell is the location

of the traps and the green cell will be the location of the ball.

The project has two stages:

- **The Mapping Stage:** The basket player robot should discover the environment before the match begins, in order to find the location of the baskets, balls and the traps. Therefore, the robot will initially create the map of the environment and store the map.
- **The Task Execution Stage:** The basket player robot will be teleported into the maze from its prison, but teleportation is random. The robot can end up in any cell. The robot should take the ball and score a basket with it.

The project consists of the following parts:

1. Localization
2. Path Planning
3. Mapping
4. Grasping Ability

2 Scenario

In the first stage of the scenario, the robot will be placed in a grid cell which is randomly selected. The robot should travel across the maze to find the important locations. Also, the robot should show the map of the environment with the important locations on the PC screen. When the robot wants to announce the completion of the mapping of the maze, the robot should make *3 beeping sounds*.

In the second stage of the scenario, the robot again will be placed in a grid cell which is randomly selected. While the robot travels, the possible location(s) of the robot should be displayed on the PC screen. The ball will not be placed in the first stage of scenario, but in the second stage, the ball will be placed on the **green** cell. The robot should grasp the ball and it should score to correct colored basket.

After scoring, the robot should find the **green** cell and end the game by making *3 beeping sounds*, and thus the robot will complete the scenario.

3 Specifications

The environment specifications are as follows:

- The size of the map shall be 132 cm $\times$ 132 cm and shall be constructed using cardboard on level **B** of the department.
- You shall have a grid world of 4 $\times$ 4 of the floor grid cells. The size of the cells shall be 33 $\times$ 33 cm.
- The height of the walls shall be 33 cm.
- The robot shall be positioned at a random grid cell with a random orientation (one of the four main directions), but the robot shall be at the center of the cell.
- The ball shall be at the center of a cell.
- The *basket location*, the *ball location* and the *traps* shall be determined at the time of final demo.
- You shall implement a button interface where you assign a task for three of the buttons and one of the buttons shall switch to the idle state which we shall use to reset the execution. (See Section **??**)
- If the robot hits one of the obstacles or the wall and the obstacle moves more than 5 cm, the robot shall be stopped with the button interface and shall be positioned at the previous start position.
- The task completion of the robot shall be determined when the robot stops and plays a sound acknowledging the completion of the task.

Notes:

- Going across the trap cell in the first phase and entering trap cells in the second phase will fail the current phase and you will start that phase again. You can restart the phase at most two times.
- You should save the map of the environment in a file because restarting the second phase should not reset the map of the first phase.

4 Button Interface

- *UP* button shall run the *Mapping* task.
- *DOWN* button shall run the *Task Execution*.
- *ENTER* button shall start the *Idle* state.
- *ESCAPE* button shall *Reset* the current task.

5 Deliverables

You need to submit:

- Design document (20 percent) : December 12, 2017
- Preliminary demo (15 percent) : December 12,2017
- Final demo (45 percent): January 9, 2018
- Final report (20 percent): January 9, 2018