To: Dr. Andrew Morton
From: Leong Si
Date: October 3, 2014
Solving Mazes with Scribbler Bot

Project Description

The goal of the project will allow the scribbler bot to solve 2-dimensional mazes, whose passages are restricted by walls. The robot will use the "wall follower" method to solve the maze. The "wall follower" method picks either left or right side of the wall, and stays on the chosen side until the maze is solved [1]. Once the maze-solving program starts running, the robot will be able to find its way to the exit without users' help. The robot will detect any wall or intersection by using the IR sensor located at the front of the robot.

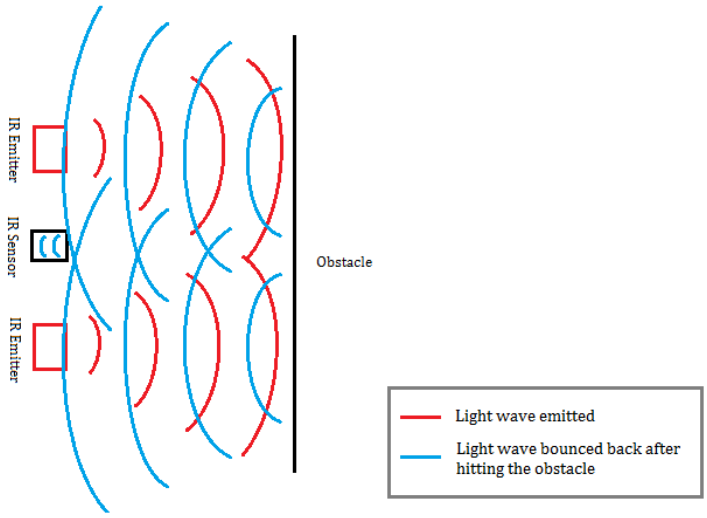
When the robot solves the maze successfully, the computer will be able to store the solution in a text file. The user can then use the solution to go through the maze again without trials. The solution will only work if the robot starts at the same position and direction.

Features

We will use Python with Myro module to program the maze-solving function of the robot. Myro is developed for programming robots such as the scribbler bot [2]. All the sensors and hardware components of the robot can be accessed by the functions in the Myro library.

The scribbler bot has multiple sensors and other hardware installed. However, we will only be using the IR sensors and the motor of the robot. The IR sensors are used to detect walls. Two IR emitters are placed on the left and right front of the robot. To detect obstacles, the emitters send out a light wave. If the wave hits an obstacle, it will bounce back to the IR receiver located between the two emitters (see Figure 1). The motor controls the two wheels of the robot, allowing the robot to explore mazes. The wheels can work independently, thereby allows the robot to rotate without turning the wheels.

We will also be using the IPRE Fluke, which allow the robot to be wireless accessible via Bluetooth. The IPRE Fluke adds 3 more obstacle sensors, a battery voltage sensor, and a camera for the robot to use [3]. However, we will not be using those hardware components in this project.

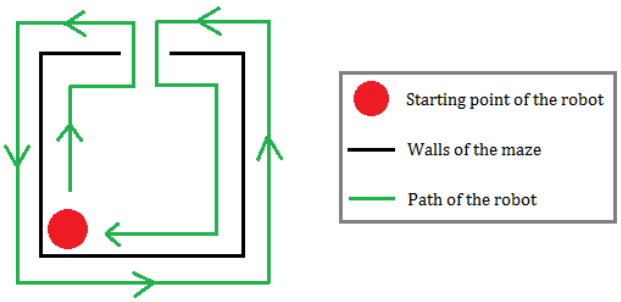


(**Figure 1**) Light wave hits an obstacle and bounces back to the IR sensor

Design Challenges

The main challenge we need to overcome in this project is to use the IR sensor correctly. Since the detection of obstacles relies on the light wave bounces back to the sensor, the sensor can only detect the signal if the obstacle is close to perpendicular to the robot. Also, we will need to know how far the obstacle is before the robot decides to change direction. If the distance is too great, the robot might hit another wall after turning. This would take many trials to find the optimal distance.

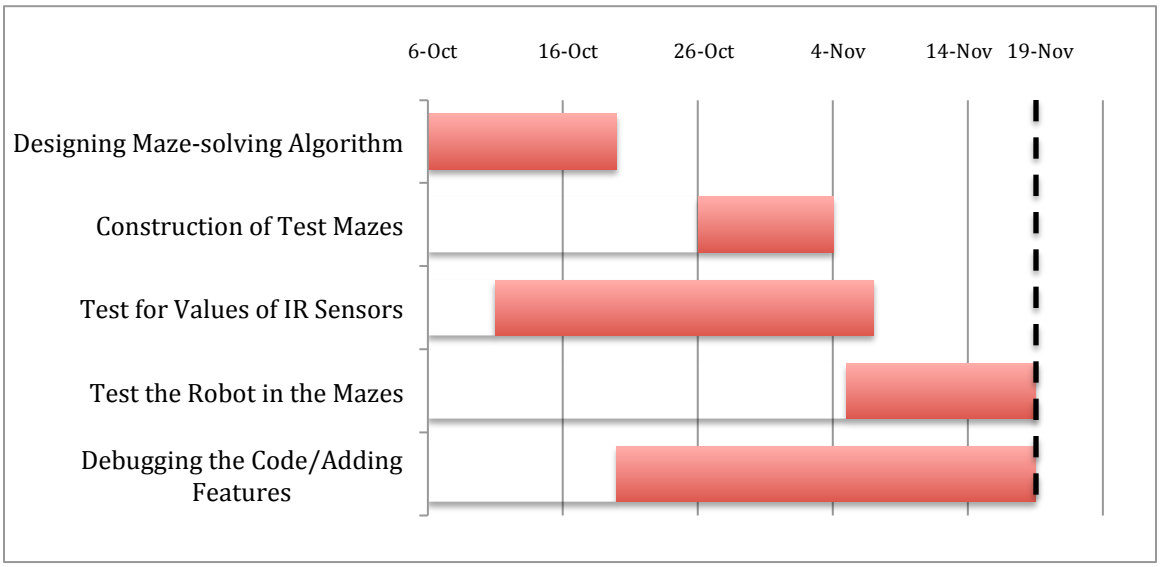
The "wall follower" method does not work in all mazes. It cannot solve a maze if the walls of the maze are discontinuous. For example, if the robot starts inside of a one-layer enclosed wall with one opening, the "wall follower" method leads the robot to repeat the following steps: follow the wall to the opening, out of the opening, go around the enclosing wall, go back in the "room", go around the edge of the "room", and repeat (shown in Figure 2).



(Figure 2) Following the "wall follower" method, the robot is stuck in the same loop.

The amount of time it takes for the robot to solve the maze can be long. All the obstacle sensors are designed to detect obstacles in front of the robot. Since there is no side-sensor, the robot must turn after it has travelled a certain distance.

Time Estimate



(**Figure 3)** Graph produced following the tutorial by Jodi Sorensen. [4]

References

[1] W. D. Pullen, "Think Labyrinth: Maze Algorithms" (2014, June 30). *Astrolog*. [Online]. Available : http://www.astrolog.org/labyrnth/algrithm.htm

[2] "Myro Reference Manual" (2013, April 24). *IPRE*. [Online]. Available: http://wiki.roboteducation.org/Myro_Reference_Manual

[3] "Myro Hardware" (2011, September 20). *IPRE*. [Online]. Available: http://wiki.roboteducation.org/Myro_Hardware

[4] J. Sorensen, "How to Create a Gantt Chart in Excel" (2013, July 8). *Smartsheet*. [Online]. Available: http://www.smartshhet.com/blog/gantt-chart-excel