

CSCI 1010 Computer Science Orientation

Maze Navigation Strategy Report

Date: 25 October 2017

Team: Team 10

Names: Sam , Binrong Gus Gamble*, Melody

1 Problem Description

Create a LeJOS EV3 program for a wheeled robot that allows it to navigate through a maze from starting cell to goal cell, acknowledge reaching of goal cell, and return to the starting point without making any wrong turns.

Expected behavior:

- Robot moves forward from starting cell
- Robot keeps moving and follows black line
- Robot uses IR sensor to determine when to turn
- Robot uses touch sensor to get out of dead ends
- When robot reaches goal cell, wiggle
- Robot moves away from goal cell and correctly returns back to starting cell

2 High level Strategy

Case 1: when the line branches (more specifically, if the robot sees a left turn with the IR sensor)

- Always take the left turn

Case 2: when the robot touches a wall (in front)

- Because of case 1, we can assume that the robot will have to turn right 90 degrees and resume movement forward. If there is no touch, move to behavior 3

Case 3: Robot is off the line

- Find the line and get back on
 - Maybe we can do this by turning opposite of the last direction?
 - Goal for lab is to collaborate and work through the logic of this

Return to the starting point: use stack structure and switch statements

3 Detailed Strategy

3.1 Robot Capabilities

- Move robot forward
- Rotate robot at an angle
- Get touch sensor input to detect when robot hits wall
- Get infrared sensor input to detect distance from obstacle
- Get color sensor input to detect the black line, blue squares, and aluminum foil
- Register when a button on robot is pressed

3.2 Robot Components

- Large regulated motors
 - Left motor (connected to port D)
 - Right motor (connected to port A)
- Touch sensor (connected to port S2)
- Color sensor (connected to port S4)
- Infrared sensor (connected to port S3)

3.3 Strategy Pseudocode

```
while(Button.getButtons() != Button.ID_ESCAPE)
{
    moveForward();
    if(IRsamplevalue[0]>distanceFromWall)
        makeLeftTurn();
    if(Touchsamplevalue==1 && !(IRsamplevalue[0]>distanceFromWall))
    {
        backUp(someSmallDouble);
        makeRightTurn();
    }
    if(Touchsamplevalue==1)
    {
        backUp(someSmallDouble);
        makeLeftTurn();
    }
    if(robotIsOffLine()){
        getBackOnLine();
    }
    if (getColorID()==2){
        checkColor(); //with getRedMode()
        if(colorIsFoil){
            wiggle();
            returnToStart();
        }
    }
}
```

```
    }  
  }  
}
```

3.4 LeJOS implementation details

Differential pilot to perform movements with robot

ColorSensor class so that the sensor can be used in the program

getTouchMode to get instance of touch sensor

getDistanceMode to get instance of IR sensor

getColorID to get instance of color sensor in colorID mode

fetchSample to retrieve sensor values

Stack class

Tested color values with getColorID, NOT with getRedMode

Black: 7

Board: 13

Blue: 2

Foil: 2

- To tell the difference between foil and blue, use a method, that we write, called checkColor() that uses getRedMode() to give different readings