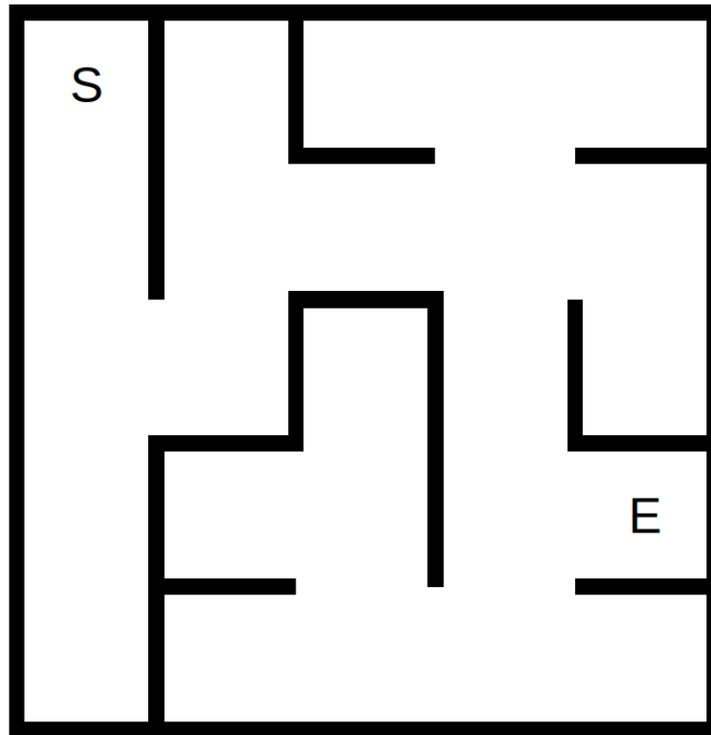

Aberystwyth Robotics Club - Elegoo Robot Maze Navigation

Introduction

This worksheet looks at how our obstacle avoidance can be applied to maze navigation and various adaptations that may need to be made.

Obstacle avoidance in a maze

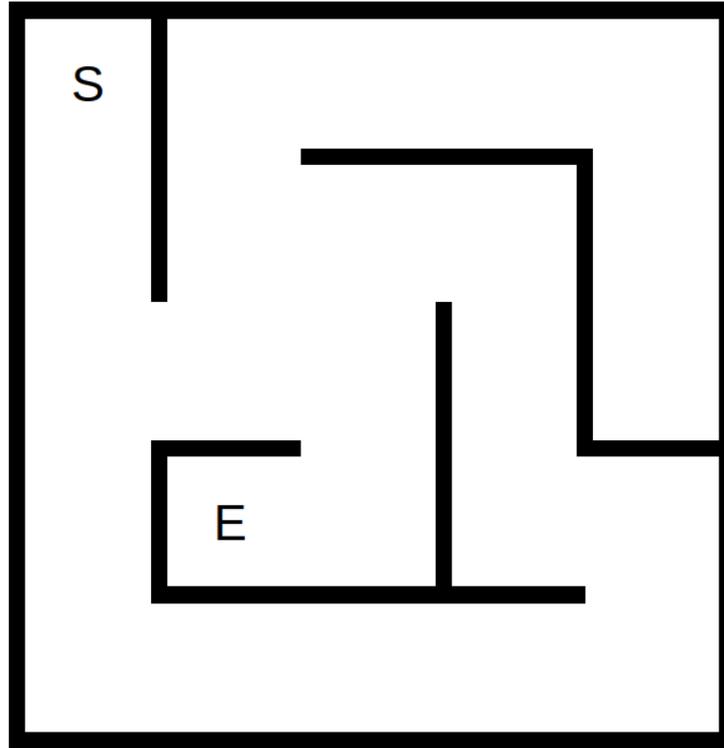
What would happen if our robot tried to navigate this maze with our obstacle avoidance?
S marks the start, and E marks the end.



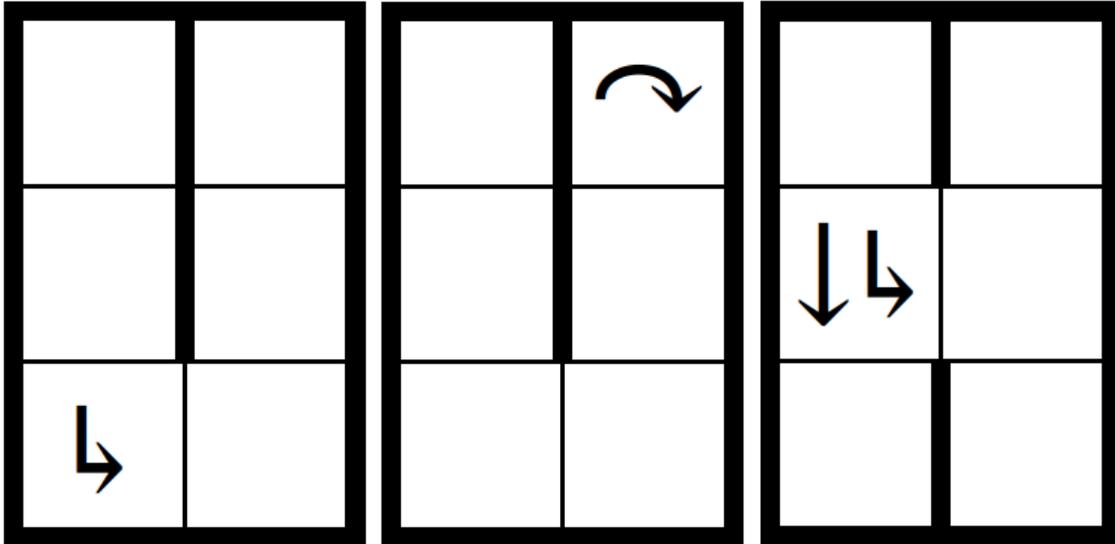
Discuss how you might make use of the ultrasonic sensor to navigate the maze. Some ideas and points to consider are given below.

1. You could follow the wall on your left. Set the servo to 0 or 180, and try to maintain a constant distance.

- What path would the robot take around the maze?
- What would happen if the maze had a loop in it?



2. Consider the maze as individual cells, move one cell at a time and decide which direction to move. Use the scenarios below to help think about the different conditions you might encounter.



- Measure how long it takes to move forward one cell
- Measure how long it takes to turn the robot 90° on the spot
- In each cell, use the servo to turn the ultrasonic sensor to detect walls and gaps
- Use the if-then-else statements to change the behaviour based on what is detected

Challenges

- Detect when you drive over a black line to mark the end of the maze and perform a celebratory dance at the finish line.
- Advanced: Can your robot learn its way through the maze?

You may find a data structure called an array useful here. This is a single variable that allows you to store a set of values, that can be referenced by an index number. You specify the length (L) of the array when you create it, then use indexes 0 to L-1 to access them:

```
int myArray[6] = {}; //create an array of length 6 (indexes 0-5)
myArray[0] = 3; //set the value of the first index to 3
myArray[5] = 12; //set the value of the last index to 5
for(int i=0; i<6; i++)
{
    Serial.println(myArray[i]); //cycle through the array printing out all
    the values
}
```

You could give a number to each type of movement or you could try to encode each cell in the maze, where position (x,y) in the maze (starting from (0,0)) is index $x*L+y$ in the array.