

# McGILL UNIVERSITY

ECSE 211: Final Design Project

## Testing Document

*Yin Jun Huang*  
*(Charles)*

supervised by

Frank P. Ferrie

## 1 SUMMARY

This document is intended to indicate the objective and the methodology of testing of our project as well as discussing the result of testing.

## 2 TASKS

**Project:** Design Principles and Methods Final Project: Capture the Flag - Team 15

**Task:** Design an autonomous robot that can play a one-on-one version of the Capture the Flag while navigating through an obstacle course.

## CONTENTS

<b>1 SUMMARY .....</b>	<b>2</b>
<b>2 TASKS .....</b>	<b>2</b>
<b>3 EDIT HISTORY .....</b>	<b>3</b>
DOCUMENT VERSION NUMBER .....	3
EDIT HISTORY .....	3
<b>4 HARDWARE TESTING .....</b>	<b>4</b>
4.1 TEST 1: LARGE MOTOR VS NXT MOTOR .....	4
4.2 TEST 2: TRAVERSING ZIP LINE (DISMOUNTING ANGLE) .....	5
4.3 TEST 3: TRAVERSING ZIP LINE (MOUNTING ANGLE) .....	6
4.4 TEST 4: TRAVERSING ZIP LINE 2 (MOUNTING ANGLE).....	7
<b>5 SENSOR CHARACTERIZATION TESTING .....</b>	<b>9</b>
5.1 TEST 5: TESTS OF THE DETECTION RANGE OF ULTRASONIC SENSOR .....	9
5.2 TEST 6: TESTS OF ACCURACY WITH ULTRASONIC SENSOR.....	10
5.3 TEST 7: GRIDLINE DETECTION TESTS WITH LIGHT SENSOR .....	11
5.4 TEST 8: COLOR BLOCK DETECTION TESTS WITH LIGHT SENSOR .....	12
5.5 TEST 9: COLOR BLOCK DETECTION TESTS WITH LIGHT SENSOR WITH COLOR ID MODE .....	14
5.6 TEST 10: ULTRASONIC SENSOR BEHAVIOR TEST .....	17
<b>6. UNIT TEST .....</b>	<b>19</b>
6.1 TEST 11: ULTRASONIC LOCALIZATION TEST .....	19
6.2 TEST 12: LIGHT SENSOR LOCALIZATION TEST .....	20
6.3 TEST 13: NAVIGATION TEST .....	21
6.4 TEST 14: ODOMETRY TEST .....	22
6.5 TEST 15: BLOCK DETECTION (SEARCHING FOR THE FLAG) .....	23
6.6 TEST 16: SEARCH FOR OPPONENT'S FLAG .....	26
6.7 TEST 17: CROSS THE RIVER.....	27
<b>7. INTEGRATION TEST.....</b>	<b>28</b>
7.1 TEST 18: NAVIGATE TO THE ZIP LINE (POINT 1).....	28
7.2 TEST 19: NAVIGATE TO THE ZIP LINE (POINT 2).....	30
7.3 TEST 20: ZIP LINE TRAVERSING (MOUNT, CROSS AND DISMOUNT).....	31
7.4 TEST 21: NAVIGATE TO THE OPPONENT'S FLAG LOCATION I (THE ZIP LINE IS PARALLEL TO THE X- AXIS).....	33
7.5 TEST 22: NAVIGATE TO THE OPPONENT'S FLAG LOCATION II (THE ZIP LINE IS DIAGONAL).....	35

7.6	TEST 23: SEARCH THE OPPONENT'S FLAG.....	37
<b>8</b>	<b>SYSTEM TESTS.....</b>	<b>38</b>
8.1	TEST 24: FINAL TEST I (8 X 8 TILES) .....	38
8.2	TEST 25: FINAL TEST II (12X 12 TILES) .....	40
<b>9</b>	<b>REFERENCE .....</b>	<b>42</b>

### 3 EDIT HISTORY

#### Document Version Number

- 0.0.1: Version presented to Prof. Ferrie on the 2017/10/27
- 0.0.2: Version presented to Prof Ferrie on the 2017/11/03
- 0.0.3: Version presented to Prof Ferrie on the 2017/11/10
- 0.0.4: Version presented to Prof Ferrie on the 2017/11/17
- 0.0.5: Version presented to Prof Ferrie on the 2017/11/24
- 0.0.6: Final Submission on the 2017/12/01

#### Edit History

1. Charles Huang, 2017/10/25 (4.30pm) - Created the document, designed the tests of the project
2. Aliah Mohd Nazarudin, 2017/10/26 (1230pm) - Change the format of the document
3. Charles Huang, 2017/10/29 (2.30pm) - Update hardware and software testing
4. Aliah Mohd Nazarudin, 2017/11/2 (1.50pm) - Add conclusions and further actions taken for every test
5. Charles Huang, 2017/11/02 (2.30pm) - Update colored block testing
6. Aliah Mohd Nazarudin, 2017/11/08 (5.12pm) - Mounting the zip line test
7. Charles Huang, 2017/11/09 (12.15pm) - Unit Test and Integration Test
8. Aliah Mohd Nazarudin, 2017/11/09 (3.00pm) - Update the content list
9. Charles Huang, 2017/11/16 (3.00pm) – Update the integration test
10. Charles Huang, 2017/11/23 (1.00pm) – Update the integration test
11. Aliah Mohd Nazarudin, 2017/11/23 (2.30pm) – Update the format of the document
12. Charles Huang, 2017/11/29 (11.00am) – Add Test Map figures to some of the tests
13. Aliah Mohd Nazarudin, 2017/11/29 (7.05pm) – Finalize the format of the document

## 4 HARDWARE TESTING

### 4.1 Test 1: Large Motor vs NXT Motor

Hardware Version: 0.0.1

Software Version: Lab 4 code

Date: October 20 to 23

**Objective:** The LeJOS library provides several motor classes for controlling the EV3 motor, Large Motor and NXT Motor. This experiment was conducted with the hope of discovering which motor performed better for use in the navigation.

#### **Methodology:**

The robot will be made to navigate from a distance of 6 tiles using the Large Motor first and then the NXT motor. Ten trial runs will be conducted. The robot will be placed at origin (0,0) and navigate to (0,6). During the experiment, record the behavior of two motors as well as the final angle error.

#### **Results:**

**Table 1: Euclidean distance error for Large Motor and NXT Motor**

<b>Trial No.</b>	<b>Euclidean distance error of Large Motor(cm)</b>	<b>Euclidean distance error of NXT Motor(cm)</b>
1	2.5	4.3
2	2.5	4.7
3	1.6	3.5
4	1.9	3.6
5	2.4	4.6
6	1	3.5
7	2.9	3.8
8	1.9	3.4
9	1.5	3.2
10	1.6	3.7
<b>Average</b>	<b>2.15</b>	<b>3.83</b>

**Conclusion & Action:** The results showed that the Large Motor performed better than NXT Motor. The robot with Large Motors is able to go straight with a final angle error smaller than 2 degrees. The NXT Motor, unlike the Large Motor, even though the speed of the left motor and right motor are set to same, the robot with NXT Motor is not able to go straight. The left motor of NXT Motor is always slow than the right Motor. Since the Large Motor performed better than NXT Motor. Our team decide to use Large Motor for the wheel of the robot.

#### 4.2 Test 2: Traversing Zip Line (Dismounting Angle)

Hardware Version: 0.0.1

Software Version: Lab 5 code

Date: October 20 to 23

**Objective:** This experiment was conducted with the hope of discovering what is dismounting angle when the robot dismount from the zip line.

**Methodology:**

The robot will be made to travel zip line by using Lab 5 code. Five trial runs will be conducted. We will measure the dismounting angle after the robot dismount from the zip line.

**Results:**

**Table 2: Dismounting angle error**

<b>Trial No.</b>	<b>Dismounting angle error(degree)</b>
1	23
2	30
3	34
4	20
5	27
<b>Average</b>	<b>26.8</b>

**Conclusion & Action:**

The results showed that our robot is not balanced during the zip line traveling. The dismounting angle is always large 20 degrees. The left wheel always touches the board first and the right wheel touches the board after when the robot dismounts from the zip line. It makes the dismount angle large.

The robot is not balanced during the zip line traveling which makes the dismount angle large. This test report should be send to the hardware team to review the hardware design of the robot especially the balance of the robot during the zip line traveling.

#### 4.3 Test 3: Traversing Zip Line (Mounting Angle)

Hardware Version: 0.0.1

Software Version: Lab 5 code

Date: Nov 3 to 9

**Objective:** This experiment was conducted to discover the angle that allows the robot to perfectly mount the zip line.

**Methodology:**

The robot will be made to travel zip line by using Lab 5 code. Ten trial runs will be conducted. We will measure the mounting angle, or the angle error that is allowed for the robot to successfully mount the zip line.

**Results:**

Based on Table 1, it shows that the heading orientation of the robot needs to be between 0 degree to 2-degree angle error for it to successfully mount the zip line.

**Table 3: Mounting the zip line**

Trial No.	Angle Error (deg)	Status
1	0.0	SUCCESSFUL
2	1.0	SUCCESSFUL
3	2.0	SUCCESSFUL
4	3.0	UNSUCCESSFUL
5	2.3	UNSUCCESSFUL
6	2.8	UNSUCCESSFUL
7	2.4	UNSUCCESSFUL
8	2.5	UNSUCCESSFUL
9	2.6	UNSUCCESSFUL
10	2.7	UNSUCCESSFUL

**Conclusion & Action:** The robot needs to have a perfect heading orientation to mount the zip line. This test report is sent to the hardware team to review the hardware design. The robot should be able to correct itself as it mounts the zip line.

#### 4.4 Test 4: Traversing Zip Line 2 (Mounting Angle)

Hardware Version: 0.0.3

Software Version: 0.0.1

Date: November 3 to 9

**Objective:** This experiment was conducted to discover the angle and position that allows the robot to perfectly mount the zip line with Hardware Version 03(attach the filter funnel and the round, plastic container lid to the pulley).

**Methodology:**

The robot will be placed on the different angle and different position from the zip line. The test time will record the range of position and angle that allow the robot to properly mount the zip line.

**Result:**

**Table 4: Mounting angle data after improving hardware design**

Trial No.	Angle Error (deg)	Status
1	-15	UNSUCCESSFUL
2	-12	UNSUCCESSFUL
3	-10	SUCCESSFUL
4	-5	SUCCESSFUL
5	0	SUCCESSFUL
6	5	SUCCESSFUL
7	10	SUCCESSFUL
8	15	SUCCESSFUL

9	20	SUCCESSFUL
10	25	SUCCESSFUL
11	28	SUCCESSFUL
12	30	SUCCESSFUL
13	33	UNSUCCESSFUL
14	35	UNSUCCESSFUL

**Table 5: Mounting position data after improving hardware design**

<b>Trial No.</b>	<b>Position error (cm)</b>	<b>Status</b>
1	-5.0	UNSUCCESSFUL
2	-4.0	UNSUCCESSFUL
3	-3.0	SUCCESSFUL
4	-2.5	SUCCESSFUL
5	-2.0	SUCCESSFUL
6	-1.0	SUCCESSFUL
7	0	SUCCESSFUL
8	1.0	SUCCESSFUL
9	1.5	SUCCESSFUL
10	1.7	SUCCESSFUL



11	2.0	SUCCESSFUL
12	2.3	UNSUCCESSFUL
13	2.5	UNSUCCESSFUL
14	3	UNSUCCESSFUL

### **Conclusion & Action:**

From the Table 5 and Table 6, if the range of the angle error right before mounting zip line is around  $[-10, 30]$  degrees, the robot can successfully mount to zip line. If the range of the position error right before mounting zip line is approximately  $[-3, 2]$  cm, the robot can successfully mount to zip line. The result will send to hardware team and software team to improve the hardware design and zip line traveling code, respectively.

## **5 SENSOR CHARACTERIZATION TESTING**

### **5.1 Test 5: Tests of the detection range of ultrasonic sensor**

Hardware Version: 0.0.1

Software Version: Testing code 01

Date: October 27 to 30

**Objective:** The objective of the tests is to determine the detection range of ultrasonic sensor.

### **Methodology:**

1. Write a program to fetch the data from the robot.
2. Place the robot at a fix position.
3. Place a block perpendicular to the ultrasonic sensor at certain distance, however, the ultrasonic sensor should not detect the block
4. Move the block slowly and perpendicular to the ultrasonic sensor
5. Record the angle when the ultrasonic sensor detects the wall.
6. Repeat steps 3-5 with different block starting position

**Results:**

The result shows that the detection range of ultrasonic sensor is like a cone. The ultrasonic sensor is able to detect a block provided the block surface is facing the robot within 10 degrees.

**Conclusion & Action:** The result shows that if the block is within 10-degree angle range to where the ultrasonic sensor is facing, the ultrasonic sensor can detect the block almost accurately. If the block is placed perpendicular to the sensor, the values returned by the sensors are nearly flawless. This test report should be sent to the software team so they can write the code accordingly.

## 5.2 Test 6: Tests of accuracy with ultrasonic sensor

Hardware Version: 0.0.1

Software Version: Testing code 02

Date: October 27 to 30

**Objective:** The objective of the tests is to test the accuracy of three ultrasonic sensors and determine which ultrasonic sensor is most accurate so that we can use it for the final project.

**Methodology:**

1. Write a program to get ultrasonic sensor data from three different ultrasonic sensors.
2. Mount three different ultrasonic sensors in parallel on the robot.
3. Place a block 5 cm in front of the ultrasonic sensors.
4. Record each ultrasonic sensor's data from the program
5. Increase the distance between the ultrasonic sensors and the block to 10 cm
6. Record new data from each ultrasonic sensor
7. Repeat steps 5 and 6 and increase the distance between the ultrasonic sensors to 15 cm

**Results:**

**Table 6: Reading of ultrasonic sensor**

Expected reading(cm)	Actual reading of Ultrasonic sensor 1 (cm)	Actual reading of Ultrasonic sensor 2 (cm)	Actual reading of Ultrasonic sensor 3 (cm)
5	5	4.9	4.9
10	10	9.9	10.1
15	15	14.8	14.9

**Conclusion & Action:** Different sensors tend to have different sensitivity level. Based on this experiment, ultrasonic sensor 1 is the most sensitive sensor we have, so the hardware of the robot is changed by placing ultrasonic sensor 1 to the robot.

### 5.3 Test 7: Gridline detection tests with light sensor

Hardware Version: 0.0.1

Software Version: Testing code 03

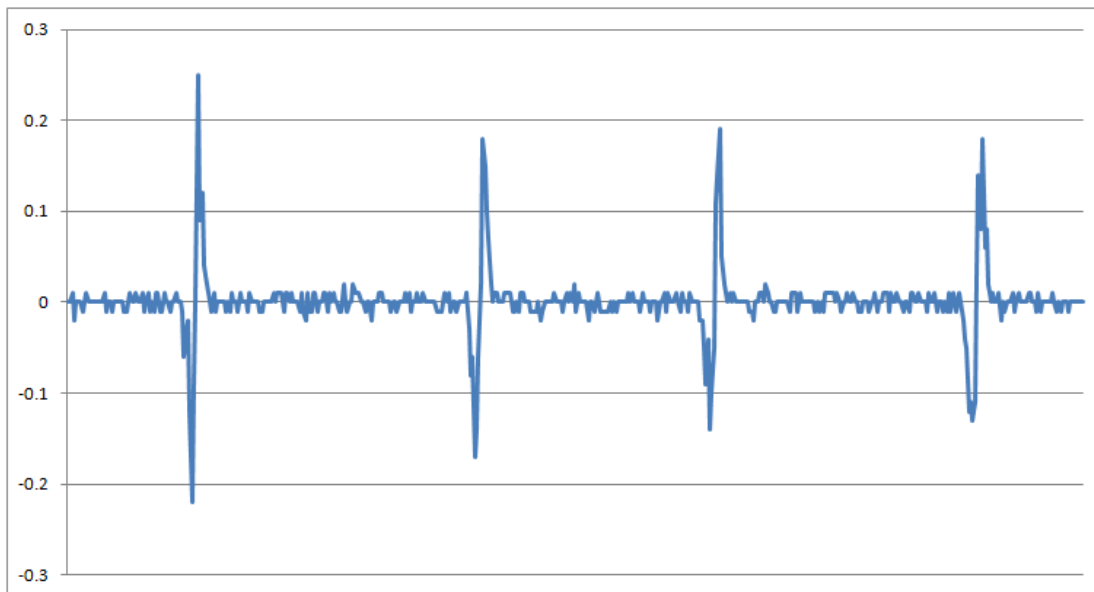
Date: October 27 to 30

**Objective:** The objective of the tests is to determine the changes of light intensity detected by light sensor when it detect the gridline.

**Methodology:**

1. Write a program to rotate the robot 360 degree and record the data from the light sensor.
2. The robot will be place on an open field.
3. The center of rotation of the robot will be place at an intersection of grid line.
4. The robot will rotate 360 degrees while continuously recording the value detected by the light sensor after a simple differential filter is applied.

**Results:**



**Figure 1: Gridline detection test data**

**Conclusion & Action:** The differential filter used in this experiment shows that when the result is more than 0.1, it shows that the sensor is detecting the black line. This result is then being informed to the software team, for them to write the code accordingly.

#### 5.4 Test 8: Color block detection tests with light sensor

Hardware Version: 0.0.1

Software Version: Testing code 04

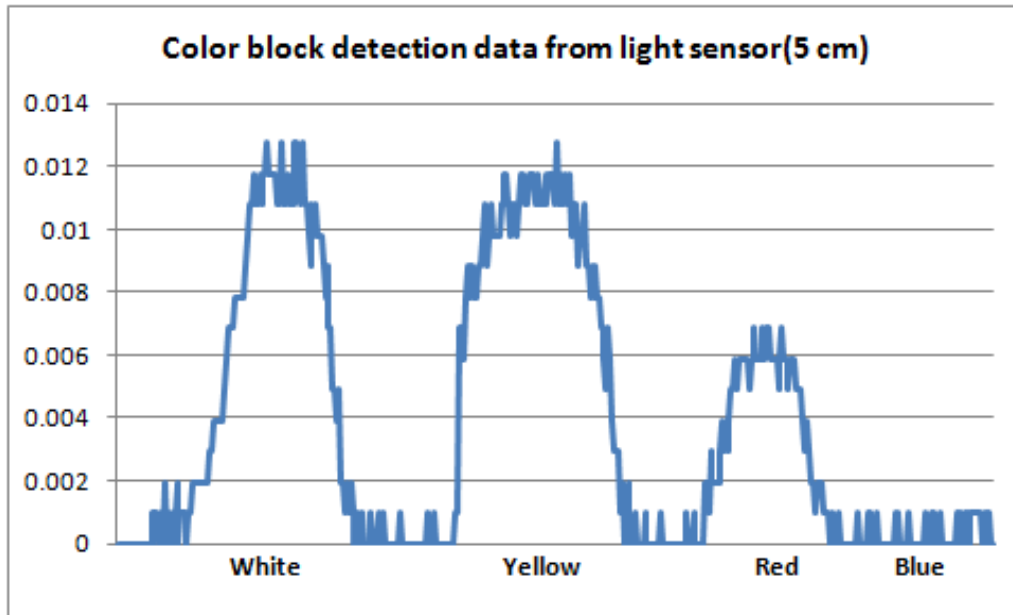
Date: October 27 to 30

**Objective:** The objective of the tests is to determine the changes of light intensity detected by light sensor when it detect different colored blocks at different distances.

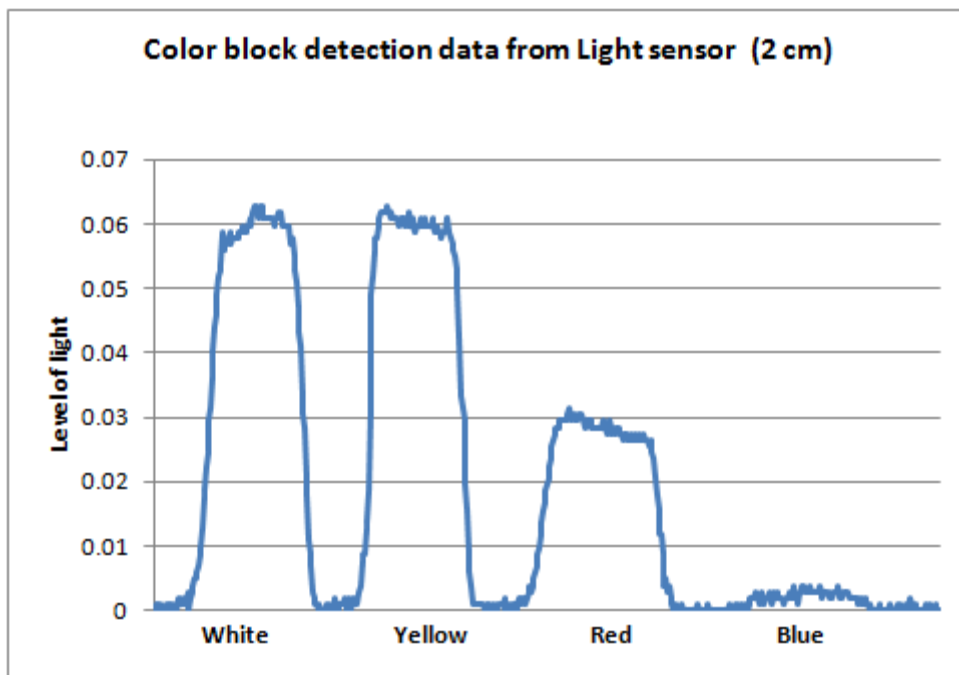
**Methodology:** For the purpose of testing, the light sensor is placed at the left side of the robot. The test robot will be placed on an open field. The robot is programmed to pass by the four blocks.

1. The light sensor mode is set to RGB mode.
2. The four different colored blocks are arranged at the left side of the robot.
3. The order of the colored blocks is white, yellow, red and blue.
4. The distance between the blocks is made constant, which is 10 cm away from each other.
5. The distance between the light sensor and the colored blocks is set to 5 cm.
6. The experiment is repeated by setting the distance between light sensor and the colored blocks to 2 cm.

**Results:**



**Figure 2: Color block detection data from light sensor with 5 cm difference from the light sensor and the colored block**



**Figure 3: Color block detection data from light sensor with 5 cm difference from the light sensor and the colored block**

**Conclusion & Action:**

Figure 2 shows that the level of light changes according to different colored blocks. Since the blue colored block can absorb the most light, therefore, the level of light does not really change when the light sensor detects it.

When the light sensor detects the white colored block and yellow colored block, the behavior of the change of light intensity is similar. The test team may need to do more test for these two-colored blocks.

According to the Figure 2 and Figure 3, the behavior of change of light intensity is similar for different colored blocks. However, the value of light intensity for different colored blocks are different when the distance between the light sensor and the blocks differ. The result shows that the light sensor is more accurate when the its distance with the blocks are minimal.

More tests need to be done by varying the distances between light sensor and the blocks to get most accurate color intensity. This result is then informed to the software team, for them to write colored block detection code accordingly.

## 5.5 Test 9: Color block detection tests with light sensor with Color ID mode

Hardware Version: 0.0.1

Software Version: Testing code 04

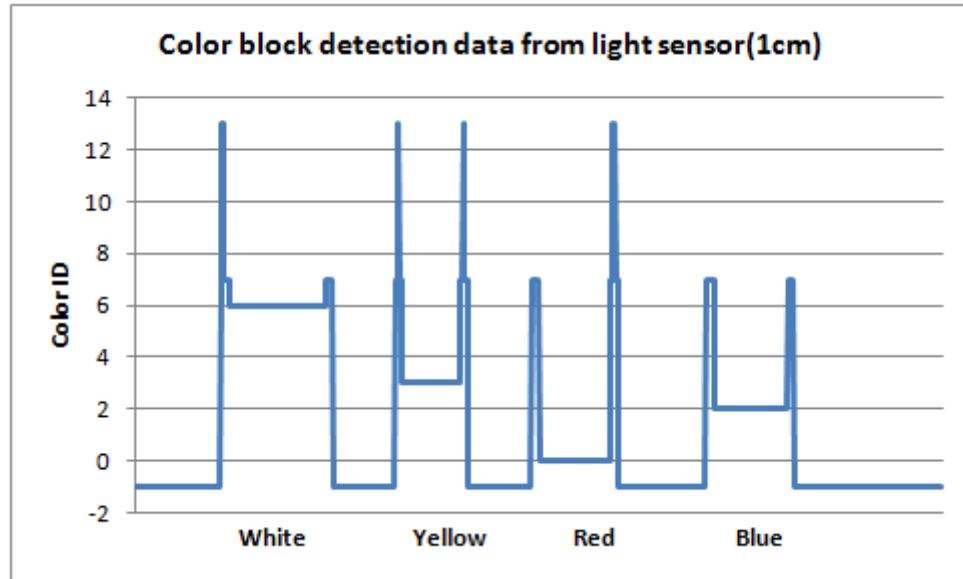
Date: November 3 to 9

**Objective:** The objective of the tests is to determine the Color ID detected by light sensor when it detect different colored blocks at different distances.

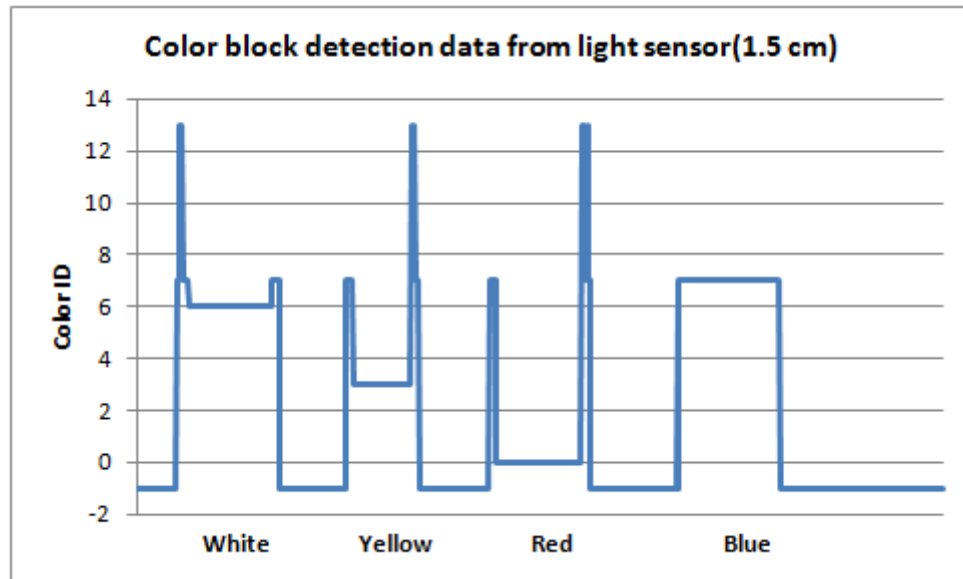
**Methodology:** For the purpose of testing, the light sensor is placed at the left side of the robot. The test robot will be placed on an open field. The robot is programmed to pass by the four blocks.

1. The light sensor mode is set to Color ID mode.
2. The four different colored blocks are arranged at the left side of the robot.
3. The order of the colored blocks is white, yellow, red and blue.
4. The distance between the blocks is made constant, which is 10 cm away from each other.
5. The distance between the light sensor and the colored blocks is set to 1 cm.
6. The experiment is repeated by setting the distance between light sensor and the colored blocks to 1.5 cm and 2 cm.

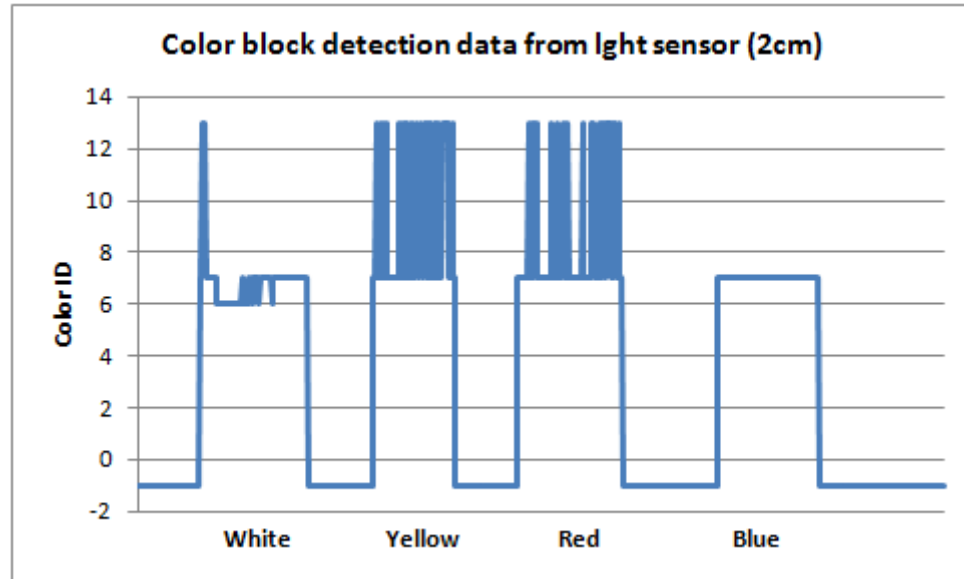
**Result:**



**Figure 4: Color block detection data from light sensor with 1 cm difference from the light sensor and the colored block**



**Figure 5: Color block detection data from light sensor with 1.5 cm difference from the light sensor and the colored block**



**Figure 6: Color block detection data from light sensor with 2 cm difference from the light sensor and the colored block**

#### **Conclusion & Action:**

From the instruction of Color ID sensor mode, the Color ID of white is 6, the Color ID of yellow is 3, the Color ID of red is 5 and the Color ID of blue is 2.

From the Figure 4, the sensor is able to detect different color block with respect Color ID. However, the Color ID of red that the sensor detected is 0 instead of 5. As the distance between the sensor and the block increases, the data becomes less accurate. When the distance is 1.5 cm, the sensor receives a different Color ID for blue. When the distance is 2 cm, the sensor is not able to distinguish different color blocks. In addition to that, the noise may influence the reading from the light sensor. This result is then informed to the software team, for them to write colored block detection code accordingly.



## 5.6 Test 10: Ultrasonic sensor behavior Test

Hardware Version: 0.0.1

Software Version: Testing code 05

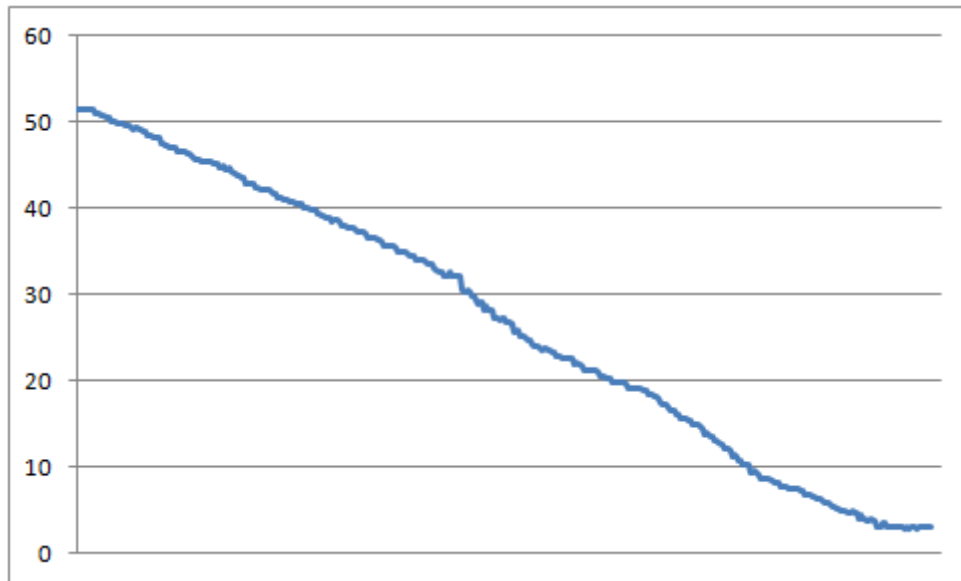
Date: November 3 to 9

**Objective:** The objective of the test is to test the behavior of ultrasonic sensor when it detects a block.

### Methodology:

1. The robot is placed on open field
2. Place a foam block inclined at 0 degrees from the robot.
3. Move the robot towards the block
4. Record data from ultrasonic sensor
5. Repeat the experiment with the block inclined at 15 degrees and 45 degrees from the robot.

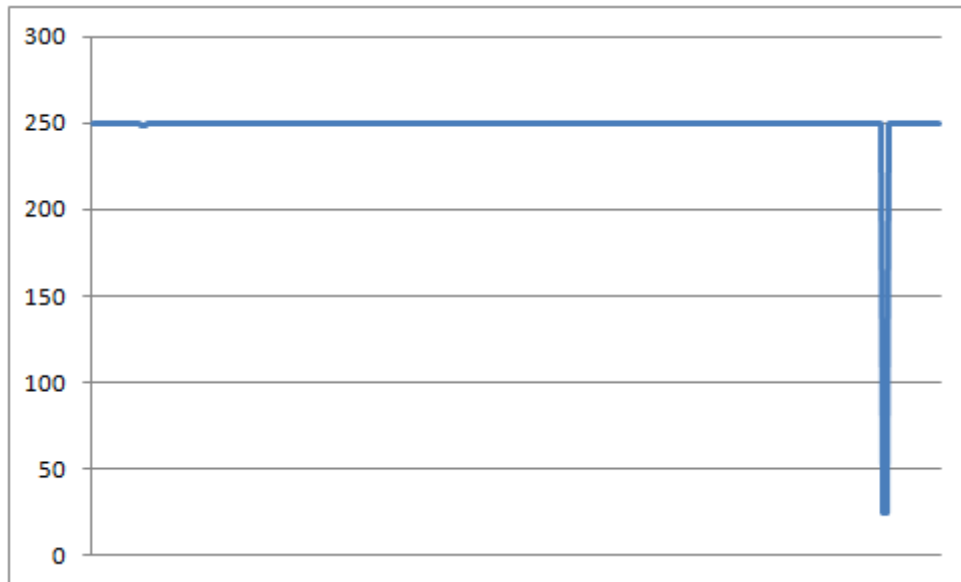
### Result:



**Figure 7: Ultrasonic data when the block is facing the robot (0 degrees)**



**Figure 8: Ultrasonic data when the block is facing the robot (15 degrees)**



**Figure 9: Ultrasonic data when the block is facing the robot (45 degrees)**

**Conclusion & Action:** From the experiment, the ultrasonic sensor is able to detect a block that is facing the robot within the range of 10 degrees. This show that the ultrasonic sensor can only receive a few values to show the existence of the block if its line of sight is not perpendicular to the surface of the block. However, if the line of sight of ultrasonic sensor is perpendicular to the surface of the block, the data from ultrasonic sensor is almost accurate.

## 6. UNIT TEST

### 6.1 Test 11: Ultrasonic localization test

Hardware Version: 0.0.2

Software Version: 0.0.1

Date: November 3 to 9

**Objective:** Since we implement both falling edge and raising edge ultrasonic localization, in this experiment, we will test both ultrasonic localization and note their accuracy and behaviors.

#### **Methodology:**

In this experiment, test team will place the robot at a random corner on the board. Then, the robot will run falling edge ultrasonic localization code and raising edge ultrasonic localization code respectively. The test team will note the behavior of the robot and record the final angle error according to the origin after localizing. The test will run 10 trials for each ultrasonic localization. So, total of 20 trials will be conducted.

#### **Result:**

**Table 7: Ultrasonic sensor localization data**

<b>Trial No.</b>	<b>Euclidean angle error of falling edge (degree)</b>	<b>Euclidean angle error of raising edge (degree)</b>
1	0	5
2	0	4
3	2	6
4	2	5
5	1	4
6	0	Failed
7	1	8
8	2	7
9	2	6
10	0	9
Average	1	6

**Conclusion & Action:**

By comparing two ultrasonic localizations, the falling edge ultrasonic localization has better behavior and accuracy. The average of final angle error of falling edge ultrasonic localization is less than 2 degrees. Unlike the falling edge localization, the raising edge localization has an average of final angle error more than 3 degrees. Moreover, during the 10 trials of falling edge ultrasonic localization, all trials are successful, which means that the robot can successfully detect the wall and turn back to origin. However, during the 10 trials of raising edge ultrasonic localization, 9 trials succeed and 1 trial failed. The reason of failure is that the robot detected one wall twice. As a consequence, the robot is not able to turn back to origin for that trial. The result will send to software team to improve the code.

**6.2 Test 12: Light sensor localization test**

Hardware Version: 0.0.2

Software Version: 0.0.1

Date: November 3 to 9

**Objective:** The objective of this test is to test the accuracy of light localization.

**Methodology:**

In this experiment, test team will place the robot at along the 45 degrees line face on the 0 degrees corresponding to origin. Then, the robot will run light localization. We will note the behavior of light sensor localization and the final angle error between the odometer reading and the robot actual angle. Ten trials will be conducted.

**Result:**

**Table 8: Light sensor localization data**

<b>Trial No.</b>	<b>Angle error (degree)</b>
1	2.5
2	1.7
3	1.3
4	2.8
5	1.5
6	1.0
7	2.7

8	Failed
9	1.7
10	2.6
Average	2.0

### **Conclusion & Action:**

From ten trials of experiment, the behavior of the light sensor localization is not ideal. There is one trial that the robot is not able to detect the last black line. Thus, the robot can not sometimes localize properly. However, if the robot detects all four lines, the final angle error between the odometer reading and the robot actual angle is less than 3 degrees. The result will send to software team to improve light sensor localization code.

### **6.3 Test 13: Navigation test**

Hardware Version: 0.0.2

Software Version: 0.0.1

Date: November 3 to 9

**Objective:** The objective of this test is to test the accuracy of navigation.

### **Methodology:**

In this experiment, test team will place the robot at a random coordinate which means the interaction of black lines. The robot is placed facing to 0 degrees. Then, the robot will be asked to navigate at random coordinate. We will note the behavior of navigation and record the final position error between the robot actual position and destination coordinate and final position. Ten trials will be conducted.

**Result:****Table 9: Navigation test data**

Trail No.	Destination point	Euclidean distance error(cm)
1	(2,2)	1.9
2	(0,4)	2.0
3	(6,6)	4.8
4	(0,6)	3.2
5	(4,6)	3.9
6	(1,2)	1
7	(3,4)	2.8
8	(3,6)	3.4
9	(2,5)	2.8
10	(3,3)	3.7

**Conclusion & Action:**

From ten trials of experiment, the behavior of navigation is ideal. The robot is able to turn properly and successfully navigate to the destination coordinate. The final position error depends on the battery and the distance of navigation. If the battery is low, the final position error will increase. Moreover, if navigation distance increases, the final position error will increase too. The average final position error is less than 5 cm. The result will be sent to software team to improve the navigation code.

**6.4 Test 14: Odometry test**

Hardware Version: 0.0.2

Software Version: 0.0.1

Date: November 3 to 9

**Objective:** The objective of this test is to test the accuracy of odometer.

**Methodology:**

In this experiment, testing team will place at origin (0,0) and make the robot is placed to face 0 degrees. Then, the robot will be asked to navigate to random coordinate. The testing team will record the Euclidean distance error between the robot actual position, odometer's reading and the angle error between the robot actual theta and odometer's reading. Ten trials will be conducted.

**Result:****Table 10: Odometry Test Data**

Trail No.	Destination point	Euclidean distance error(cm)	Euclidean angle error(degree)
1	(2,2)	0.5	0.2
2	(4,4)	0.6	0.3
3	(6,6)	0.7	0.1
4	(2,6)	0	0.3
5	(4,2)	0.2	0.5
6	(1,1)	0	0.4
7	(3,4)	0.6	0.2
8	(2,5)	0.8	0.3
9	(2,3)	0.3	0.5
10	(1,3)	0.3	0.4

**Conclusion & Action:**

From ten trials of experiment, the odometer is very accurate. The Euclidean distance error between the actual position and odometer's reading is always smaller than 1 cm and the angle error between the robot's actual theta and the odometer's reading is less than 0.5 degree. The result will send to software team.

**6.5 Test 15: Block Detection (Searching for the flag)**

Hardware Version: 0.0.3

Software Version: 0.0.1

Date: November 15 to 22

**Objective:** The objective of this test is to test whether the robot can detect different block within the searching area by using light sensor.

**Methodology:**

In this experiment, we set our team as green team and the size of the board is 8 x 8 tiles. The robot will be placed at lower left-hand corner of search region in red player zone (6,2). The robot will receive the information including the coordinate of lower left-hand

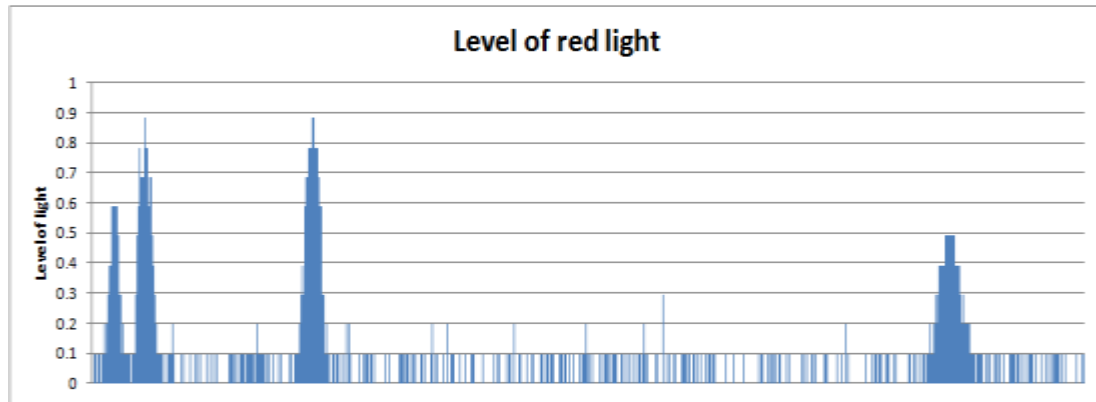
corner of search region in red play zone (5,2) and upper right-hand corner of search region in red player zone (7,4), etc. Therefore, the searching region is 2 x 2 tiles. Two different color blocks will be placed within the search region with 1 cm of distance from the robot. Then, the robot will be asked to circle around the search region. Once the robot detects a block, it will beep. The testing team will record whether the robot beeps when the light sensor passes by a block and record the light sensor data of each circle. Ten trials will be conducted. For each trial, the test team will replace the color blocks so that the distance between light sensor and the blocks will increase by 1 cm.

**Result:**

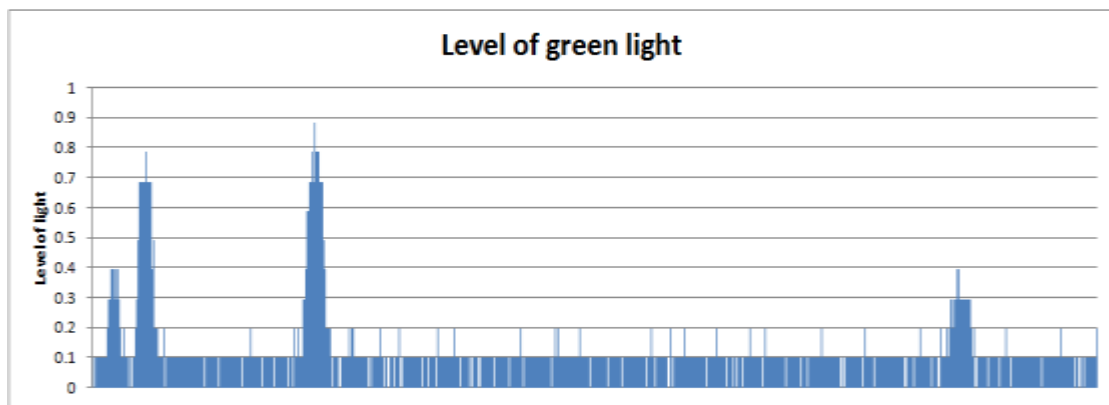
**Table 11: Block detection test data**

<b>Distance between the block and the light sensor (cm)</b>	<b>Status</b>
1	Detected
2	Detected
3	Detected
4	Detected
5	Detected
6	Detected
7	Detected
8	Undetected
9	Undetected
10	Undetected

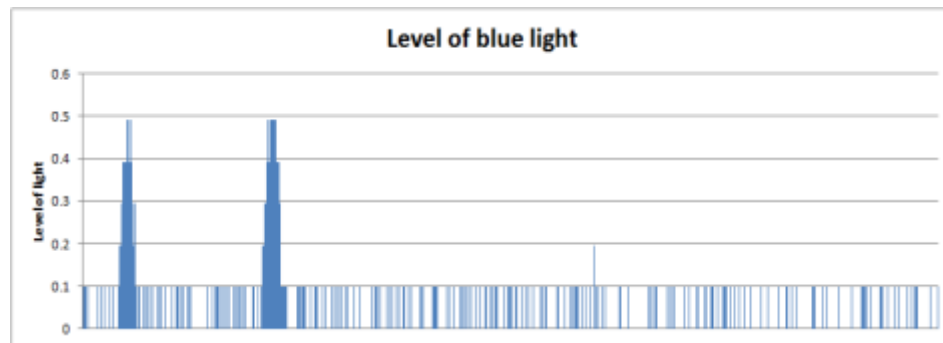




**Figure 10: Red light level data from light sensor for the fifth trial**



**Figure 11: Red light level data from light sensor for the fifth trial**



**Figure 12: Blue light level data from light sensor for the fifth trial**

### **Conclusion & Action:**

Since RGB mode of light sensor measures the level of red, green and blue light, therefore, it gives 3 data every time it detects a block. From the Figure 10, Figure 11 and Figure 12, when the light sensor detects a block, the level of red, green and blue light obviously increases. Therefore, we can conclude if level of one of three light is larger than 0.1, it means the light sensor detect a block. However, from the Table 8, the robot is only able to

detect a block within distance of 8 cm between the light sensor and the block. If the distance between the block and the light sensor is more than 8 cm, the robot cannot detect the block anymore and the level of lights from the light sensor will always less than 0.1. The result will send to software team to implement search for opponent flag properly.

## 6.6 Test 16: Search for Opponent's Flag

Hardware Version: 0.0.3

Software Version: 0.0.1

Date: November 15 to 22

**Objective:** The objective of this test is to test whether the robot can successfully search the opponent's flag within the searching area by using light sensor.

### Methodology:

In this experiment, we set our team as green team and the size of the board is 8 x 8 tiles. The robot will be placed at lower left-hand corner of search region in red player zone (6,2). The robot will receive the information including the coordinate of lower left-hand corner of search region in red play zone (5,2), upper right-hand corner of search region in red player zone (7,4) and the color of green opponent flag. Therefore, the searching region is 2 x 2 tiles. Two different color blocks will be randomly placed within the search region. Then, the robot will be asked to search opponent flag. Once the robot detects the opponent's block, it will beep three times. The testing team will record whether the robot can successfully search the opponent's flag. Ten trials will be conducted with different color of opponent flag.

### Result:

**Table 12: Block color detected**

<b>Trial No.</b>	<b>Color blocks in the search region</b>	<b>Color of opponent's flag</b>	<b>Status</b>
1	White, Yellow, Red, Blue	White	Detected
2	White, Yellow, Red, Blue	Yellow	Detected
3	White, Yellow, Red, Blue	Red	Detected
4	White, Yellow, Red, Blue	Blue	Detected
5	White, Yellow, Red	Yellow	Detected
6	White, Blue	Blue	Detected
7	Red, Blue	Blue	Undetected

8	White, Red, Blue	White	Detected
9	Yellow, Red, Blue	Red	Detected
10	White, Yellow, Blue	Yellow	Detected

### **Conclusion & Action:**

From the Table 12, the robot is able to detect the opponent's color block for majority of trials. There is one trial that the robot is not able to detect the opponent's color is because the opponent's color block is placed at the center of the searching region. Therefore, the distance between the block and the light sensor is very large. When the robot circle around the search region, it cannot detect the block, so it cannot distinguish the color as well. The result will send to software team to improve search opponent's flag.

## **6.7 Test 17: Cross the River**

Hardware Version: 0.0.3

Software Version: 0.0.3

Date: November 15 to 22

**Objective:** The objective of this test is to test whether the robot can successfully return back to starting corner by crossing the river.

### **Methodology:**

In this experiment, we set our team as green team and the size of the board is 8 x 8 tiles. Since our robot suppose return to lower left-hand corner of search region in red player zone after it finishes searching opponent's flag, so we will place the robot at lower left-hand corner of search region in red player zone (6,2). Then robot will receive the information from wifi including SH\_LL (6,6), SH\_UR (8,7) SV\_LL (7,3) and SV\_UR (8,7). Then the robot will be asked to cross the river and return back to starting corner 1. Five trials will be conducted. The testing team will record whether the robot can cross the river inside of shallow water zone and return back to starting corner.

**Result:****Table 13: Cross the river data**

<b>Trial No.</b>	<b>Status</b>
1	SUCCESSFUL
2	SUCCESSFUL
3	SUCCESSFUL
4	SUCCESSFUL
5	FAILED

**Conclusion & Action:**

From this experiment, the robot can successfully move within the shallow water zone. However, it cannot 100% return back to starting corner. There is one trial that the robot returns back to the tile which is next to the starting corner. The result will send to software team to improve cross the river code.

**7. INTEGRATION TEST****7.1 Test 18: Navigate to the zip line (Point 1)**

Hardware Version: 0.0.3

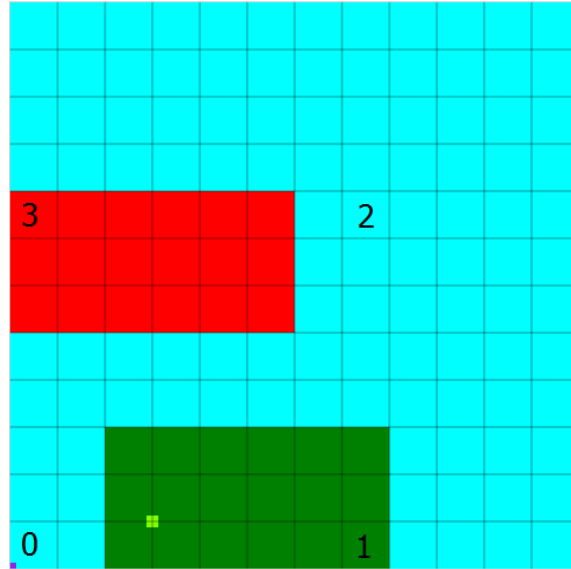
Software Version: 0.0.1

Date: November 3 to 9

**Objective:** This experiment was conducted to discover the behavior and accuracy of robot to navigating to the end point of zip line.

**Methodology:**

We first set our team as green team and the size of the board is 8 x 8 tiles. The robot will be place at different starting corner (corner 0 to 3). Then the robot will receive corresponding information from wifi including starting corner and coordinate of green zone zip line other (3,1), etc. Then the robot will start ultrasonic localization, light localization and navigate to the green zone zip line end point. Test team will record the behavior of robot during the test. Four trials will be conducted for different corner.



**Figure 13: Testing Map for Test 18**

**Result:**

**Table 14: Navigate to the zip line (Point 1)**

Starting corner	Status		
	US localization	Light localization	Navigate to the zipline other point
0	SUCCESSFUL	SUCCESSFUL	UNSUCCESSFUL
1	SUCCESSFUL	SUCCESSFUL	UNSUCCESSFUL
2	SUCCESSFUL	SUCCESSFUL	UNSUCCESSFUL
3	SUCCESSFUL	SUCCESSFUL	UNSUCCESSFUL

**Conclusion & Action:**

From the experiment, the robot can successfully do the ultrasonic localization and light localization. However, after light localization, the robot is not able to navigate the indicated coordinate for some reasons. Therefore, the final position error between end point of zip line and the robot actual position is large. This experiment indicate that the robot is not able to navigate destination position after light localization. The result will send to software team to improve the code.

## 7.2 Test 19: Navigate to the zip line (Point 2)

Hardware Version: 0.0.3

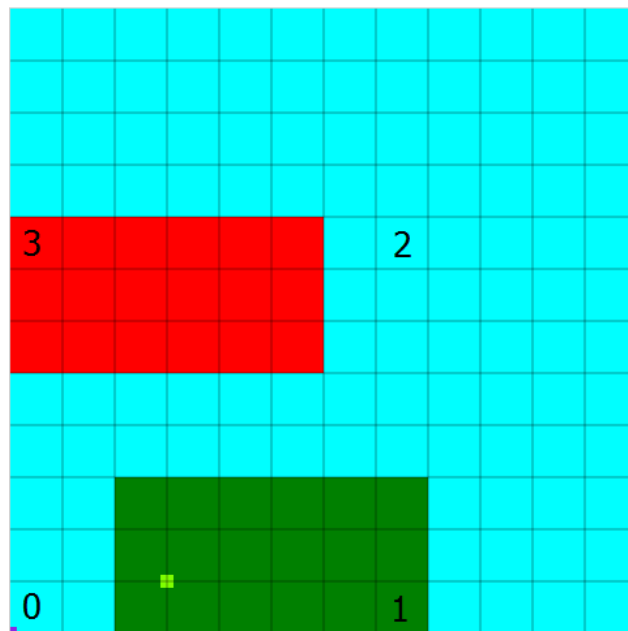
Software Version: 0.0.2

Date: November 10 to 14

**Objective:** This experiment was conducted to discover the behavior and accuracy of robot to navigating to the end point of zip line after the improvement of software.

### **Methodology:**

We first set our team as green team. The robot will be place at different starting corner (corner 0 to 3). Then the robot will receive corresponding information from wifi (include starting corner and coordinate of green zone zip line other (3,1) etc.). Then the robot will start ultrasonic localization, light localization and navigate to the green zone zip line other point. Test team will record the behavior of robot during the test. Four trials will be conducted for different corner.



**Figure 14: Testing Map for Test 19**

**Result:****Table 15: Navigate to the zip line (Point 2)**

Starting corner	Status		
	US localization	Light sensor localization	Navigate to the zip line other point
0	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL
1	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL
2	SUCCESSFUL	SUCCESSFUL	UNSUCCESSFUL
3	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL

**Conclusion & Action:**

From the experiment, the robot can successfully navigate to the green zone zip line endpoint except for corner 2. The ultrasonic localization and light localization perform well. However, the corner 2 is farthest corner to the green zone zip line other point. Therefore, the robot need to navigate very long distance to arrive the green zone zip line other point. During the navigation, the robot is not able to keep moving straightly, the angle error increases when the navigation distance increases. Therefore, after the robot finishes the navigation to the green zone zip line other point, the angle error is very large. In conclusion, the robot is not able to accurately navigate the zip line other point if the starting corner is very far. The result will send to software team and hardware team in order to improve the behavior of the robot.

**7.3 Test 20: Zip line Traversing (mount, cross and dismount)**

Hardware Version: 03

Software Version: 0.0.2

Date: November 10 to 14

**Objective:** This experiment was conducted to discover whether the robot can successfully mount, cross and dismount the zip line after navigating to green zone other zip line other point.

**Methodology:**

We first set our team as green team. The robot will be place at different starting corner (corner 0 to 3). Then the robot will receive corresponding information from wifi (include starting corner, coordinate of green zone zip line other (3,1), coordinate of green zone zip line endpoint (3,2) and coordinate of red zone zip line endpoint (3,6) etc.). Then the robot will do ultrasonic localization, light localization navigate to the green zone zip line other

point. Then, it will localize again and cross the zip and dismount. Test team will record the behavior of robot during the test. Four trials will be conducted for different starting corner.

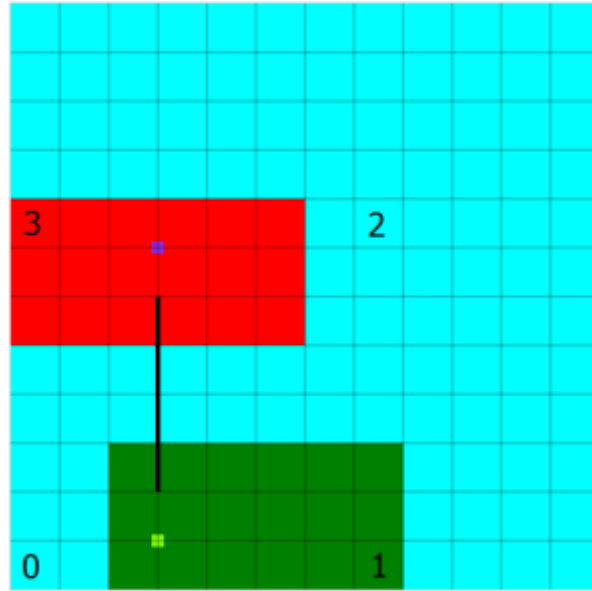


Figure 15: Testing Map for Test 20

**Result:**

Table 16: Traversing Zip line

Starting corner	Status			
	US localization	Light sensor localization	Navigate to the zip line other point	Traverse the zip line
0	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL
1	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL
2	SUCCESSFUL	SUCCESSFUL	UNSUCCESSFUL	UNSUCCESSFUL
3	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL

#### Conclusion & Action:

From the experiment, the robot can successfully mount, cross and dismount the zip line except for starting corner 2. Since if the starting corner is corner 2, the robot is not able to accurately navigate to green zone zip line other point. The reason is same as Test 15. The ultrasonic localization and light localization perform well. However, the corner 2 is farthest corner to the green zone zip line other point. Therefore, the angle error after the robot



navigates to the green zone zip line other point is very large. Even though the robot is able to localize again at the green zone zip line other point. It is not able to accurately mount to zip line if the starting corner is corner 2. However, when the starting corner is corner 0, corner 1 or corner 3. The robot can successfully mount, cross and dismount the zip line after navigating to green zone zip line other point. The result will send to software team and hardware team in order to improve the behavior of the robot.

#### 7.4 Test 21: Navigate to the opponent's flag location I (the zip line is parallel to the x-axis)

Hardware Version: 03

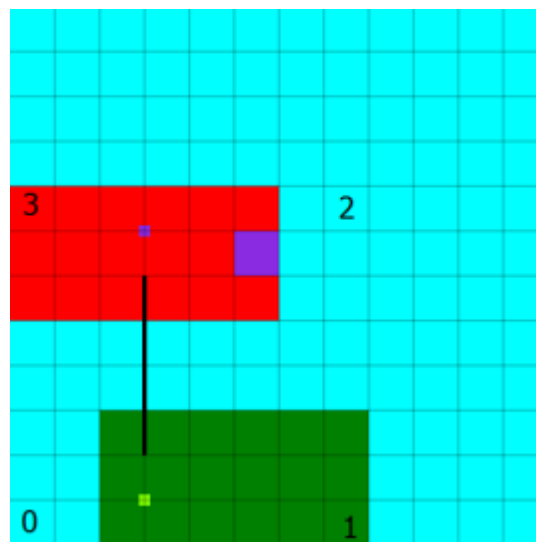
Software Version: 0.0.2

Date: November 10 to 14

**Objective:** This experiment was conducted to discover whether the robot can successfully navigate to the opponent's flag location.

##### **Methodology:**

We first set our team as green team. The robot will be place at different starting corner (corner 0 to 3). Then the robot will receive corresponding information from wifi (include starting corner, coordinate of green zone zip line other (3,1), coordinate of green zone zip line endpoint (3,2), coordinate of red zone zip line endpoint (3,6) and coordinate of upper right-hand corner of search region in red zone (5,6) etc.). Then the robot will do ultrasonic localization, light localization navigate to the green zone zip line other point. Then, it will localize again and cross the zip and dismount. After the robot dismount from the zip line, it will localize again and navigate to the upper right-hand corner of search region in red zone. Test team will record the behavior of robot during the test. Four trials will be conducted for different starting corner and for different opponent's flag location.



**Figure 16: Testing Map for Test 21**

**Result:****Table 17: Navigate to the opponent's flag location test data**

<b>Starting corner</b>	<b>Status</b>				
	<b>US localization</b>	<b>Light sensor localization</b>	<b>Navigate to the zipline other point</b>	<b>Traverse the zipline</b>	<b>Naviagte to the opponet's flag location</b>
<b>0</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>
<b>1</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>
<b>2</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>
<b>3</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>

**Conclusion & Action:**

From the experiment, the robot can successfully navigate to opponent's flag location after dismounting from the zip line except for starting corner 2. Since starting corner 2 is farthest corner to the green zone zip line other point. Therefore, the robot is not able to navigate to the green zone zip line other point, so the robot is not able to mount to zip line. However, for other 3 corners, the robot is able to mount, cross and dismount from the zip line and successfully navigate upper right-hand corner of search region in red zone. However, since our robot is not able to move straightly during navigation, if the opponent's flag location is very far away from the dismounting point, the result becomes inaccurate. The result will send to software team and hardware team in order to improve the behavior of the robot.

## 7.5 Test 22: Navigate to the opponent's flag location II (the zip line is diagonal)

Hardware Version: 0.0.4

Software Version: 0.0.3

Date: November 15 to 22

**Objective:** This experiment was conducted to discover whether the robot can successfully navigate to the opponent's flag location.

### Methodology:

We first set our team as green team and the size of the board is 8 x 8 tiles. The robot will be placed at different starting corner (corner 0 to 3). Then the robot will receive corresponding information from wifi including starting corner, coordinate of green zone zip line other (6,7), coordinate of green zone zip line endpoint (5,6), coordinate of red zone zip line endpoint (2,3) and coordinate of upper right-hand corner of search region in red zone (1,2), etc. For this map, the zip line is diagonal from the coordinate (5,6) to the coordinate (2,3). Then the robot will do ultrasonic localization, light localization navigate to the green zone zip line other point. Then, it will localize again and cross the zip and dismount. After the robot dismount from the zip line, it will localize again and navigate to the upper right-hand corner of search region in red zone. Test team will record the behavior of robot during the test. Four trials will be conducted for different starting corner.

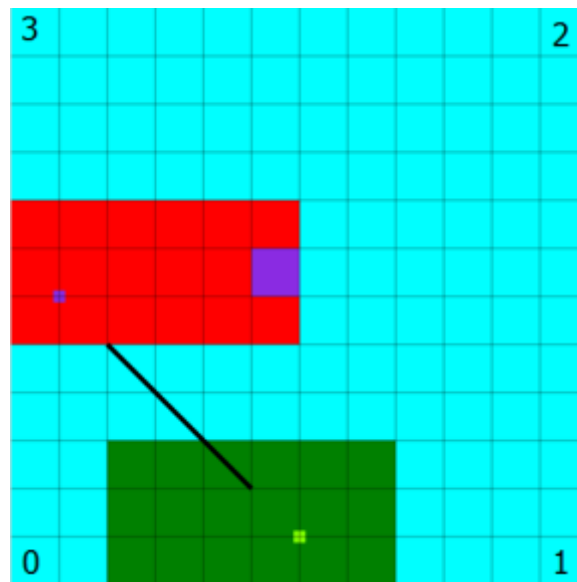


Figure 17: Testing Map for Test 22

**Result:****Table 18: Navigate to the opponent's flag location test data**

Starting corner	Status				
	US localization	Light sensor localization	Navigate to the zip line other point	Traverse the zip line	Navigate to the opponent's flag location
<b>0</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>
<b>1</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>
<b>2</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>
<b>3</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>	<b>SUCCESSFUL</b>

**Conclusion & Action:**

From the experiment, the robot can successfully navigate to opponent's flag location after dismounting from the zip line for all the starting coordinate. However, unlike the previous test, the Euclidean distance error after the robot navigate to the opponent's flag location is more than 3 cm. The reason is that the zip line is diagonal rather than parallel. Therefore, the robot is not able to accurately navigate to the opponent's flag location. Since this test is to test the ability to navigate the opponent's flag position rather than the accuracy. Therefore, in conclusion, the robot is able to navigate to the opponent's flag location, nevertheless, the robot final position is not accurate, the Euclidean distance error between the robot's actual position and the opponent's flag location is large. The result will send to software team and hardware team in order to improve the behavior of the robot.

## 7.6 Test 23: Search the opponent's flag

Hardware Version: 0.0.4

Software Version: 0.0.3

Date: November 15 to 22

**Objective:** This experiment was conducted to discover whether the robot can successfully search the opponent's flag after navigating to the opponent's flag location.

### Methodology:

We first set our team as green team and the size of the board is 8 x 8 tiles. The robot will be placed at different starting corner (corner 0 to 3). Then the robot will receive corresponding information from wifi including starting corner, coordinate of green zone zip line other (6,7), coordinate of green zone zip line endpoint (5,6), coordinate of red zone zip line endpoint (2,3), coordinate of upper right-hand corner of search region in red zone (1,2), and the color of green opponent flag, etc. For this map, the zip line is diagonal from the coordinate (5,6) to the coordinate (2,3). Different color blocks will be placed in the search region. Then the robot will be asked to localize, navigate and cross the zip line, and then navigate to opponent's search region and search the opponent flag. Test team will record whether the robot can successfully detect the opponent flag. Four trials will be conducted for different starting corner and for different color of opponent's flag.

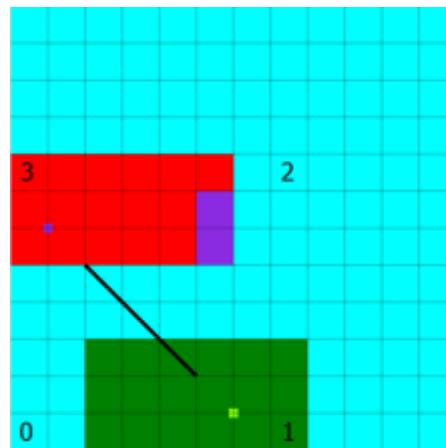


Figure 18: Testing Map for Test 23

**Result:**

**Table 19: Navigate to the opponent's flag location test data**

Starting corner	Status				
	Localization	Navigate to the zip line other point	Cross the zip line and navigate to the opponent's flag location	Color of the opponent's flag	Search the opponent flag
0	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	White	SUCCESSFUL
1	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	Blue	SUCCESSFUL
2	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	Yellow	SUCCESSFUL
3	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	Red	SUCCESSFUL

**Conclusion & Action:**

From the experiment, the robot can successfully detect the opponent's block after navigating to opponent's flag location for all different corners. The result will send to software team and hardware team in order to improve the behavior of the robot.

## 8 SYSTEM TESTS

### 8.1 Test 24: Final Test I (8 x 8 tiles)

Hardware Version: 0.0.4

Software Version: 0.0.3

Date: November 15 to 22

**Objective:** This experiment was conducted to discover whether the robot can successfully return back to its starting corner by either traversing the zip line or crossing the river.

**Methodology:**

We first set our team as green team and the size of the board is 8 x 8 tiles. The testing map is shown below. The robot will be place at starting corner 1. Two color blocks will be placed in the search region. The robot will receive all the information about coordinates through wifi. Then, the robot will be asked to localize, navigate and cross the zip line,

search the opponent's flag at opponent's flag region, and then return back to starting corner 1 by cross the river. Five trials will be conducted.

After, we set our team as red team and place the robot at starting corner 3. Two color blocks will be placed in the search region. The robot will receive all needed information Then, the robot will be asked to localize, cross the river, search the opponent's flag at opponent's flag region and return back to starting corner 3 by traversing the zip line. Five trials will be conducted.

Ten trials will be conducted. Testing team will record whether the robot complete all the task.

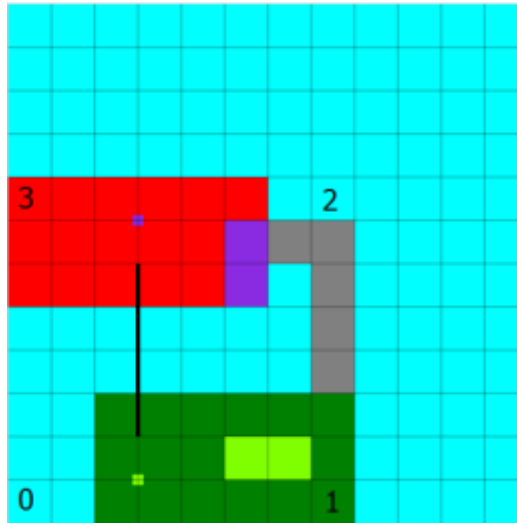


Figure 19: Testing Map for Test 24

**Result:**

**Table 20: Green team test data (8 x 8 tiles)**

Trial No.	Status				
	Localization	Navigate and cross the zip line	Search opponent's flag	Cross the river	Return back to starting corner
1	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL
2	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL
3	SUCCESSFUL	SUCCESSFUL	FAILED	SUCCESSFUL	SUCCESSFUL
4	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL
5	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL

**Table 21: Red team test data (8 x 8 tiles)**

Trial No.	Status				
	Localization	Cross the river	Search opponent's flag	Navigate and cross the zip line	Return back to starting corner
1	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL
2	SUCCESSFUL	SUCCESSFUL	FAILED	SUCCESSFUL	SUCCESSFUL
3	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL
4	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL
5	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL

**Conclusion & Action:**

From the experiment, the robot can successfully complete most of tasks in 8x8 tiles board for both green team and red team. However, the robot has probability to fail detect the opponent's flag. Since, the robot would return back to opponent's flag location after searching, it can complete other tasks without successfully detect opponent's flag.

**8.2 Test 25: Final Test II (12x 12 tiles)**

Hardware Version: 0.0.4

Software Version: 0.0.3

Date: November 23

**Objective:** This experiment was conducted to discover whether the robot can successfully return back to its starting corner by either traversing the zip line or crossing the river.

**Methodology:**

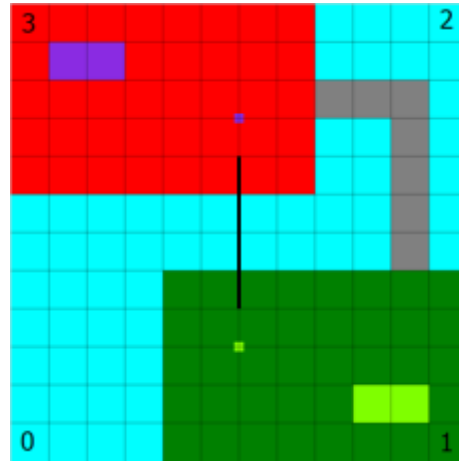
We first set our team as green team and the size of the board is 12 x 12 tiles. The testing map is shown below. The robot will be place at starting corner 1. Two color blocks will be placed in the search region. The robot will receive all the information about coordinates through wifi. Then, the robot will be asked to localize, navigate and cross the zip line, search the opponent's flag at opponent's flag region, and then return back to starting corner 1 by cross the river. Three trials will be conducted.

After, we set our team as red team and place the robot at starting corner 3. Two color blocks will be placed in the search region. The robot will receive all needed information Then,



the robot will be asked to localize, cross the river, search the opponent's flag at opponent's flag region and return back to starting corner 3 by traversing the zip line. Three trials will be conducted.

Total of six trials will be conducted. Testing team will record whether the robot complete all the task within 5 minutes.



**Figure 20: Testing Map for Test 25**

**Result:**

**Table 22: Green team test data (12 x 12 tiles)**

Trial No.	Status				
	Localization	Navigate and cross the zip line	Search opponent's flag	Cross the river	Return back to starting corner
1	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL
2	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	FAILED
3	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	FAILED

**Table 23: Red team test data (12 x 12 tiles)**

<b>Trial No.</b>	<b>Status</b>				
	<b>Localization</b>	<b>Cross the river</b>	<b>Search opponent's flag</b>	<b>Navigate and cross the zip line</b>	<b>Return back to starting corner</b>
<b>1</b>	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	FAILED	FAILED
<b>2</b>	SUCCESSFUL	SUCCESSFUL	FAILED	SUCCESSFUL	FAILED
<b>3</b>	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL	SUCCESSFUL

**Conclusion & Action:**

From the experiment, the robot can complete most tasks in 12 x 12 tiles board for both green team and red team. However, it has high probability to fail return back to starting corner. When it failed to return back to starting corner, it actually returned to the tile next to the starting corner. The reason is when the robot navigates for a long distance, it cannot keep moving straight. Moreover, there is one trial that robot is not able to detect opponent's flag in the opponent's flag region. Furthermore, there is one trial that robot failed to cross the zip line. The reason is that the robot in red team cross the river first and then return back to starting corner by traversing the zip line, therefore, the robot need to navigate for a long distance in order to traverse the zip line. For the same reason, the robot cannot keep moving straight when it navigate for a long distance, there it failed to localize at the zip line other point before traversing the zip line.

**9 REFERENCE**

- All the test maps shown in Figure 13, 14, 15, 16, 17, 18, 19 and 20 are from <https://d23uykxzxkfjvj.cloudfront.net/>