

Lego Ev3 Mindstorms Assignment

by Richard Lamnea

FILE	11200_RICHARD_LAMNEA_LEGO_EV3_MINDSTORMS_ASSIGNMENT_173939_1118916371.DOCX (1.09M)		
TIME SUBMITTED	04-MAY-2019 10:39AM (UTC+0100)	WORD COUNT	1697
SUBMISSION ID	106307331	CHARACTER COUNT	7837

University of Wales Trinity St David

School of Applied Computing

Lego Mindstorms Project

Integrating Project (M4X01506)

Richard Lamnea

1802749

20/03/2019

Table of Contents

Table of Figures.....	2
1. Introduction	3
2. Research.....	4
3. Plan	6
4. Design.....	7
5. Implementation	8
6. Results.....	11
7. Conclusion.....	12
References	13

Table of Figures

Figure 1- Obstacle Course	3
Figure 2 - Right hand wall follow algorithm.....	5
Figure 3- Robot Construction Design	7
Figure 4 - Ev3 Program for start of the maze.....	8
Figure 5 - Ev3 program - Check front sensor	9
Figure 6 - Ev3 program - Check for right opening.....	9
Figure 7 - Ev3 program - Wall follow adjustments	9
Figure 8 - Ev3 program - End celebration	10

1. Introduction

The task assigned was for a team to create a Lego Mindstorms Ev3 robot, and to program the robot to navigate through an obstacle course from point A to point B.

The obstacle course had two identifiable stages which were split into 3 problems to solve: -

1. A guidance line for the robot to follow, with a curved section.
 - I. An obstacle placed on the guidance line.
2. A 3D physical maze.

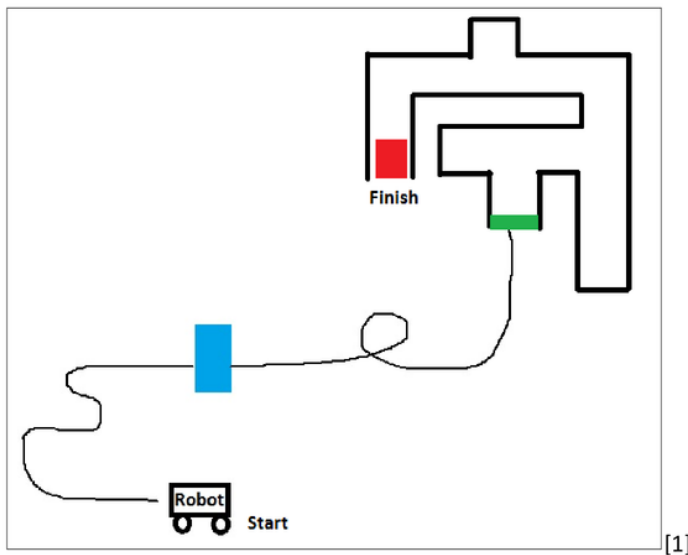


Figure 1- Obstacle Course

The team consisted of Richard Lamnea, Jaromir Pauer, and Richard Green who each contributed to the creation and programming of the robot. Jaromir was the lead member for the first section which was to solve how the robot would sense and follow the guidance line. Richard Green was the lead member for the second section which was to solve the problem of removing the obstacle off the guidance line and to continue back onto the course. Richard Lamnea was the lead member for the final section, which was to solve how the robot would traverse the maze and complete the course.

2. Research

Research was conducted into maze theory algorithms to be able to understand how to approach the software development of the maze section. There are multiple algorithms for approaching the problem of solving a simple maze.

- Random Choice Algorithm

This algorithm is the simplest way of maze solving. The robot randomly chooses to turn right or left, the robot will eventually exit the maze, this is not a very fast or efficient approach.

[2]

- Left or Right Wall follow algorithm

This algorithm can solve any maze with no islands. i.e. if the robot follows one of the walls, the robot will eventually exit the maze. [2]

- Branch and return algorithm

The principle of this algorithm is for the robot to explore every branch of the maze. This algorithm requires the robot to store data of where a branch has been located and return to that position before continuing into the next branch. [2]

The left or Right wall follow algorithm was chosen to be used for the design of the robot because it is the fastest method out of the simplest of algorithms to create in the time given for the project.[3]

Further research into how the algorithm would be approached was conducted.

The left or right wall following algorithm works the same way for each version. The robot keeps an eye on the right or left wall and navigates the maze until it finds the end. The Right-hand method works by focusing on right and front sensors with an option to turn depending on the readings from

those sensors. The robot will turn right if it can, and if there is a wall in front it will turn left which can be seen in Fig 2.[4]

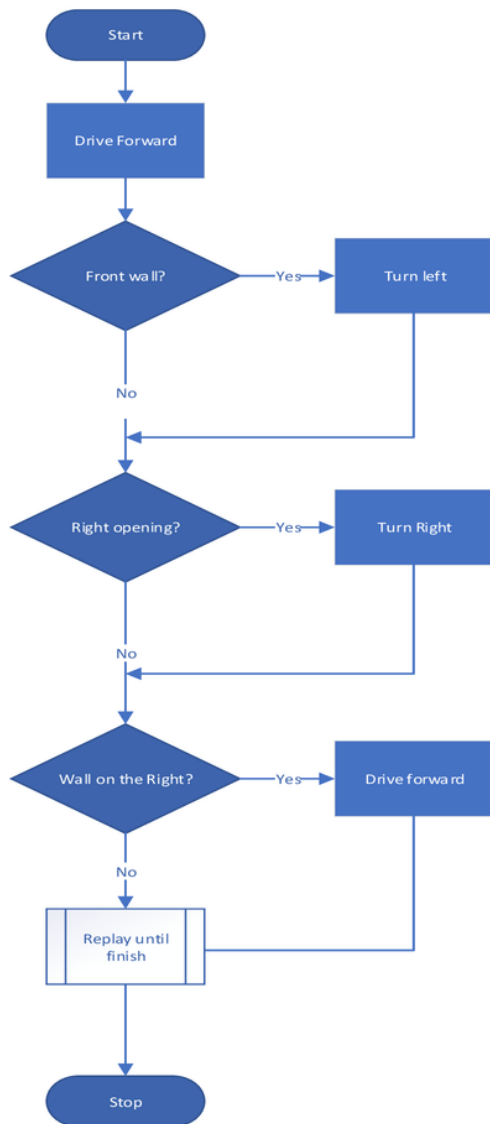


Figure 2 - Right hand wall follow algorithm

The problem of the right-hand wall follow method is that it is not the fastest or most efficient method but as the maze assigned was quite simple and had an exit on the outer wall this method was chosen.

3. Plan

The team planned that to overcome the challenge, research would be the most important tool to start the project. Two weeks were dedicated to researching how to use the Ev3 software and hardware, and which algorithm was best for their section of the maze.

Once the team had researched their subjects and understood how to use the software the team discussed how the robot's construction would be applied based on their findings. A week was spent creating a basic prototype with each member of the team having an equal input to the construction design.

After the prototype had been constructed, basic algorithms were put together based off the research conducted, so that the robot could be tested and tweaked when approaching each obstacle.

That last two weeks were spent refining the algorithms so that the robot could complete the course as fast as possible, without many errors.

4. Design

The robot's physical design was created to be able to approach all 3 sections of the obstacle course and allow it to complete them all. The robot has one colour sensor, positioned below the lift motor to allow the robot to follow the guidance line using a zig-zag line following algorithm. The robot's CPU unit was positioned at the rear of the robot to allow for counter balance for the lift and grab mechanism used to move the obstacle on the guidance line. The robot also has 2 ultra-sonic sensors, one at the front to detect objects and one on the right-hand side, to be used for right hand side wall following algorithm. The robot was built with a wide wheel space and reverse force motors applied to the wheels to increase torque. This allowed the robot to lift a heavier object which can be seen in Fig 3.

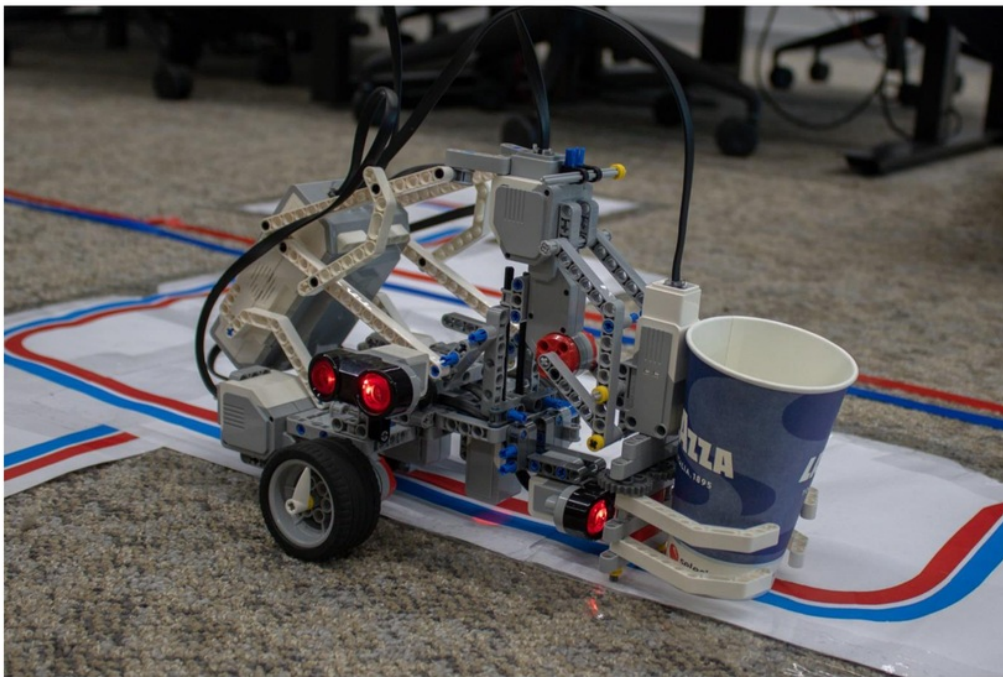


Figure 3- Robot Construction Design

The initial prototype had an even wider design but was changed as it affected the turning circle of the robot which in turn made it more difficult to follow the guidance line, and re adjust when following the right-hand wall.

5. Implementation

The maze solving algorithm was created in the Ev3 Lego Mindstorms programming environment. The maze program starts once the robot has reached the black start line and continues to drive forward 10cm, then plays a sound just before the maze loop starts which can be seen in Fig 4. Once the maze loop has started, the robot first checks if there is an object in close proximity to the front sensor, If it is it will turn 90 degrees to the left, if not it will check the light sensor to detect if the finish line has been reached. If the finish line has been met the program will end which can be seen in Fig 5. If the robot has not detected a red colour on the colour sensor or has not detected an object in proximity to the front sensor then it will continue the program and check if the robot can make a right turn which can be seen in Fig 6. If the robot cannot make a right turn because there is an object detected within range, the robot will then attempt to adjust its position to be in range of the wall and to follow it. The wall follow loop only iterates once as to let the program continue checking the sensors quickly which can be seen in Fig 7. Once the robot has reached the finish line the robot stops, plays a tune and spins on the spot to emulate a celebration which can be seen in Fig 8.

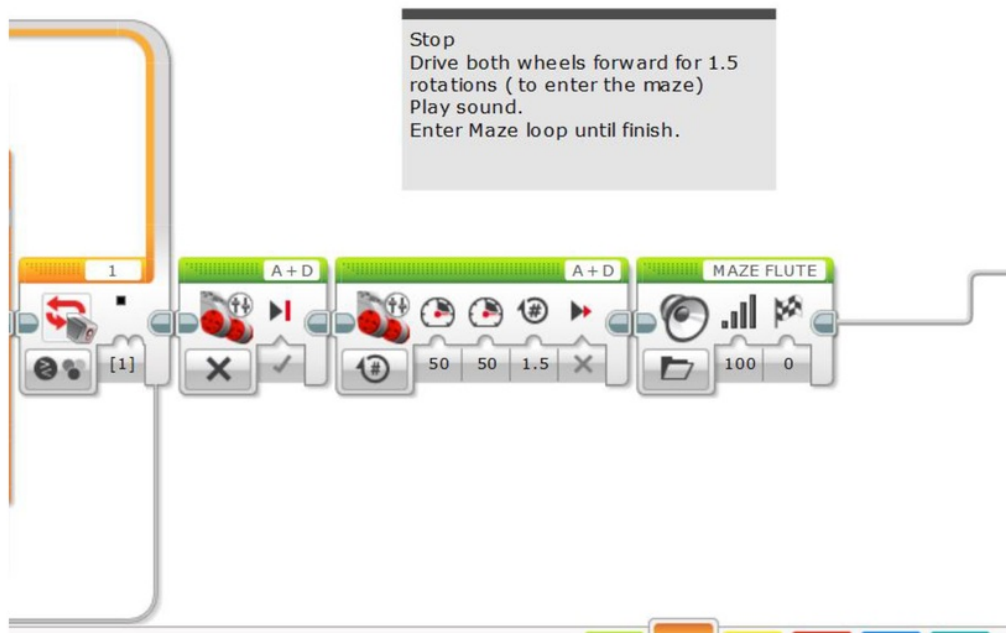


Figure 4 - Ev3 Program for start of the maze

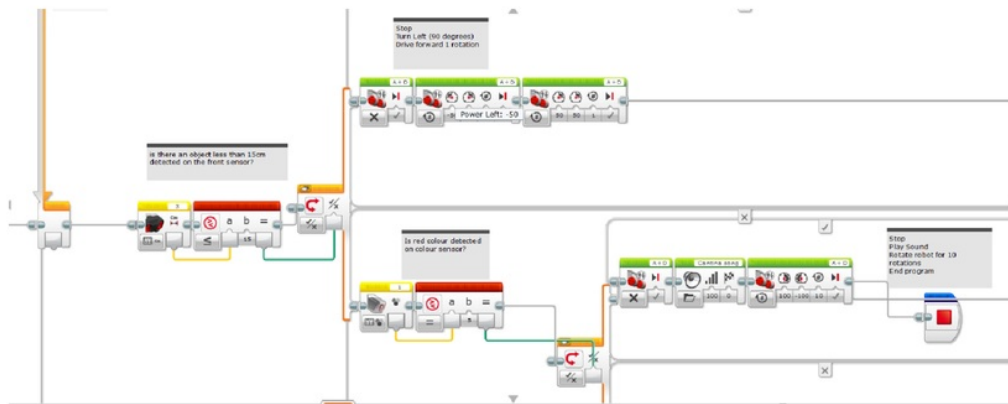


Figure 5 - Ev3 program - Check front sensor

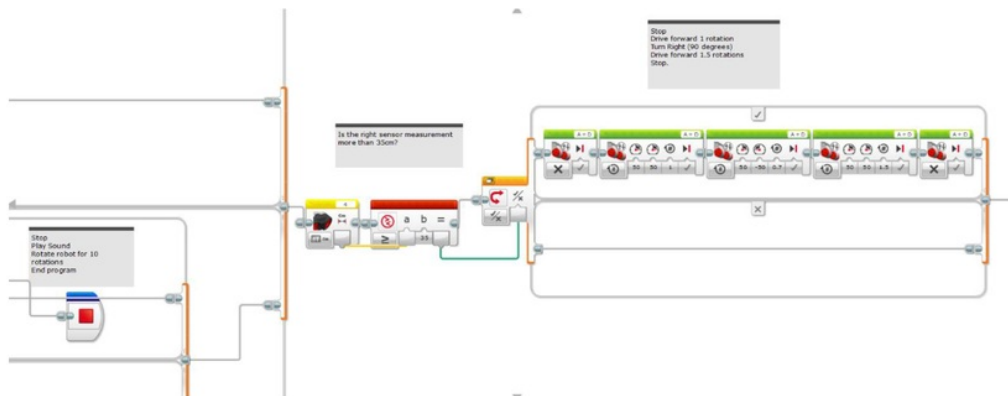


Figure 6 - Ev3 program - Check for right opening

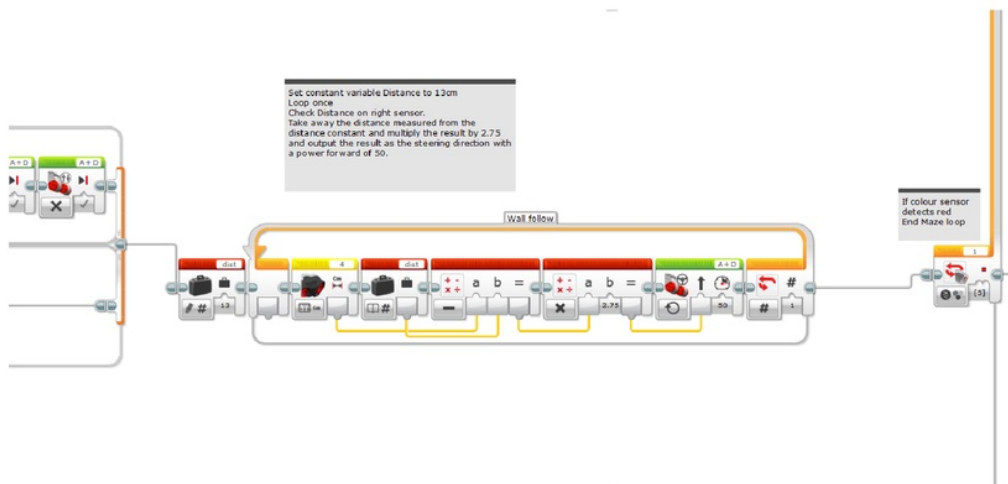


Figure 7 - Ev3 program - Wall follow adjustments



Figure 8 - Ev3 program - End celebration

The software for the maze algorithm had to meet certain criteria to be able to complete the maze. The program needed to have a start point, an end point and be able to make simple decisions based off the data read from the ultra-sonic sensors. As researched the initial function would be to read the front sensor, which would then check if the robot was in proximity of an object in front, if so, make a left turn. Once complete the next function must read the right-hand sensor and determine if it can turn right because there is not an object in proximity or if there is an object in proximity adjust to keep within the distance range of the object. The program must end when it reaches the end point.

6. Results

After finalizing the software development of the robot, the robot was tasked with attempting the obstacle course. The robot correctly followed the line, until it reached the object. The robot lowered its grab mechanism and drove forward until it was in contact with the obstacle. The robot's claw grabbed the object and correctly lifted the object back onto the robot. The robot then continued to follow the line until it reached the start line of the maze. The robot correctly detected the right-hand wall and followed it until it reached the end of the maze. The robot successfully completed the maze and met all the criteria it was set.

7. Conclusion

The entire project was successful, the robot performed all the tasks set. The team managed to plan and correctly implement their solutions to the problem and completed it within the time frame given. To be able to improve the robot's design more time was needed but the project had a deadline. If given more time modifications could have been added to the line-follow algorithm and the maze to improve performance and speed. The project allowed the team members to learn how computer logic can be implemented into real world scenarios by using sensor data. The team also learnt about different approaches to solving mazes and basic engineering principles when constructing the robot.

References

- [1] G. Thomas and M. Abuhmida, 'Lego Mindstorms Project Assignment Brief'. 02-May-2019.
- [2] M. Chand, M. Goel, and S. Rathore, *G.G.S.I.P.U., New Delhi*.
- [3] 'ArcBotics - Maze Solving'. [Online]. Available: <http://arcbotics.com/lessons/maze-solving-home-lessons/>. [Accessed: 28-Apr-2019].
- [4] A. B. S. Saman and I. Abdramane, 'Solving a Reconfigurable Maze using Hybrid Wall Follower Algorithm', 2013.

Lego Ev3 Mindstorms Assignment

GRADEMARK REPORT

FINAL GRADE

70/100

GENERAL COMMENTS

Instructor

Report (60%) overall : 54%

Introduction - 3/5

Introduction is very brief, it is a basic review of the task required.

Background research - 7/15

Very limited reference list. There is good in text citation. The research activity is limited with gaps in the knowledge.

Project plan - 5/10

The project plan is of a good standard but the information provided is quite limited. .

Design Process - 23/40

clear planing and discussion of the available hardware and software. detailed design process.

Evaluation and Results - 16/30

the results analysis is mostly combined in other section. the conclusion is very brief. but the over all quality of the results is of a good standard.

Demo (40%) - Overall 93%

Technical knowledge/skill - 18/20

Analytical and problem-solving skills - 18/20

Quality of result - 57/60

An excellent implementation. Line is followed precisely, good speed, obstacle picked up and released at the end. Maze is particularly good because you have a side sensor to mitigate for any unexpected maze layouts or imprecision in angle of turn. Maintains good distance from the wall. Good use of sensors throughout. Good speed maintained throughout.

(We enjoyed the robot dance at the end!)

Overall mark for module: 70%

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11
