Final Report

Automated Farming Route Tractor by BatMobile

Version: 1.0

Date: 23/01/2023

Group Members

Name	Matric Number
Tan Yong Le	s2164472
Boon Yin Yin	s2149527
Darkhan Baibulat	s2172558
Xue Bai	s2116024
Liow Gian Hao	s2178227

1. Introduction

The globe has historically undervalued the agricultural sector, and even though it is one of the most significant industries, it is nevertheless given less attention than other sectors. The absence of funding from the public and commercial sectors is one of the critical reasons for the low success rate of talent recruitment and eventually causing the slow advancement of technology in the agriculture industry.

The purpose of our project is to contribute to the technologies used in the farming sector, mainly the spraying process. Inspired by the success of various automation devices like self-driving cars and robotic vacuum cleaners, we aim to apply the same concept to automate the selective spraying process. Our developed prototype can learn the permitted driving routes and choose the best route to reach the destination. When used in a real-world setting, selectively spraying of preventive treatments onto a portion of the fields can be automated, optimized, and done in the shortest period to ensure the harm to the crops will be treated in the shortest time possible.

Insecticides, pesticides, fungicides, and other preventative treatments are frequently applied to crops through crop spraying operations to control pests, treat diseases, and enhance product quality. Historically, the procedure has been completed by sending out aircraft to spray onto the crops from a distance. Despite the fact that most pests and diseases have irregular spatial distributions, this agricultural method routinely applies pesticides uniformly throughout all of the fields.

However, reducing the use of pesticides and fertilizers in agriculture while maintaining the same level of crop production has become a key goal for modern agriculture due to the growing health concerns caused by pesticide residues in food and water as well as occupational exposure (farm workers frequently work in open fields and greenhouses where pesticides are present). Consequently, research and development into selective spraying have occurred during the past two decades (Paice et al., 1996; Slaughter et al., 1999). Selective spraying will reduce the treatment's usage by allowing it to be applied more precisely and, at the same time, reduce the harm caused to the nearby residences or wildlife.

The modern method of selective spraying typically involves quad bikes and boom sprayers, where the worker drives to the designated region and performs the activity; moreover, this still needs to address the problem of occupational exposure noted above. Therefore, a system for autonomous selective spraying is required which must fulfill the criteria below:

1. Travel to the destination along the permitted route without destroying any crops.
2. Travel to the destination in the shortest path to ensure that harm to the crops is eliminated as soon as possible.
3. Automatically execute the selective spraying activity.

Main objectives:

- To base an optimized algorithm for the robot in finding the shortest path
- Learn more about the basic Floyd-Warshall algorithm
- Experience teamwork by completing tasks knowing the strengths and weaknesses of each
- Determine the path to choose by following the drawn line, determine all paths and calculate the shortest distance to the goal

2. Roles

The WOA team includes 5 people, every two weeks 1 time we provided a FILA form to show the overall progress of each participant.

During 9 weeks, a lot of tasks were included.

Common tasks:

1. Search for articles and materials for detailed study

2. Building a robot and implementing an algorithm

3. Providing support in writing code

Roles of participants separately:

1. Boon Yin Yin & Xue Bai - research and understanding of the algorithm. Drawing lines for a car, building a robot through an algorithm, hardware process. Install the camera module on the car. Python code to process the video input of the camera, identify lanes & goals, verify whether it is a valid path, and whether it is a goal and which direction to turn. Will test on pre-recorded video first while waiting for the camera to be assembled. Prepare updated version of SRS report

2. Darkhan Baibulat - finding the shortest path in different directions, finding a stop for the direction of the robot, helping to build a robot and a road template using lines. Prepare Final Report.

3. Tan Yong Le and Liow Gan Hao - implementation of the algorithm through Python and testing, implementation of the robot into this algorithm. Integrate the final algorithm inside the robot. Predefined graph of farm in the code, and use the goal detection to determine whether the robot already reached a goal and when it should stop. Implementation of Floyd warshall and navigation algorithm in robot. Prepare slides.

3. Project Analysis

This project was developed in 9 weeks, including 3 main areas of activity:

1. Assembling car and hardware components

2. Finding a solution in the form of the Floyd Warshall algorithm

3. Implementation and integration of the algorithm, code into the robot

Your project is to assemble an autonomous driving car that is able to help with pesticide control in a farm. The car will be able to traverse from the location of one crop to another in the shortest distance possible. The algorithm that we are using to find the shortest distance is a path finding algorithm called Floyd–Warshall. Below shows an example of the farm and its distance matrix:

To do this, the first step of our project was to search for material to analyze the issue in agriculture and propose a solution. Enough sources have been used to fully investigate the algorithm and its optimization to solve the problem. The necessary skills in computer science and robotics were included. A Software Requirements document was presented to disassemble the full analysis of the idea and its solutions, and a basic prototype was provided.

Have been identified:

1. Product features
2. The roles of the problem
3. Functional requirements
4. Graphics and drawings of the prototype
5. Algorithm analysis and optimization

After studying all the necessary materials, we started working using the necessary Raspberry Pi components and the robot's adaptation to movement.

Was included:

1. Implementation of the shortest path search and the best option
2. Embedding life into a robot using an algorithm
3. Sketches on a map with lines for certain structures (the map is presented in the Github link).

During the implementation of our algorithm, the robot turns in different directions, stopping by the amount of time that was successfully developed. A camera was installed to connect and embed through the data camera. Python code to process the video input of the camera, identify lanes & goals, verify whether it is a valid path, and whether it is a goal and which direction to turn. Will test on pre-recorded video first while waiting for the camera to be assembled.

After that, the algorithm was implemented inside the robot and navigation. After that, there was integration through the car.

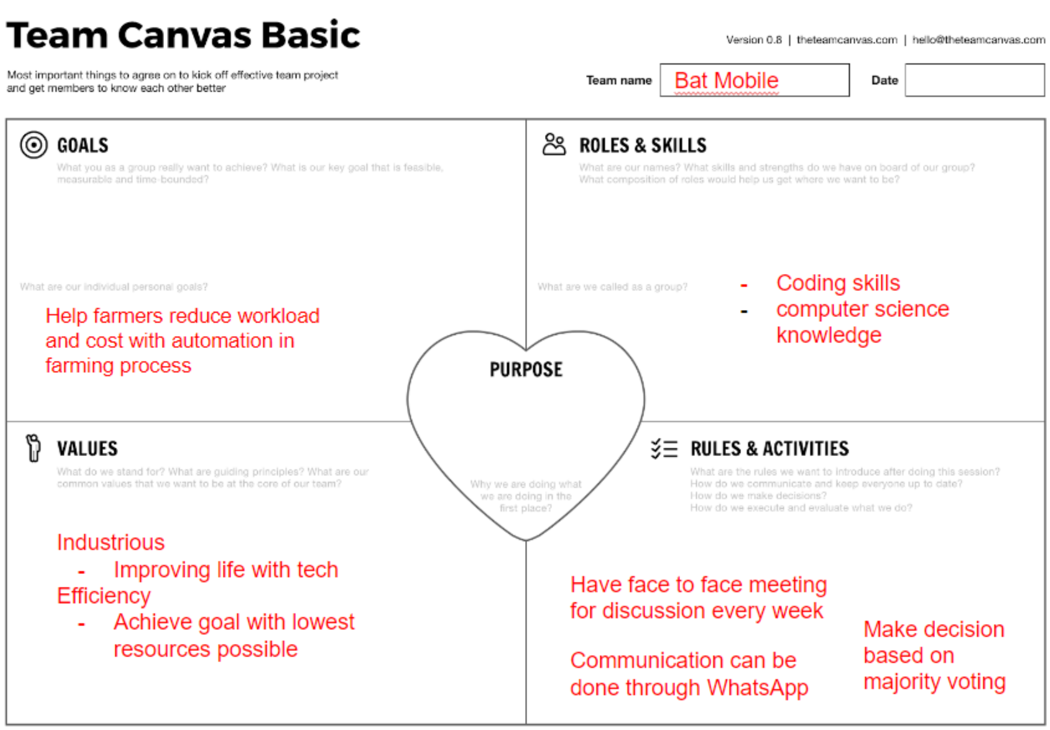
At the final stage, we were finishing the rest of it. Predefined graph of farm in the code, and use the goal detection to determine whether the robot already reached a goal and when it should stop. Tidy up the code and upload to Github. Prepare demos for presentation. After successful work, a general report, SRS v2..0 and comments of each participant on the project were written. The last step was the presentation of the robot and the operation of the car algorithm through our solution.

4. Planning and Execution

4.1 Team Building and Empathize

4.1.1 Team Profile

- **Team Name :** Bat Mobile
- **Team Canvas**
    - https://drive.google.com/drive/folders/1AkdoRcIiCcuJiVyR7i47P78EZL47TNGl?usp=share_link
- **Github repository (Implementation)**:
    - https://github.com/tyl1999tyl/WOAxWOXRoboticCar.git
- **Github repository (Docs & personality test)**):
    - https://github.com/tyl1999tyl/WOAxWOXRoboticCarMaterials.git
- **Google Drive Link** :
    - https://drive.google.com/drive/folders/1I4bN9buDr0T3zpeHPQgwQsV5EBbGLjfa?usp=sharing



4.1.2 Empathize (Understanding The Users)

Interview

Who?

- Farmers

Where?

- Online farmers community group

Questions

- What are the most common tasks as a farmer?
- What is the most time consuming/challenging task?
- How many people are required to complete the task?
- How often do you perform these tasks?
- How long does it take?
- What risks are there working day to day as a farmer?
- What technology is available to you to perform tasks?

4.2 Define

4.2.1 Defining the Problem

- Agriculture in Malaysia is experiencing a low growth rate compared to other sectors.

- Lack of capital and support from government and private parties alike halts efforts for rapid transformation, structural change and integration with manufacturing.

- Technology is also developing at a slow rate, as well as innovative solutions with regard to product development, process and packaging.

4.2.2 Problem Statement

Agricultural operations like planting, plowing, spraying and harvesting take a long time to perform. Technology for agriculture is crude and outdated. Manual labor is heavily relied on.

Insecticides, pesticides, fungicides, and other preventative treatments are frequently applied to crops through crop spraying operations to control pests, treat diseases, and enhance product quality. Historically, the procedure has been completed by sending out aircraft to spray onto the crops from a distance. Despite the fact that most pests and diseases have irregular spatial distributions, this agricultural method routinely applies pesticides uniformly throughout all of the fields.

However, reducing the use of pesticides and fertilizers in agriculture while maintaining the same level of crop production has become a key goal for modern agriculture due to the growing health concerns caused by pesticide residues in food and water as well as occupational exposure (farm workers frequently work in open fields and greenhouses where pesticides are present). Consequently, research and development into selective spraying have occurred during the past two decades (Paice et al., 1996; Slaughter et al., 1999). Selective spraying will reduce the treatment’s usage by allowing it to be applied more precisely and, at the same time, reduce the harm caused to the nearby residences or wildlife.

4.2.3 Proposed Solution

Design a vehicle that is able to take the optimal path to go through the whole farm to perform basic agricultural operations like spraying.

The modern method of selective spraying typically involves quad bikes and boom sprayers, where the worker drives to the designated region and performs the activity; moreover, this still needs to address the problem of occupational exposure noted above. Therefore, a system for autonomous selective spraying is required which must fulfill the criteria below:

1. travel to the destination along the permitted route without destroying any crops.
2. travel to the destination in the shortest path to ensure that harm to the crops is eliminated as soon as possible.

4.2.4 Sources

Sources 1 - Interview with the Farmers

https://www.youtube.com/watch?v=Jj7Tfv9rTZo&ab_channel=UNGCMalaysia

- When asked about farming technology, farmers respond that it is quick basic and most area does not have advanced technologies
- The agriculture business mostly comprise of old people, therefore labor intensive jobs may be a burden for them to perform

Source 2 - Tractors

- Benefits of using tractors: https://fordtractor.ph/benefits-using-tractor-agriculture/
- Different types of tractors:
    - https://www.godigit.com/motor-insurance/commercial-vehicle-insurance/find/uses-of-tractors#:~:text=Tractors%20are%20generally%20associated%20with,the%20farming%20operations%20more%20convenient.
    - https://www.tractorjunction.com/blog/role-of-tractors-in-modern-agriculture/

Source 3 - Robotics and Automation in Agriculture: Present and Future Applications

http://arqiipubl.com/ojs/index.php/AMS_Journal/article/view/130

- agricultural operations divided into planting, inspection, spraying and harvesting
- autonomous navigation has been widely
- used in the development of Precision Farming (PF)
- agricultural automation is mainly focusing on autonomous vehicle applications such as robot or tractor where it is being used to minimize the tough, deadly, risky and long working conditions experienced by farmers and at the same time offers a precise and efficient operation and control system
- agricultural operation needs to be executed by different robotics and vehicle structure based on the type of land and operation requirement

Source 4 -EFFICIENCY OF TRACTORS USE IN AGRICULTURAL PRODUCTION

https://www.tf.llu.lv/conference/proceedings2018/Papers/N482.pdf

- analyzed the efficiency of tractors based on different type of agriculture work

4.3 Ideate

4.3.1 Planning

1. Product features

- Maze solving robot car with the ability to identify the path to take by following a drawn out line, identify all the paths, and compute the shortest distance to the goal.
- Users will place the car on the starting point of the maze, the car will automatically go through the route with the shortest distance to reach the goal.

2. Hardware & Software requirements

- Will be using Raspberry Pi for the computing chip
- Camera to recognize the path and goals
- Python programming language

4.3.2 Design Idea (reference)

4.4 Design

4.4.1 Software Requirements

- Operating System - Windows 10
- Web Browser - Google Chrome
- Microcomputer Operating System - Raspbian OS image
- Integrated Development Environment - VSCode
- Version Control System & Software - GitHub
- Programming Language - Python

4.4.2 Hardware Requirements

- Raspberry Pi 4 model B 4GB (Using)
- Raspberry Pi 4 - UK Plug power adapter 5V 3A USB-C (Using)
- 32GB MicroSD (Using)
- Micro HDMI Cable (Using)

- LoRaWAN GPS HAT for Raspberry Pi [Seeedstudio]
- Raspberry Pi camera module
- Mobile robot set (Using)
- 5 way line sensor
- 4WD robot car chassis with motor gear (Using)
- Motor driver (Using)
- Female SM cable (Using)
- Charger

5. Prototype Implementation

5.1 Optimization Algorithms used to find shortest path between two goals:

- Floyd Warshall
- The Floyd-Warshall algorithm is a dynamic programming algorithm that is used to find the shortest path between all pairs of vertices in a weighted graph. It is particularly useful for solving problems that involve finding the shortest path between multiple sources and destinations, such as the one described in this scenario. The algorithm can handle graphs with negative edge weights, and it can also detect negative cycles in a graph, which makes it well-suited for problems that involve finding the shortest path while avoiding certain constraints, such as damaging crops.

5.1.1 Function to print the shortest cost with path information between all pairs of vertices

```
# Recursive function to print the path of given vertex `u` from source vertex `v`
def printPath(path, v, u, route):
    if path[v][u] == v:
        return
    printPath(path, v, path[v][u], route)
    route.append(path[v][u])
```

```
# Function to print the shortest cost with path
# information between all pairs of vertices
def printSolution(path, n):

    all_shortest_path = []

    for v in range(n):
        for u in range(n):
            if u != v and path[v][u] != -1:
                route = [v]
                printPath(path, v, u, route)
                route.append(u)
                route_1 =  [x+1 for x in route]

                #print(f'The shortest path from {v+1} —> {u+1} is', route_1)
                all_shortest_path.append([v+1,u+1,route_1])
    return all_shortest_path
```

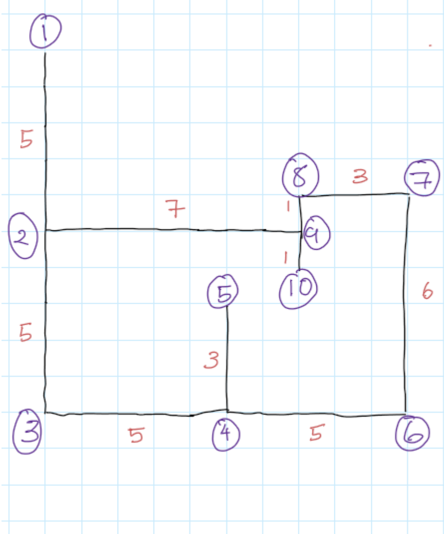
5.1.2 Function to run the Floyd–Warshall algorithm

```
def floydWarshall(adjMatrix, start, end):

    # base case
    if not adjMatrix:
        return

    # total number of vertices in the `adjMatrix`
    n = len(adjMatrix)

    # cost and path matrix stores shortest path
    # (shortest cost/shortest route) information

    # initially, cost would be the same as the weight of an edge
    cost = adjMatrix.copy()
    path = [[None for x in range(n)] for y in range(n)]

    # initialize cost and path
    for v in range(n):
        for u in range(n):
            if v == u:
                path[v][u] = 0
            elif cost[v][u] != float('inf'):
                path[v][u] = v
            else:
                path[v][u] = -1

    # run Floyd-Warshall
    for k in range(n):
        for v in range(n):
            for u in range(n):
                # If vertex `k` is on the shortest path from `v` to `u`,
                # then update the value of cost[v][u] and path[v][u]
                if cost[v][k] != float('inf') and cost[k][u] != float('inf') \
                        and (cost[v][k] + cost[k][u] < cost[v][u]):
                    cost[v][u] = cost[v][k] + cost[k][u]
                    path[v][u] = path[k][u]

            # if diagonal elements become negative, the
            # graph contains a negative-weight cycle
            if cost[v][v] < 0:
                print('Negative-weight cycle found')
                return

    # Print the shortest path between all pairs of vertices
    all_shortest_path = printSolution(path, n)

    return get_shortest_path(all_shortest_path,start,end)
```

5.1.3 Retrieve shortest path from start node to end node

```
def get_shortest_path(routes,start,end):
    for start_node,end_node,path in routes:
        if start_node==start and end_node==end:
            return path
```

5.1.4 Example of farm and its distance matrix



	1	2	3	4	5	6	7	8	9	10
1	0	5	-	-	-	-	-	-	-	-
2	5	0	5	-	-	-	-	-	7	-
3	-	5	0	5	-	-	-	-	-	-
4	-	-	5	0	3	5	-	-	-	-
5	-	-	-	3	0	-	-	-	-	-
6	-	-	-	5	-	0	6	-	-	-
7	-	-	-	-	-	6	0	3	-	-
8	-	-	-	-	-	-	3	0	1	-
9	-	7	-	-	-	-	-	1	0	1
10	-	-	-	-	-	-	-	-	1	0

- The numbers in dark purple and circled are the nodes (a.k.a location of the crops). They are the location that the robot car needs to reach.
- The numbers in red are the distances.
- The black lines are the path

5.2 Prototype Implementation - Version 1

5.2.1 Progress Summary

- Managed to assemble the physical robot car, joint effort from both WOC and WOA
- WOA implement algorithm for it to move in different directions (forward, reverse, left, right, pivot)
- WOA tested the car and it is able to move forward and left
- Implementation and a demo of Floyd Warshall is done.

5.2.2 Task Distribution

- Assembling the robot car : Tan Yong Le, Darkshan, Xue Bai, Boon Yin Yin, Liow Gan Hao & WOC Members
- Programming robot to move in different directions : Darkhan
- Research and understand the Floyd Warshall algorithm & How to apply the algorithm to our scenario? Drawing the example map and creating distance matrix : Xue Bai, Boon Yin Yin
- Implementing the optimization algorithm in Python : Liow Gan Hao

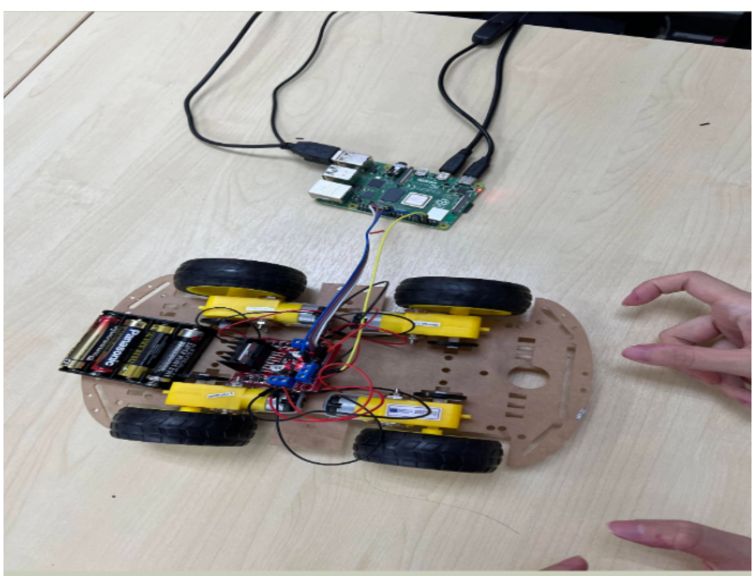
5.2.3 Optimization Algorithms used:

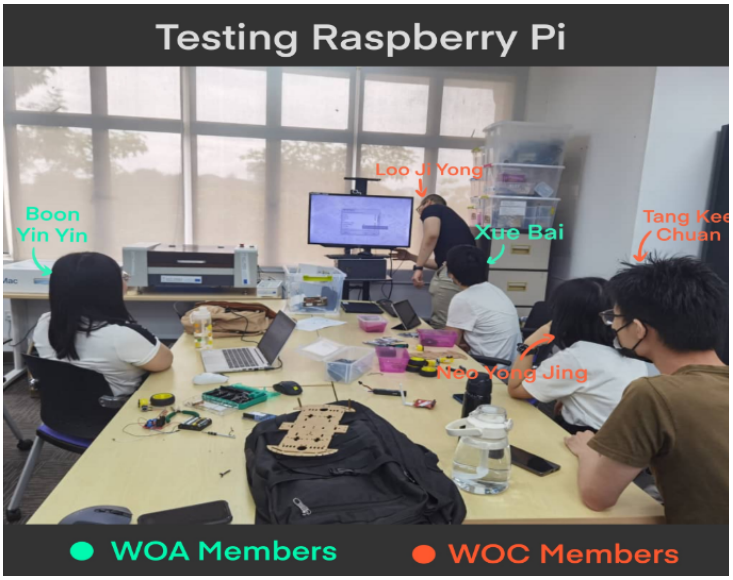
Shortest path algorithm - Floyd Warshall

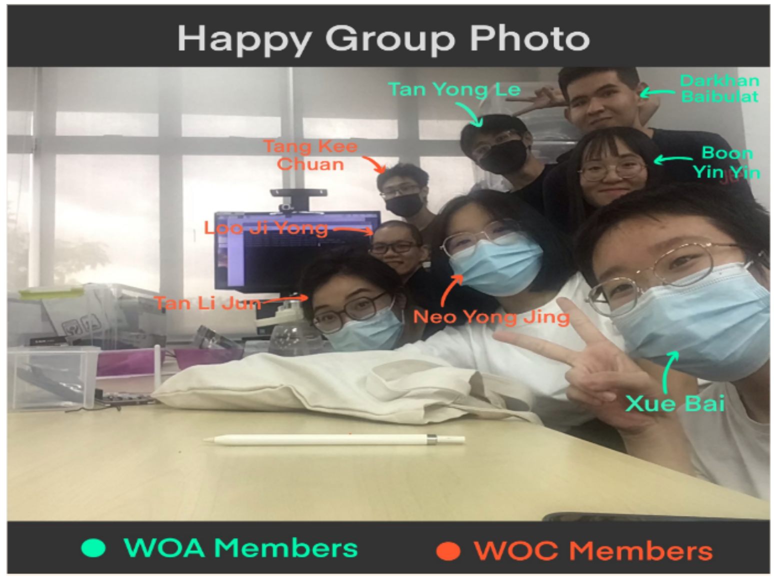
After running the Floyd Warshall algorithm, we can get the shortest path from location A to location B. Algorithm can be tested out in floyd_warshall_demo.py in Github

The shortest path from 1 —> 2 is [1, 2]
The shortest path from 1 —> 3 is [1, 2, 3]
The shortest path from 1 —> 4 is [1, 2, 3, 4]
The shortest path from 1 —> 5 is [1, 2, 3, 4, 5]
The shortest path from 1 —> 6 is [1, 2, 3, 4, 6]
The shortest path from 1 —> 7 is [1, 2, 9, 8, 7]
The shortest path from 1 —> 8 is [1, 2, 9, 8]

5.2.4 Assembled Car







5.2.5 Reference:

1. Raspberry Pi Robot Car
https://www.instructables.com/Raspberry-Pi-Robot-Car/
2. Raspberry powered RC car
https://github.com/holgi-s/rc-car

5.3 Prototype Implementation - Version 1.1

5.3.1 Progress

- Fixed the issue of the robot not being able to turn in different directions
- Implementation of navigation algorithm with pre-recorded video (without robot). It includes:
  - Detecting whether a goal is reached
  - Detecting the lane for robot to traverse
  - Recognize the direction the car should turn

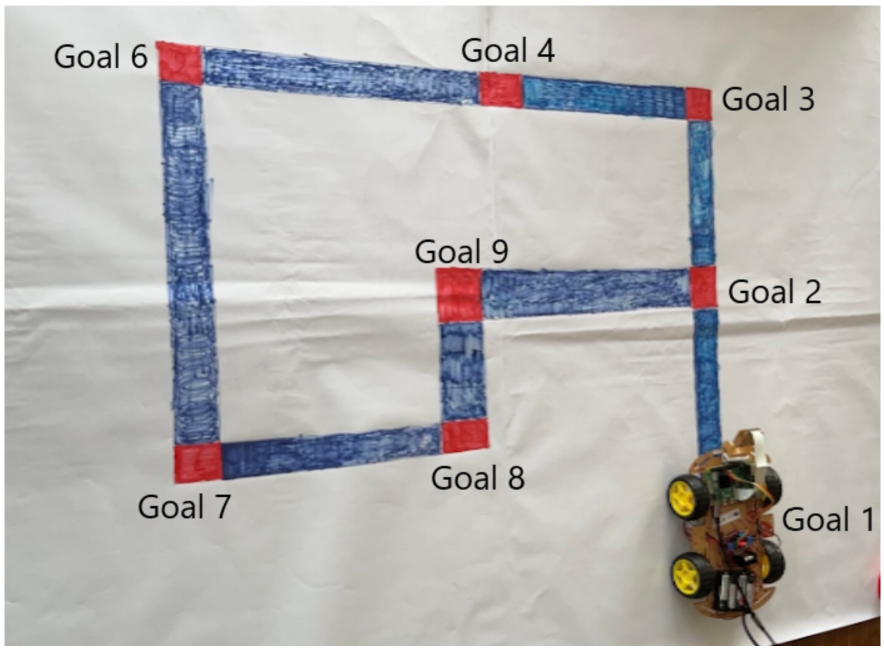
5.3.2 Task Distribution

- Solve the bug of robot not being able to move in different directions : Darkhan & Xue Bai

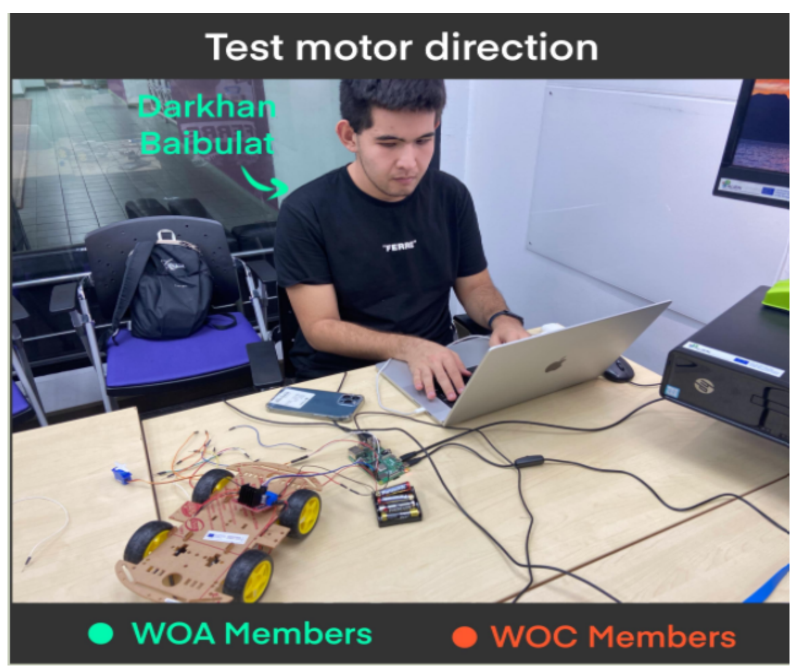
- Install camera module : Liow Gan Hao & Xue Bai
- Navigation code in Python : Boon Yin Yin
- Integration of navigation code in robot : Tan Yong Le

5.3.3 The Farm

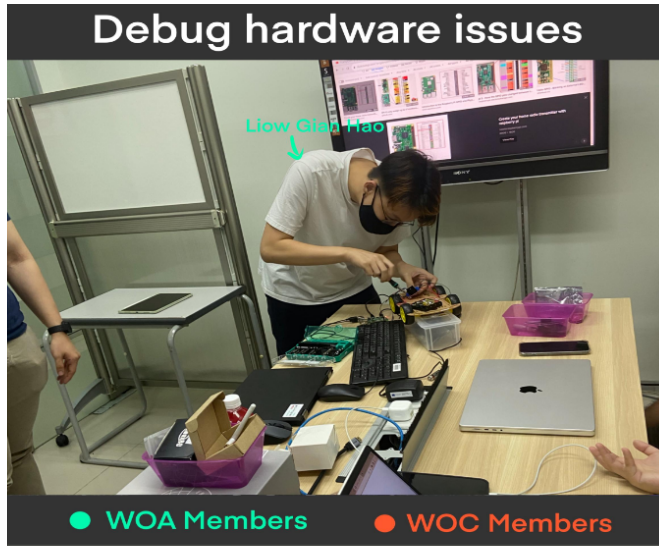
- This is a modified version of the path we drew previously (refer to Github page).
- The lanes are blue, and the red rectangles are the goals.



5.3.4 Photo




Query the solution of Bug
Liow Gian Hao
WOA Members
WOC Members


Debug hardware issues
Liow Gian Hao
WOA Members
WOC Members


Happy Group Photo
Darkhan Baibulat
Ding Zhan Chia
Xue Bai
Liow Gian Hao
Loo Ji Yong
Boon Yin Yin
WOA Members
WOC Members

5.4 Prototype Implementation - Version 1.2

5.4.1 Progress

- Implement navigation algorithm in the robot.
- Tested out whether the robot can detect lanes, detect goals and recognize the direction to turn. Testing is done in real time using a robot's camera instead of pre-recorded video.
- Integrate Floyd warshall algorithm with the navigation algorithm in the robot.

5.4.2 Task Distribution

- Implement the navigation algo in a robot. Ensure it is able to detect the lanes, stop when goal is reached and turn in different directions based on the curve of the lane : Boon Yin Yin & Xue Bai
- Combine navigation algo with floyd warshall. So that the robot is able to know which direction to take, which node it is at, and how to get from starting node to end node : Tan Yong Le, Liow Gian Hao, Darkhan Baibulat

5.4.3 Navigation algorithm breakdown

- Navigation algorithm is mainly done with image processing using OpenCV
- Goal Detection : Get the red rectangular shapes from the frames
- Lane Detection: Get the blue lines from the frames
- Direction Recognition: Split the frame in half. If there are more blue lines in the left side of the frame, turn left. If there are more on the right, turn right. Else it will just move forward.

5.4.4 Car moves different directions based on the lane detected and the direction recognized

- Demo video:
    - https://drive.google.com/file/d/1EJZ0HLnXWLbz5ee8UyqM0y6J7M6Dx_-n/view?usp=share_link

5.4.5 Lane and Goal Detection

- Demo video:
    - https://drive.google.com/file/d/1z2Sc5No0MJ7gFIjBnC28s8Tl3Ftyjzf0/view?usp=share_link

5.5 Prototype Implementation - Version 2

5.5.1 Adjustments Made

- Based on testing, the robot is not able to recognize the direction that it should turn well.
- There are several causes to this:
  - Robot is not able to function well once the battery is not enough. The robot consumes a lot of power, and has to keep replacing the battery.
  - Camera overheating and lagging. The camera is not able to keep up the speed of real time video processing and it will lag a few seconds before telling the car to turn.
- The whole graph of the farm is predefined in the code, and we used the goal detection to determine whether the robot already reached a goal and when it should stop.

5.5.2 Final process flow

1. Create adjacency matrix of the farm.
2. Given adjacency matrix, floyd warshall is used to calculate the shortest distance matrix.
3. The shortest path from start node and end node is retrieved. Example of output path [1,2,9,8]. Where 1 is the start node and 8 is the end node. The robot has to go through 2 and 9 to reach 8.
4.The robot will traverse through the nodes from the shortest path output above.
5. Robot will correct it's direction to be always north facing.

5.5.3 Optimization Is Use

Floyd warshall is used to calculate the shortest path between two nodes so that the robot can get to the end goal in shortest distance, instead of letting it blindly and sequentially navigate through the paths and nodes.

Example:

Non-optimized path from 7 —> 1: [7, 6, 4, 3, 2, 1] (Cost : 26)
Optimized path from 7 —> 1: [7, 8, 9, 2, 1] (Cost : 16)

5.6 Prototype Deployment - Version 2

5.6.1 Demo video of car getting from between 2 nodes

- https://drive.google.com/file/d/1EJZ0HLnXWLbz5ee8UyqM0y6J7M6Dx_-n/view?usp=share_link

6. Reflection

6.1 Tan Yong Le

One of the biggest challenges faced in this project is related to hardware. First, we do not have experience with implementation using hardware before. Second, hardware limitations exist. For example, inconsistency of the wheel rotation on different surfaces, camera latency causing goal and lane detection do not work sometimes. Another challenge that we faced is the duration for this project. This is because some of us are working as well and others are having many other subjects in this semester. It is hard for us to find a time for everyone to attend, causing us to work separately sometimes, hence lesser collaboration. However, we are still able to deliver a robot car with minimal functionality that fulfills the requirements, which includes assemblement and movement of the robot car, optimization of paths for a given map and so on.

6.2 Boon Yin Yin

This project was a challenging experience for me as I have no experience in dealing with hardware before, my group members and I have to spend a lot of time learning how to assemble the hardware and build the car to make it run. It is also hard for us to find time and work on the robot together in the classroom as some of us are working and having classes. We have also faced some difficulties in getting the materials needed. We were unable to find the camera from the box of materials provided and had to buy our own.

Although we are able to make the lane/goal detection and direction navigation work in the software and simulated environment, when implemented into the robot we still faced some hardware issues like the camera latency is not able to keep up with real time processing, part of it is due to our limited understanding of the hardware. Other products like color sensors may be able to fix this issue but due to time limitations and the lack of hardware parts provided, we were unable to fix it.

But despite the difficulties, we were able to complete the project and achieve the objective, which is to implement the optimization in the car and get it running.

5.3 Darkhan Baibulat

The project for this course turned out to be an excellent opportunity to gain useful experience, each of us tried to integrate into the life situation and implement a solution that will help many people in the agricultural sector. It was very difficult to find time to meet with the team, but we managed to get together and get a piece of knowledge and benefit from everyone. Personally, I had to work with a robot, developing an algorithm along the line with the rest of the team. In addition, for all 9 weeks, we faced various problems when assembling the robot, but quickly solved this problem, which gave us a good experience for future projects. The most important thing was not to give up and give a piece of desire to work in a team, as well as adaptation.

I learned a lot of information about algorithms, especially in Floyd-Warshall algorithm, as well as knowledge in the field of robotics. We are sure that there are better optimizations, but the implementation of our idea already gives a lot of useful things and we have encountered disadvantages in the number of components and devices for building a robot. Considering all the pros and cons, the project was successfully completed.

5.4 Xue Bai

The project was a new experience for us, as none of us had prior knowledge in handling hardware. Our group had to spend a lot of time figuring out how to put together the robot and make it run. It was also tough coordinating and finding time to work together as some of us were busy with work and classes. I had to be adaptable to changing requirements and work collaboratively with my team members. For that, I developed a range of skills, including problem-solving, teamwork and adaptability. Obtaining the required materials was a challenge, as we had to buy our own hardware part as it was missing from the kit. Additionally, I gained experience in the use of a specific algorithm, the Floyd Warshall algorithm, as well as knowledge in the field the project pertained to, such as lane/goal detection and direction navigation, and robot hardware.

Despite our efforts in making the lane/goal detection and direction navigation work in the software, implementing it into the robot was still a challenge. The camera's latency was not up to par with real-time processing, and this was partly due to our limited understanding of the hardware. Utilizing color sensors or other hardware could have helped resolve this issue, but time limitations and insufficient hardware parts provided us from doing so.

Despite these obstacles, we overcame them and completed the project successfully.

5.5 Liow Gian Hao

When I received the project requirements for this group assignment, it looked very intriguing to try out as our group selected the car as our main product. The project interests me as it combines elements of IOT, Algorithm theory and computer vision combined to run solve a real world problem which involves agricultural solutions.

Going through the project brought in numerous challenges, I had minimal prior experience with IOT devices and Raspberry Pi, and almost no experience with building a robot car with it. It was difficult to navigate how to assemble the entire car modules and camera modules and getting it to display the RaspberryPi OS on a monitor or local pc took time as well. Other than that, me and some of my groupmates were working full time as well, alongside the restriction that we are only able to work on the car at school at designated times, it made it difficult to work on the project together. We were also frequently missing parts and materials to progress, which took time for us to find the resources required.

Fortunately, we as a group managed to overcome these challenges. I learnt that to solve these challenges require a lot of research and trial and error for the technical side, and working as a group required compromise and adaptability. We did not managed to maximize the potential of our robot car due to hardware limitation, or time constraints that limited our ideas to fully optimize the algorithm and robot car, but I am proud of the work that was done as we have managed to successfully achieve what was initially required by our specifications.

6.0 Conclusion

In this project, we aimed to use the Floyd-warshall algorithm to find the shortest path and to reach the crops and eliminate damage from pests as fast as possible. We were successful in implementing this algorithm and applying it to the navigation of the robot.

Image processing was also used for real-time lane detection and direction navigation. However, due to hardware issues like camera overheating, causing latency and the power supply for the car is not appropriate for it to run at full speed, it is hard to make sure that the robot's processing is kept up with real-time and able to identify the path quickly. Although it is able to move in the correct general direction, it will still have some delays while making the navigation decisions.

In conclusion, we were successful in completing the project's overall objective, which is to implement the optimization algorithm inside the robot car. but were unable to fully improve on latency issues due to hardware limitations. In the future, we will continue to seek other solutions to achieve the goals of the project.