

Integrating Project Assignment - Final

by Sophie Meyrick-Vernon

Submission date: 07-May-2023 10:31AM (UTC+0100)

Submission ID: 205058097

File name: 41934_Sophie_Meyrick-Vernon_Integrating_Project_Assignment_-_Final_489678_1182543711.docx (256.73K)

Word count: 1595

Character count: 7856

Sophie Meyrick-Vernon

Student number: 2221762

**Computer Networks and Cyber
Security: Foundation Year**

Integrating Project (ACCA4008)

Table of contents

1. Introduction
2. Research
3. Planning
 - a. Line & obstacle solution
 - b. Maze solution
 - c. Robot design
 - d. Time management
4. Testing
5. Assessment

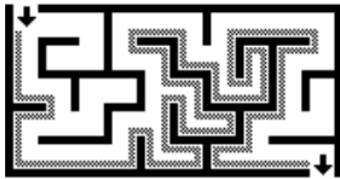
1. Introduction

The aim is to build a robot using Lego Mindstorms EV3 program environment. The team must create a program which will direct the robot to navigate a route. The route should include four segments; following a line, navigating around an obstacle, re-joining the line and complete a maze. The robot should also play a sound to indicate the maze has been completed. This should be done as a part of a group where each team member is responsible for taking the lead on at least one segment. The team members should also be able to demonstrate working as a team.

2. Research

When researching robots following lines within the EV3 Programming environment many examples included a simple black line to follow. The solution included a black line over a white background using a colour sensor to determine which direction to steer the robot. The sensor detects how much of the black line is under the sensor keeping the robot at the edge of the line and moving left and right when necessary [1]. This was done by using a loop with a switch block inside. When the colour threshold is below a set figure it would steer left otherwise it would steer right. Using these rules the sensor determines which part of the switch block to run [1].

Upon further exploration it was found that it is possible to use the touch sensor to interrupt a loop and carry out a new set of instructions. The code waits for the touch sensor to be true before executing specific lines of code [2].



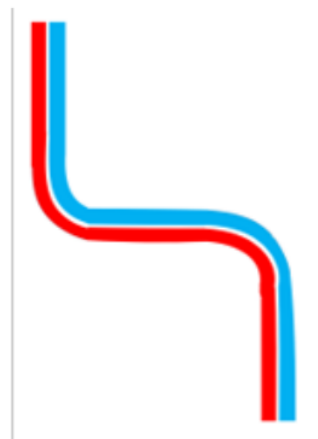
[Figure 1] CC BY-SA 3.0

When researching automated maze solving one theory explored was the wall follower method. This is a commonly used maze solving algorithm for robots. The theory is that when the robot enters the maze it is to either follow the right-hand wall or the left-hand wall until reaching the end [2]. As seen in figure 1 the grey line represents the route of the robot when programmed to follow the right-hand wall. This method of maze solving is most successful when all the walls are connected or to the outside boundaries [3].

3. Planning

The planning process started with the allocation of team leader roles for each segment. The team discussed each team member's strengths and weaknesses. The team decided that one person would take charge of the entire line segment including the manoeuvre around the obstacle, another person would oversee the maze programming, and the third person would be leader of the robot design. The team were to collaborate on all segments then the segment leader would make the final decision. This method ensured that each team member contributed equally to the whole project.

a. Line & Obstacle Solution

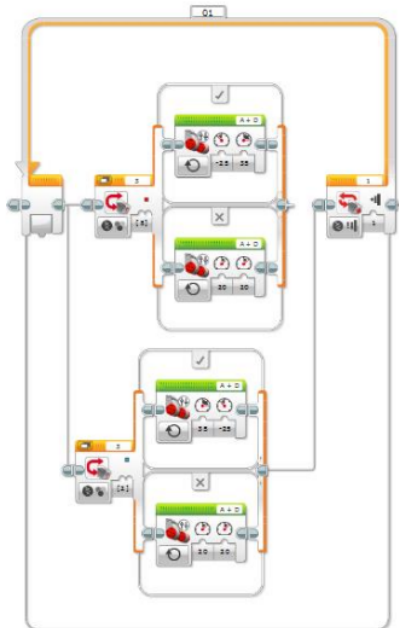


[Figure 2]

After discussion the segment leader decided that using a colour sensor built into the robot to detect two different coloured lines would yield the best results. The team decided to use red and blue lines as a guide for the robot. Taking what had been determined from the previously completed research the solution chosen was to use a loop to set a condition where each time the sensor detected red the robot would move slightly to the right. Then when the sensor detected blue it would move the robot slightly to the left. This would keep the sensor in between the two colours enabling it to follow the line. To begin with the team tested this method with a straight line to manipulate the left and right movements. It was found that small movements left and right were successful in moving the robot forwards while following the line. When the solution was tested on a curved line, as seen in figure 2, it was important that the left and right movements were small enough that they did not lead the robot off track.

To avoid the obstacle the robot must first detect the obstacle. The loop for this segment will have a condition where when the touch sensor is pressed the robot will carry out a set of movements to move backwards to make enough space to continue around the block to the right.

Code for line following solution:



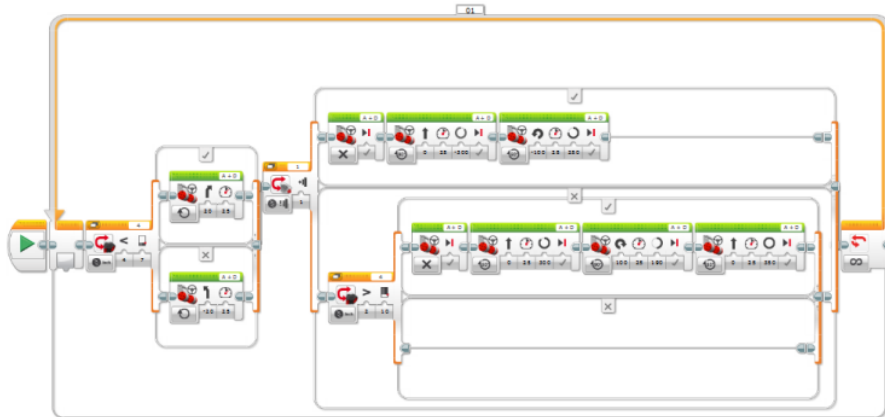
Code for manoeuvre around the block:



b. Maze Solution

After discussing the research completed by the team it was determined that the hug the left theory would be a strong solution to solve the maze. For this to be a viable solution the team decided to use two ultrasonic sensors. One sensor would measure the distance from the robot to the left wall and the other sensor would measure the distance for obstacles ahead. The left sensor would constantly check that that the robot is near the left wall. This means the robot will detect an upcoming turn and continue to follow the left wall. Once the ultrasonic sensor detects an obstacle ahead the robot would make a right turn to avoid the obstacle then continue following the wall on its left.

Code for maze solution:



c. Robot design

For the proposed solutions to complete the segments it was determined that two ultrasonic sensors would be required for the robot. The first sensor needed to be neatly located on the left side of the robot facing out. It was decided that the most effective positioning would be the front left of the robot to ensure the robot has the most amount of time to react. The second sensor should be located at the front of the robot to detect obstacles ahead.

For the line following segment it was necessary for the robot to have a touch sensor to detect the block in front of it. The sensor needed to have a flat contact piece to detect the touch and a long enough stick so that when the robot touches an obstacle in front it has enough space to turn to avoid colliding with the obstacle.

For the follow the line segment the robot requires a colour sensor to detect the lines on the floor. The colour sensor is also needed to recognise the change in environment. When the robot detects green on the floor the program will then use that information to transition into the maze program. Once the robot detects yellow on the floor it is programmed to signal the maze ending and to play the sound. In testing it was found that the colour sensor being placed in a central position at the front of the robot and a few centimetres from the ground gave the most accurate readings. Therefore this was determined to be the most successful position terms of the robot design.

4. Time management

The team collaborated on the solution once per week. The group utilised their time effectively by planning out the tasks and allocating specific timings for each. The program was split into segments. The team focused on completing them one at a time. During each class session the group reviewed their progress from the previous week and set goals for the current session. Once a goal was completed it was tested and amended until there was a positive outcome.

5. Testing

Testing was done throughout the coding process using trial and error. When creating the line segment solution the robot's speed was adjusted in increments of five until the robot completed the course as planned. Similarly for the manoeuvre around the block segment changing the angle at which the robot moved in small increments until it was successful.

When testing the maze solution the robot was placed into the maze to monitor how it performed. Each time the robot did not perform as expected the program would be paused to adjust the code. This was done until the robot completed the maze multiple times without any issues.

6. Assessment

The overall outcome was a success. The line following segment performed as planned. The movement around the block and re-joined the line. The first bug that was identified was after the colour sensor detected green on the floor it paused as planned but did not follow straight into the maze program. The code for the maze had to be run separately. This was missed during the testing process. After manually running the maze program the robot performed as planned moving into the maze hugging the left side. It turned left on the first corner successfully but following that on the right turn the robot did get caught on one of the maze walls as the second turning was too wide. This could have been easily prevented if more maze variations were used in the testing process. Though the robot design is small the turns it was programmed to take could have been sharper. Overall the solutions were successful the robot completed the maze and played a tune to confirm its completion.

References

[T. Griffin, "The Art of Lego Mindstorms EV3 Programming Environment," in *The Art of Lego*
1 *Mindstorms EV3 Programming Environment*, USA, William Pollock, 2014, pp. 70-71.
]

[T. Griffin, "The Art of Lego Mindstorms EV3 Programming Environment," in *The Art of Lego*
2 *Mindstorms EV3 Programming Environment*, USA, William Pollock, 2014, pp. 78-79.
]

[Unknown, "Maze solving algorithm," Swuecho, [Online]. Available:
3 https://swuecho.fandom.com/wiki/Maze_solving_algorithm. [Accessed 24th April 2023].
]

[A. Y. Z. Dao, "Maze Escape with Wall-Following Algorithm," 5th February 2021. [Online]. Available:
4 <https://andrewyong7338.medium.com/maze-escape-with-wall-following-algorithm-170c35b88e00#:~:text=Wall%2DFollowing%20Algorithms&text=The%20rules%20are%20based%20on,that%20leads%20to%20the%20exit>. [Accessed 24th April 2023].
]

Integrating Project Assignment - Final

GRADEMARK REPORT

FINAL GRADE

80/100

GENERAL COMMENTS

Instructor

ASSESSMENT COMPONENT 1 (60%)

Project Introduction 4/5

Background research 7.5/15

Project plan 9/10

Design Process 32/40

Evaluation and results 22/30

Total: 74.5%

Comments:

The Introduction would have been improved by inclusion of a diagram, it's a bit hard to visualise!

Research section is a bit on the short side. You should use this section to talk about the theory rather than your own implementation (save this for later in the report_ - what possible maze solving algorithms are available. You have mentioned right-hand wall or left-hand wall, but very briefly.

Planning section is good, and is clearly shown. You could have elaborated on why the two lines were necessary - why not one line and detect the white?

Good design shown with suitable code excerpts, and the concept was proven in the demo. Always say why you did things - eg why did you use ultrasonic sensors over touch sensors for the maze?

For the results, it would have been good to show some screenshots of the robot in action - this is a very visual assignment so would assist the reader. But you have considered some of the problems and resolved them well. You might want to include a section on what you would do differently next time, could anything be improved or would you consider a different approach?

ASSESSMENT COMPONENT 2 (40%)

Technical knowledge/skill 17/20

Analytical and problem-solving skills 17/20

Quality of result (performance of the finished application or suitability of the design) 54/60

Total: 88%

Comments: A strong demonstration, which met the requirements. There was a bit of improvement to be made around transition between line and maze, but this was a minor point. Well done.

OVERALL TOTAL FOR MODULE: 80%

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8
