CZ3004-Multi-Disciplinary project

Log report 1(Design)

Submitted by

Team 2

Member 1: Aye Pwint Phyu Matric No. : U1920903F
Member 2: Lin Yan Matric No. : U1920925F
Member 3: Jesline Ng Matric No. : U1920213C
Member 4: Phone Myint Thu Mya Min Matric No. : U1920225E

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

NANYANG TECHNOLOGICAL UNIVERSITY

	Member	Section
1	Aye Pwint Phyu	Robot movement and Rpi communication
2	Lin Yan	Image recognition
3	Jesline Ng	Android development
4	Phone Myint Thu Mya Min	Algorithm

1) Robot movement and Rpi communication

Project Plan and Timeline

PROJECT TITLE	Multi-Disciplinary project
Team	2
Project Start:	Tue, 6/22/2021
Display Week:	1

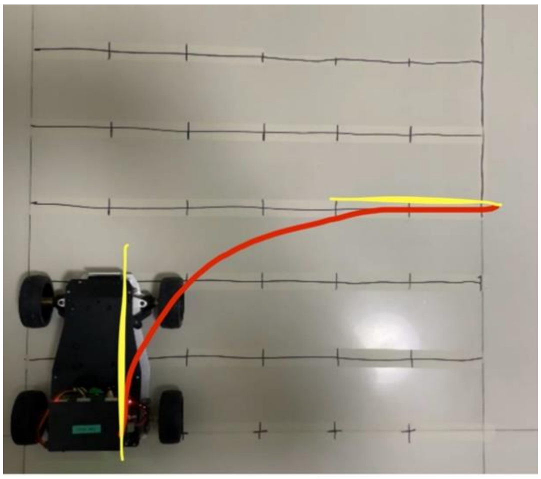
TASK	ASSIGNED TO	START	END
Set-up Rpi and Robot movement			
Installation and update Raspberian OS	Aye Pwint	6/22/21	6/23/21
Setup Rpi for serial-USB communication	Aye Pwint	6/24/21	6/27/21
Develop code for robot movement and Testing	Aye Pwint	6/28/21	7/5/21
Establishing Rpi communication	Aye Pwint		
Remote access over Wifi access	Aye Pwint	7/3/21	7/8/21
Communication with Android over bluetooth	Aye Pwint	7/9/21	7/12/21
Communication with Camera	Aye Pwint	7/12/21	7/13/21
Implementation			
Combine codes for movement, algo, image processing ar	All	7/13/21	7/15/21
Test out and modify codes for final run	All	7/16/21	7/22/21

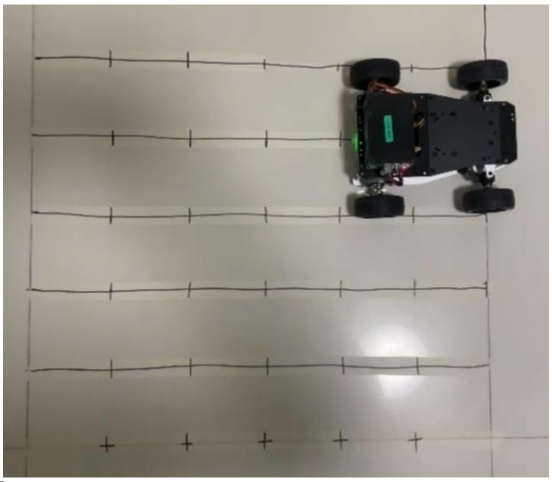
Research & Learning

For the robot movement, the two parts: Rpi for program control and SMT board needs to communicate through USB->Serial connection. For our project, we will be using python with Serial module. In addition to that we will be using time module to control robot movement as well.

The robot movement's focus was to establish communication with all other accessories such as Android and Camera and to minimize the manoeuvre distance around the map.

For successful communication with Android and Camera, the provided guideline as very helpful.

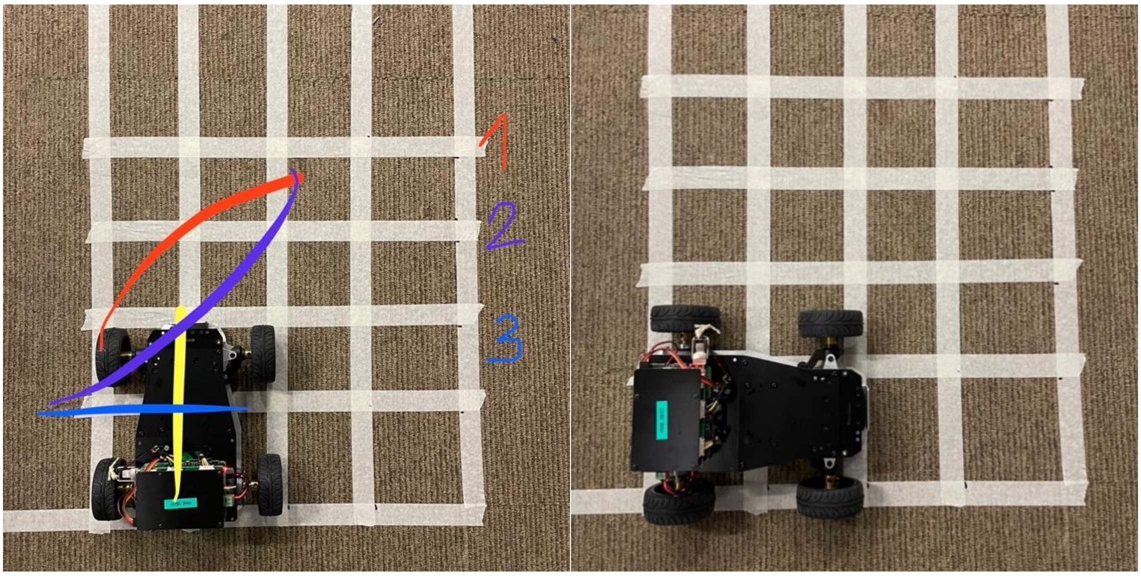




For the actual movement, one thing I have learnt is that turning angle creates the path as a circumference of the circle (max turning angle creates smallest circle achievable) and the robot will be moving along as a tangent. With four wheels robot, we will only have the robot turned desired angle after moving along the circumference. Let us say we want to turn right 180 degrees; the robot will then move from west side of the circle to east side of the circle. Speed only determines the time taken to complete the movement. I will be discussing more in the design strategies.

Design strategies

- <u>Straight line movement:</u> As simple as it sounds, I have chosen speed 130mm/sec to move 10cm. It may seem odd but through many verifications, I found out that the robot needs a few extra milliseconds to start up the motors to get going. So, if we use straight up 100mm/sec, it will not be able to move exactly 10cm after exactly 1sec. Hence it comes down to either increasing speed or time. I have chosen to increase speed as one of the incentives is to move faster.
- <u>Turning:</u> For simplicity, we have chosen to turn quarterly (i.e., 90/180/270/360 degrees). To achieve the turning with little manoeuvre distance and to have start and end point the same, I have chosen to go with 3 point turning approach. For the example of turning right 90 degrees:
  - Turn right with max negative angle and appropriate speed to move only 10cm all around the robot.
  - Reverse back with the same turning angle
  - Turn right with reduced angle to end up in the same spot



2) Image recognition

Project plan and timeline

PROJECT TITLE	Multi-Disciplinary Project		
Team	2		
oject Start:	Tue, 6/22/2021		
isplay Week:	1		

TASK	ASSIGNED TO	START	END
Problem Analysis			
Goal Setting	Lin Yan	6/22/21	6/23/21
Research and Learning	Lin Yan	6/23/21	6/27/21
Program Design			
Data Collection	Lin Yan	6/28/21	6/29/21
Program Development & Modification	Lin Yan	6/30/21	7/5/21
Testing & Optimizing	Lin Yan	7/6/21	7/12/21
Implementation in pi			
Combine codes for movement, algo, im	All	7/13/21	7/15/21
Test out and modify codes for final	All	7/16/21	7/22/21

Project Objective (part)

Spot recognition of target images by RPi camera.

Research & Learning

According to my research, there are two different ways to perform image recognition. (Due to time reasons, I only tried the following two methods. There are many other different image recognition methods on the Internet that will not be discussed here.)

- Image Contour Recognition --- Binary Image

The picture will be processed into a binary image to calculate its contour, and the picture will be recognized by comparing it with the target images' contour.

- Design defect: Binary images are susceptible to the effects of light, but the light cannot be controlled during on-site photography.

- Tensorflow Object Detection

Use Tensorflow Object Detection API for model training and perform object detection through the trained model.

- Design advantage: It can correctly identify the position and type of different objects in the picture in the natural environment.

Design strategies

According to research, the project will perform image recognition based on Tensorflow object detection, the steps as follow:

- The first is image data acquisition and pre-processing. Get as much relevant image data as possible and process them into the form required for model training.
- The next step is to implement image recognition. Choose a suitable deep learning model, use Tensorflow Object Detection to train the pre-processed image data set, and generate a model that can accurately detect and recognise the target image.
- After that is the deployment of the image recognition model. To deploy the trained model into the Raspberry Pi, and use camera to process the frame in real time to achieve real-time image recognition.
- The testing and optimizing are very important, the last step which is to combine the hardware and software with movement part, android part and algorithm part to do final testing and optimizing.

3) Android development

Project Plan and Timeline

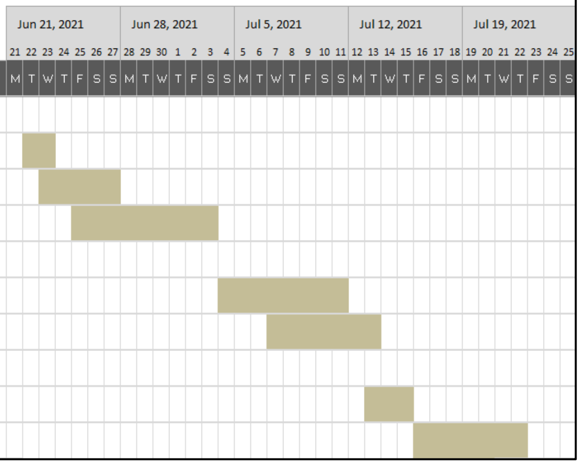
PROJECT TITLE Multi-Disciplinary project

Team 2

Project Start: Tue, 6/22/2021

Display Week: 1

TASK	ASSIGNED TO	START	END
Problem Analysis & Design			
Analyse the problem	Jesline	6/22/21	6/24/21
Study materials and do research	Jesline	6/23/21	6/27/21
Propose Algorithm & Design	Jesline	6/25/21	7/3/21
Implementation of Program			
Program Development	Jesline	7/4/21	7/11/21
Testing	Jesline	7/7/21	7/13/21
Implementation in pi			
Combine codes for movement, algo, image processing and android	All	7/13/21	7/15/21
Test out and modify codes for final run	All	7/16/21	7/22/21



Proposed Design & Tools

The Android application will need to be able to connect via Bluetooth to the robot and establish a two-way communication. The information that needed to be passed from the application to the robot are Starting Point, Obstacle Position and at time to allow to manually steer the robot. The android will receive status update and result from the image recognition.

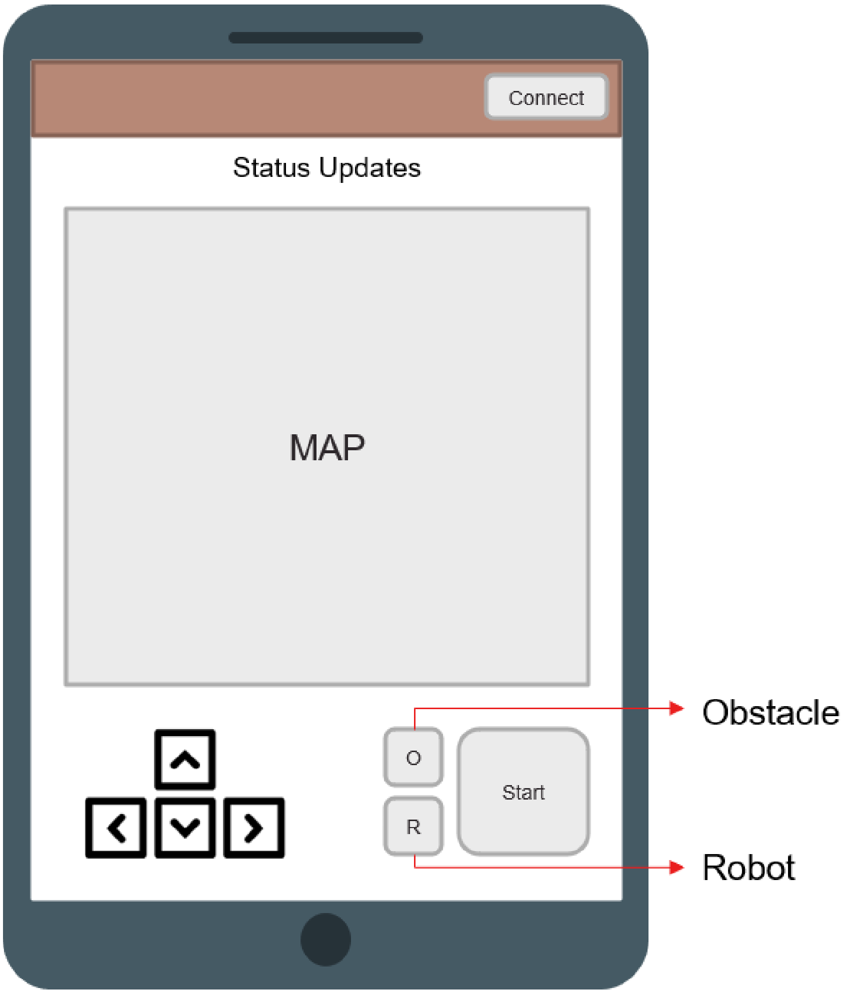
The application that will be used to develop the app is Android Studio V4.2.

- ### Bluetooth Connection
  The basic Bluetooth connection will be based upon the depository available in the Android Developer Guides. A handler will be added to parse the incoming messages and display only the necessary information.

  The messages will be displayed using text view on the top of the map.

- ### Map Area
  The map area will be populated using buttons. To indicate the starting point or the obstacle position, the user may drag the corresponding button to the map position. To remove this the user may click on the location of the obstacle in the map. The obstacle will be shown by change of button in the map area, while the robot will be shown using and image overlay on top of the map.

- Design of the UI
  The concept of the UI is as follow.



Design Strategies

Either single responsibility principle or open-closed principle will be used to implement the code. Single responsibility allow for easier implementation and any additional function can be easily added.

While open-closed principle will allow an easier addition of functionality with out modifying the original code. Using the open-closed principle can be a good choice as existing Bluetooth depository will be used.

4) Algorithm

Project plan and timeline

PROJECT TITLE Multi-Disciplinary project

Team 2

Project Start: Tue, 6/22/2021

Display Week: 1

TASK	ASSIGNED TO	START	END
Hamiltonian path			
Analyse the problem	Phone	6/22/21	6/24/21
Study materials and do research	Phone	6/25/21	6/28/21
Develop algorithm for shortest path	Phone	6/29/21	7/3/21
Path navigation			
Create flow chart for different scenarios	Phone	7/4/21	7/6/21
Write code for path navigation algorithm	Phone	7/7/21	7/10/21
Combine Hamiltonian path and path navigation	Phone	7/11/21	7/12/21
Implementation in pi			
Combine codes for movement, algo, image processing and android	All	7/13/21	7/15/21
Test out and modify codes for final run	All	7/16/21	7/22/21

Algorithms proposed for Hamiltonian path

Hamiltonian path is like the travel salesman problem. The difference is that in Hamiltonian path, there is no need to return to original vertex. There are a few methods to solve this problem.

1. The Brute-Force Approach
2. The Branch and Bound Method
3. The Nearest Neighbour Method

Among them, the nearest neighbour method is chosen since it is the simplest TSP heuristic and there are only 5 obstacles or vertexes. The coordinates of the obstacles can be received from android. First, transform them into travelling distances between vertexes or cost or edges. Second, create a graph to connect all vertexes with weighted edges. Then perform nearest neighbour algorithm to get Hamiltonian path.

```
nearestNeighbour()
{
        v = S; initialize Visited[j] = false for all locations.
        put v into List.
        repeat 5 times
        {
                select location w where cost[v][w] is minimum and Visited[w] is not true
                put w into List;
                Visited[w] = true;
                v = w;
        }
        return List;
}
```

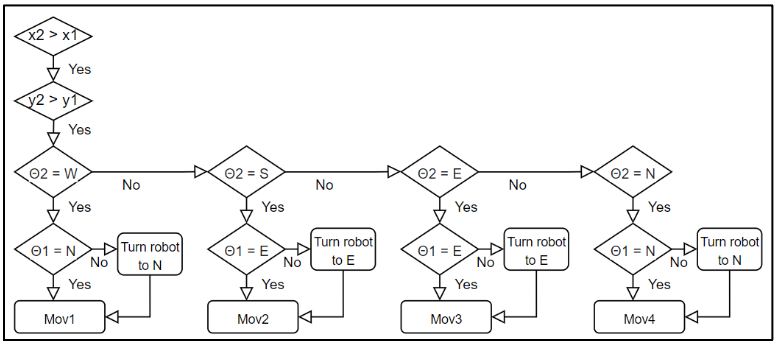
Pseudo code for the nearest neighbour method

Algorithms proposed for path navigation

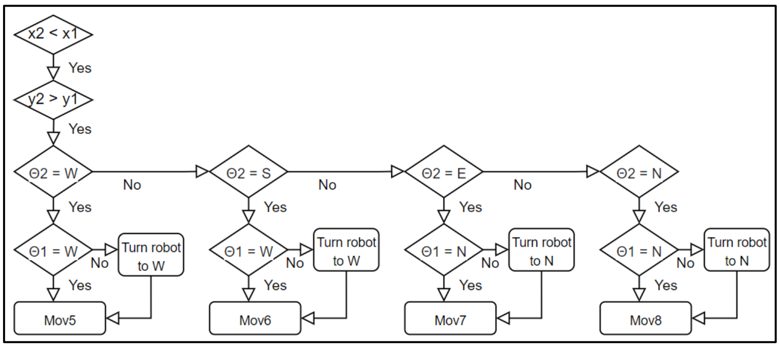
There are 4 possible locations for the obstacle from the robot. They are top right, top left, bottom right, bottom left of the robot.

There are 4 possible camera locations for robot and 4 image locations for obstacles as well such as north, east, south, and west. Therefore, there are total 64 scenarios for robot paths.

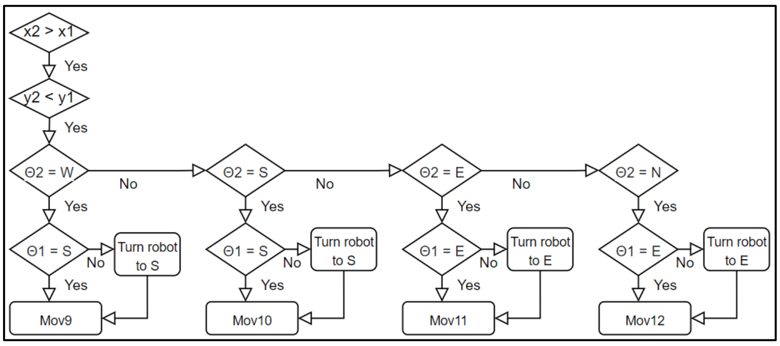
There will be only 16 movement functions for the robot since it can be repositioned back to desired direction.



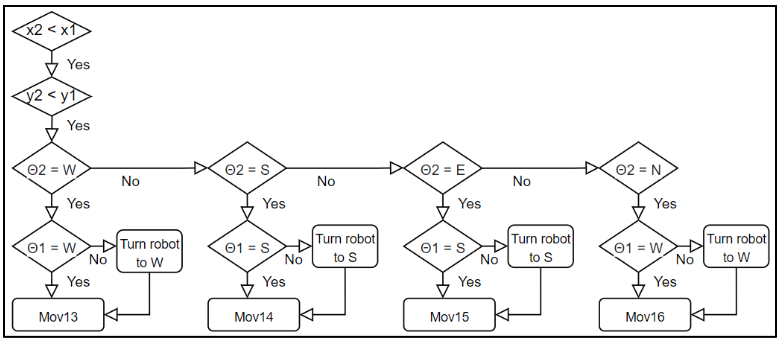
Obstacle at top right of the robot



Obstacle at top left of the robot



Obstacle at bottom right of the robot



Obstacle at bottom left of the robot

Design strategies

Low coupling and single responsibility principle will be used to implement the algorithms for Hamiltonian path and path navigation for easier extension and modifications.