SYSC 4805 Computer Systems Lab

Snow Plow Project Final Report

Lab Section: L2 - Group 4

Date: April 11th 2022

Group Members:

Name	Student Number
Huzaifa Mazhar	101071739
Leenesh Kumar	101115874
Will Brooks	101075150
Alex Tasseron	101067304

PROJECT CHARTER:

OBJECTIVE:

The objective of the project is to create a mobile snow plow robot. The robot will remove as much snow as possible from a simulated area.

DELIVERABLES:

The purpose of the project is to design and implement a robot in Coppeliasim to clear snow from an enclosed area. The design of the robot must adhere to the physical specifications (size, starting position, speed) outlined in the project description. Any sensors mounted to the robot must be realistic and have a corresponding link to the real-world vendor of a similar sensor for documentation purposes. External programming libraries are allowed to help automate the robot, however augmentation of the main script is prohibited. Additionally, the robot must operate in an enclosed area defined by a black line during simulation. Ideally, this task should be accomplished such that the maximum amount of snow is cleared and the minimum number of collisions with obstacles occur within a given timeframe. Success is measured by the number of ‘snow balls’ cleared by the robot. A deduction of 20 snowballs is assessed anytime the robot collides with an obstacle. The robot will be tested on 3 testing maps and must be able to run for the entire 5-minute duration of the simulation. Design, testing, and grading of the robot will be completed by April 11th.

Scope:

List of Requirements:

Listed below are the requirements that the robot must adhere, as outlined in the project documentation.

1) The robot must fit within a volume of 0.5 m x 0.8 m x 1 m when parked
2) The robot's volume must not exceed 1 m x 0.8 m x 1 m at any time during the simulation.
3) The robot must navigate the testing area without getting stuck for the duration of the simulation
4) The robot must not leave the testing area during operation for the duration of the simulation
5) The robot must not collide with obstacles in the testing area for the duration of the simulation
6) The robot must remove the snow from the testing area using a plow. Snow cannot be blown out of the area.
7) The robot must not exceed 2 m/s during the simulation.

Work Breakdown Structure:

The following WBS structure is a decomposition of all the AOIs(Areas of Interest) for our project. From each branch is a description of every activity that pertains to that AOI. This WBS is an integral part of our project as it covers the entire scope and the desired deliverables at the end.

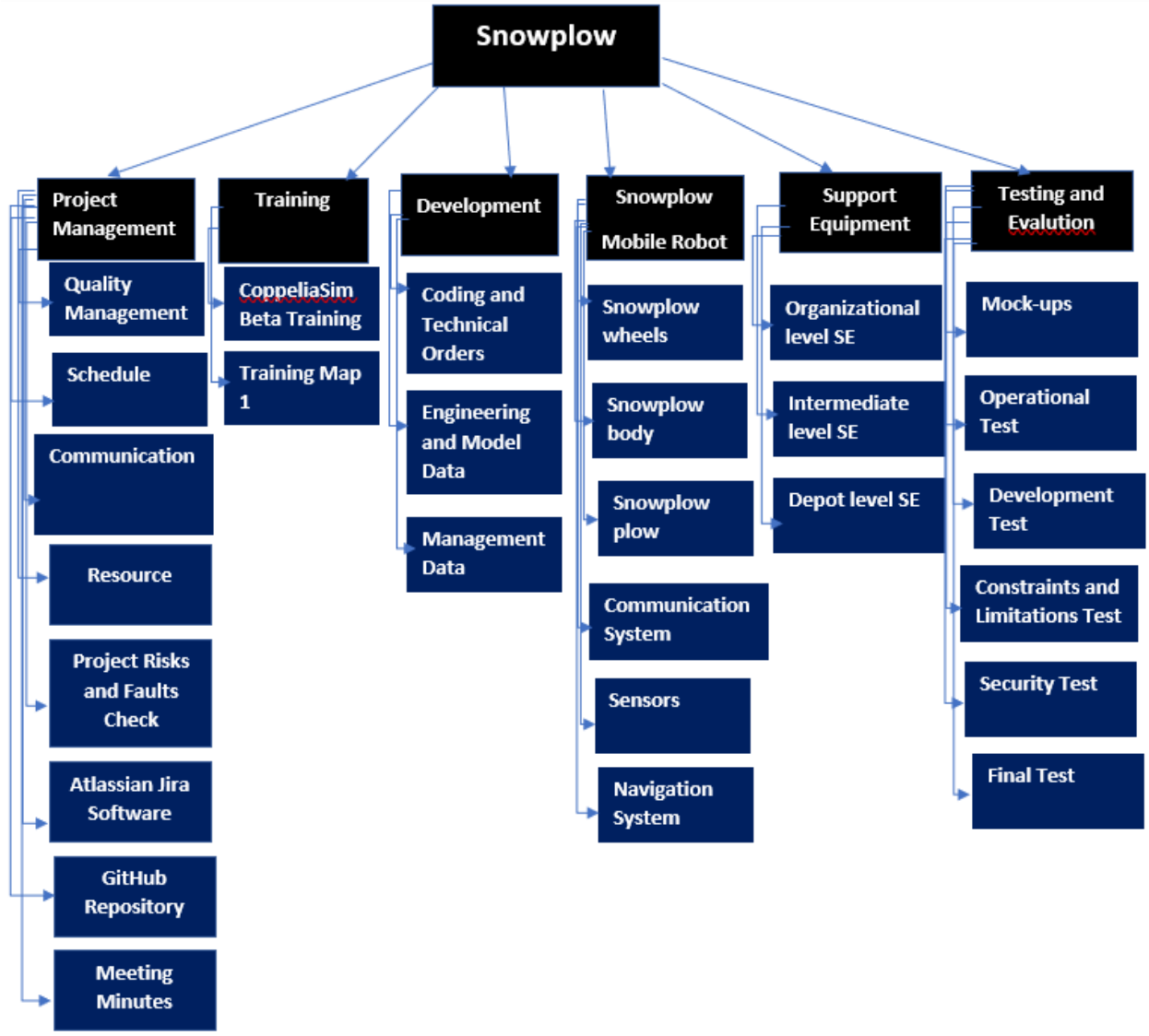


Figure 1: Work-Breakdown-Structure diagram for the SYSC4805 Snow plow lab project

Testing:

The proposed acceptance tests for the project may be seen in the table below. Successfully completion of the project depends on all acceptance tests being passed.

Test Number	Tested Require-ment(s)	Test Procedure	Pass/Fail Conditions
1	1	Measure the x, y, and z dimensions of the robot in parked configuration.	If the dimensions are less than the maximum then the test passes. If not the test fails
2	2	Measure the x, y, and z dimensions of the robot in the configuration that maximizes each dimension.	If the dimensions are less than the maximum then the test passes. If not the test fails
3	3	Begin simulation. Allow the robot to navigate the testing area for 5 minutes.	If the robot gets stuck within the time the test fails. If the robot does not get stuck within the time, the test passes.
4	4	Begin simulation. Allow the robot to navigate the testing area for 5 minutes.	If the robot leaves the testing area during the simulation time the test fails. If the robot remains within the testing area during the time the test passes.
5	5	Begin simulation. Allow the robot to navigate the testing area for 5 minutes.	If the robot collides with an obstacle during the simulation time the test fails. If the robot does not collide with an obstacle during the simulation time, the test passes
6	6	Begin simulation. Allow the robot to navigate the testing area for 5 minutes.	If snow is removed from the testing area using only a pushing action from the robot the test passes. If snow is removed from the area using any action other than pushing the test fails.
7	7	Begin simulation. Allow the robot to navigate the	If the speed exceeds 2 m/s then the test fails.

		testing area for 5 minutes. Measure the speed of the robot during the simulation.	If the speed does not exceed 2 m/s then the test succeeds
8	7	Calculate the maximum speed of the robot based on wheel size and angular velocity.	If the calculated speed exceeds 2 m/s the test fails. If the calculated speed does not exceed 2 m/s the test passes.

Table 1: Tests that need to pass to meet functional requirements of project

Scheduling:

The schedule has been created to allocate each group member one activity to complete for each lab week. There are a total of 28 activities amongst four members. Each activity is designed and compartmentalized to take approximately one week so that it neatly coincides with the weekly lab schedule.

List of activities:

Format
[lab #]:
    [activity #] - [assignee] - title and description - est time required

Lab4 (starting week of February 4th):
    Activity 1 - Huzaifa - Iteration 1 of snow plow design option 2 - 1 week
    Activity 2 - Will - Iteration 1 of snow plow design option 1 - 1 week
    Activity 3 - Leenesh - Work on training map 1 - 1 week
    Activity 4 - Alex - Investigate obstacle detecting sensors/camera and collision avoiding methods - 1 week

Lab5:
    Activity 5 - Huzaifa - Iteration 2 on snow plow design option 2 + testing effectiveness - 1 week
    Activity 6 - Will - Iteration 2 snow plow design option 1 - 1 week
    Activity 7 - Leenesh - Test and implement line-following till faultless - 1 week
    Activity 8 - Alex - Iteration 1 of implementing obstacle detection and collision avoidance in CoppeliaSim - 1 week

Lab 6:

- Activity 9 - Huzaifa - Iteration 3 on snow plow option 2 + testing effectiveness at snow removal - 1 week
- Activity 10 - Will - Iteration 3 on snow plow option 1 and test effectiveness at snow removal - 1 week
- Activity 11 - Leenesh - Collect data on the two snow plows and analyze the effectiveness of the two plow options - 1 week
- Activity 12 - Alex - Iteration 2 of implementing obstacle detection and collision avoidance - 1 week

Lab 7:

Group Task: Select best snow plow option.

- Activity 13 - Huzaifa - Integrate best snow plow to the body. Test all components to ensure functionality - 1 week
- Activity 14 - Will - Integrate best snow plow to the body. Test all components to ensure functionality - 1 week
- Activity 15 - Leenesh - Run simulation with integrated robot and identify faults/incomplete requirements - 1 week
- Activity 16 - Alex - Integrate the sensors to the final robot - 1 week

Lab 8:

- Activity 17 - Huzaifa - Snow plow scenario simulation testing - 1 week
- Activity 18 - Will - Snow plow simulation testing - 1 week
- Activity 19 - Leenesh - Analyze cause of any faults in the simulation testing - 1 week
- Activity 20 - Alex - Revise and fine-tune collision avoidance code - 1 week

Lab 9:

- Activity 21 - Huzaifa - Run simulation test 1 through 4 - 1 week
- Activity 22 - Will - Implement any needed changes based on Leenesh's findings from lab 8 - 1 week
- Activity 23 - Leenesh - Identify edge cases in simulation testing and anything else that doesn’t meet the functional requirements for the robot - 1 week
- Activity 24 - Alex - Run simulation tests 4 through 8 - 1 week

Lab 10:

- Activity 25 - Huzaifa - Final simulation testing - 1 week
- Activity 26 - Will - Final simulation testing - 1 week
- Activity 27 - Leenesh - Identify and record any functional requirements not met - 1 week
- Activity 28 - Alex - Demonstration practice runs - 1 week

Schedule Network Diagram:

The schedule network diagram lists an activity for each member for each lab week, and displays any dependencies an activity may have. The activity descriptions can be found listed above.
The date format is dd/mm/yyyy.

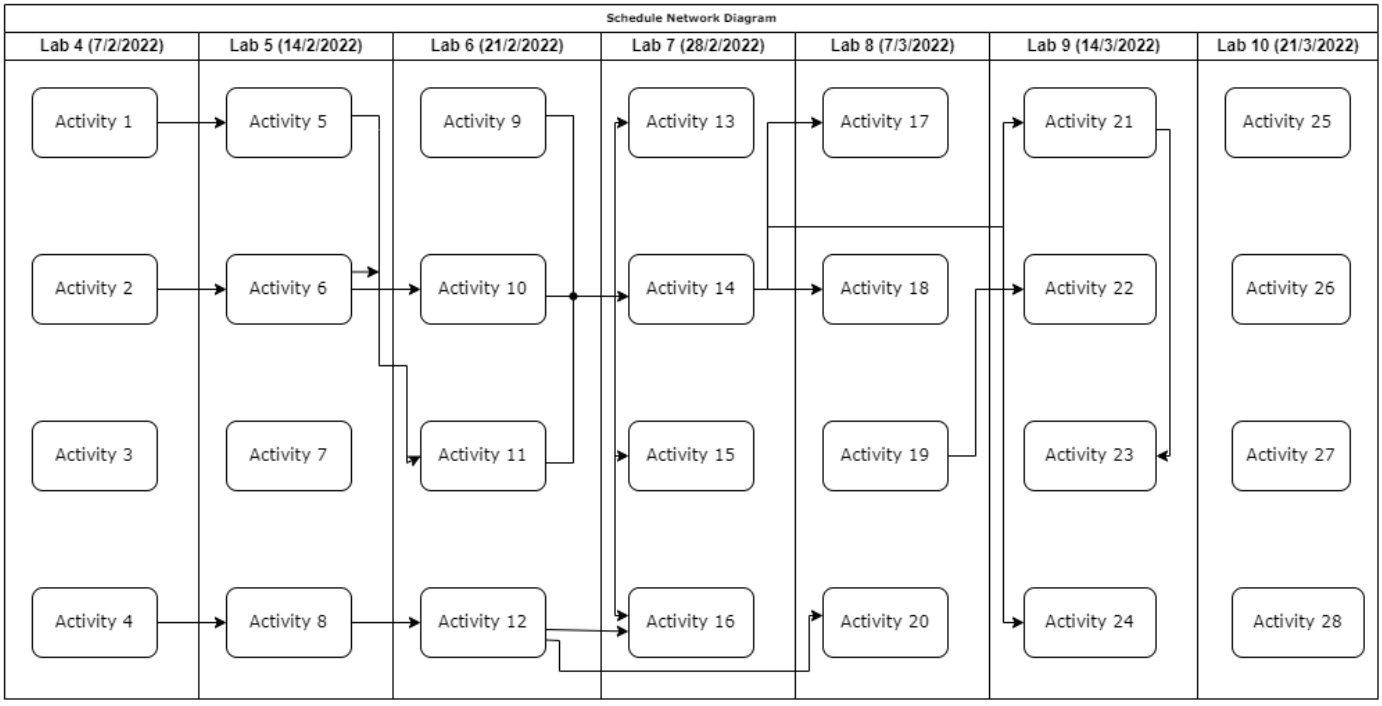


Figure 2: Schedule Network Diagram and dependencies for planned weekly activities

GANTT CHART:

The Gantt chart below displays the duration for each activity over the course of the project.

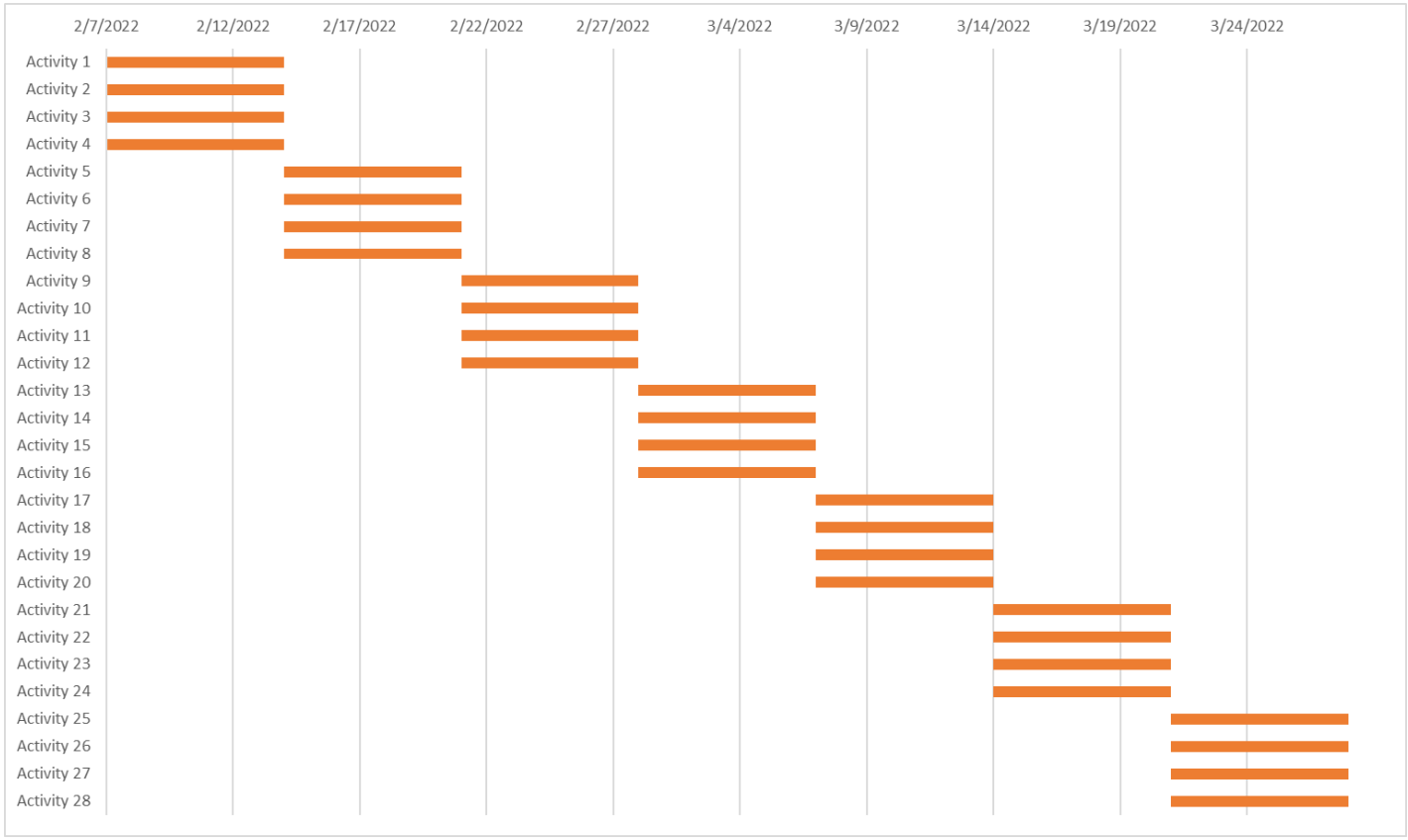


Figure 3: Gantt chart displaying duration of each activity

Human Resources:

Responsibility Assignment Matrix:

The activities required to complete the project have been organized into the Responsibility Assignment Matrix below. The Activity Number corresponds to the ones found in the Scheduling section of this document. Each group member has seven activities that they are assigned to complete, and seven activities for which they are responsible. Due to the nature of working with such a small team, all members will be informed of, and consulted on each activity, so these notations have been omitted from the matrix. The label “R” denotes a member is Responsible for an activity. The label “A” denotes a member is Accountable for an activity.

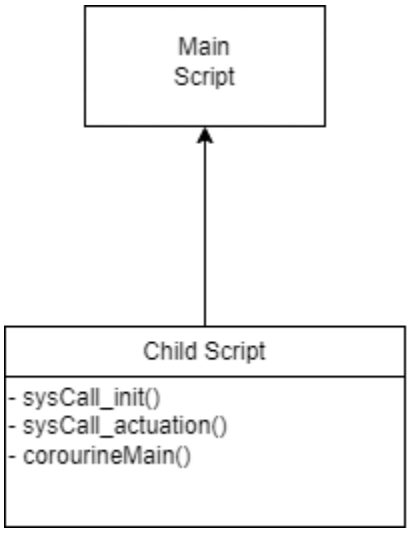
Activity Number	Alex	Huzaifa	Leenesh	Will
1	R	A		
2			R	A
3		R	A	
4	A			R
5		A	R	
6	R			A
7		R	A	
8	A			R
9	R	A		
10			R	A
11		R	A	
12	A			R
13		A		R
14		R		A

15	R		A	
16	A		R	
17		A		R
18		R		A
19	R		A	
20	A		R	
21	R	A		
22			R	A
23			A	R
24	A	R		
25		A	R	
26	R			A
27			A	R
28	A	R		

Table 2: Responsibility Assignment Matrix

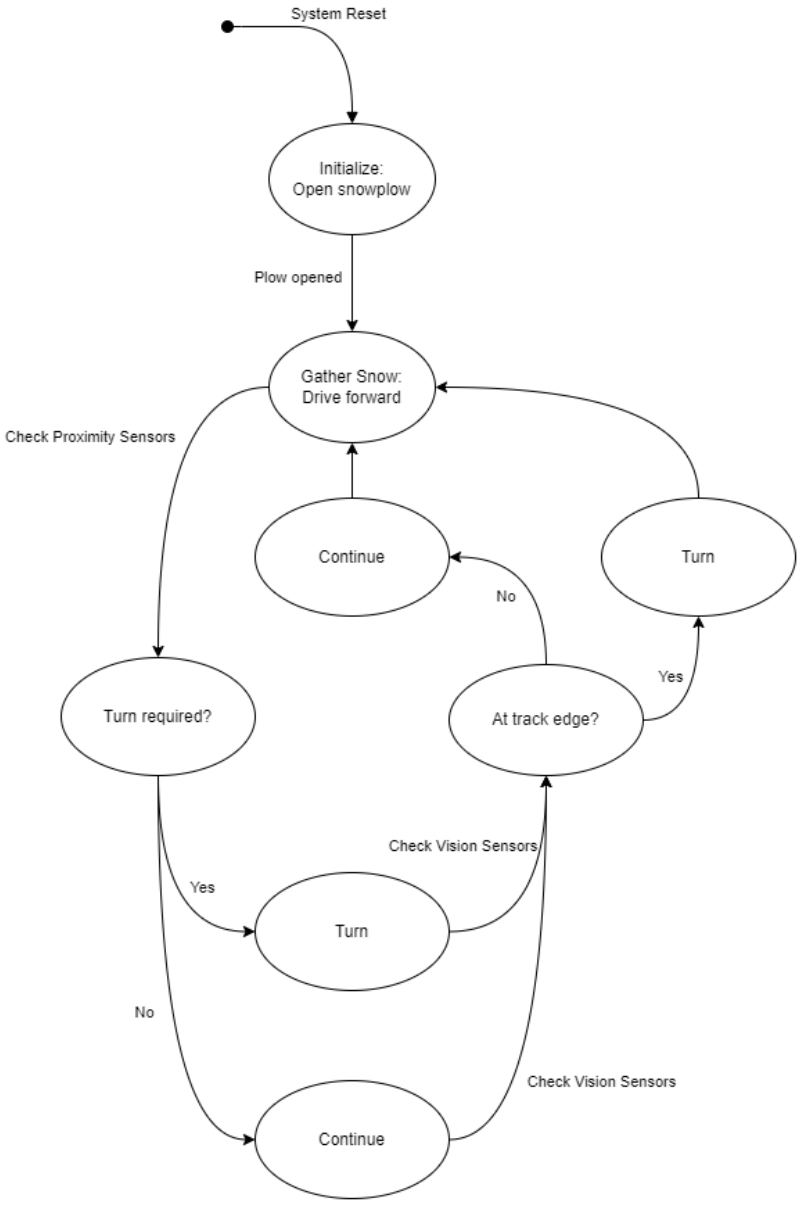
Overall Architecture

The robot is controlled by a child script that is attached to the model. The script is written in LUA and uses the standard CoppeliaSim API to control the robot. The control script is synchronous, meaning the robot's actions are decided by polling the proximity and vision sensors. This is done with a single loop. The movement of the robot is determined by the state of the sensors.

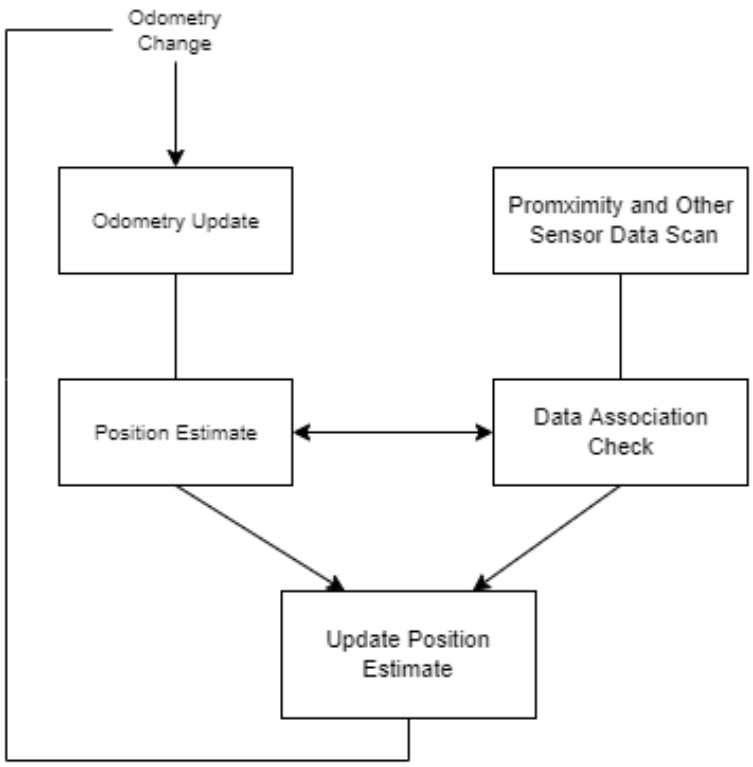


State Chart

The statechart representing the project can be seen below. The system is polling driven. This massively simplifies the writing of the script. This simplicity comes at the cost of slower response to stimuli and a higher overall power consumption. The additional power cost does result in more consistent power requirements since the system is always acting similarly.

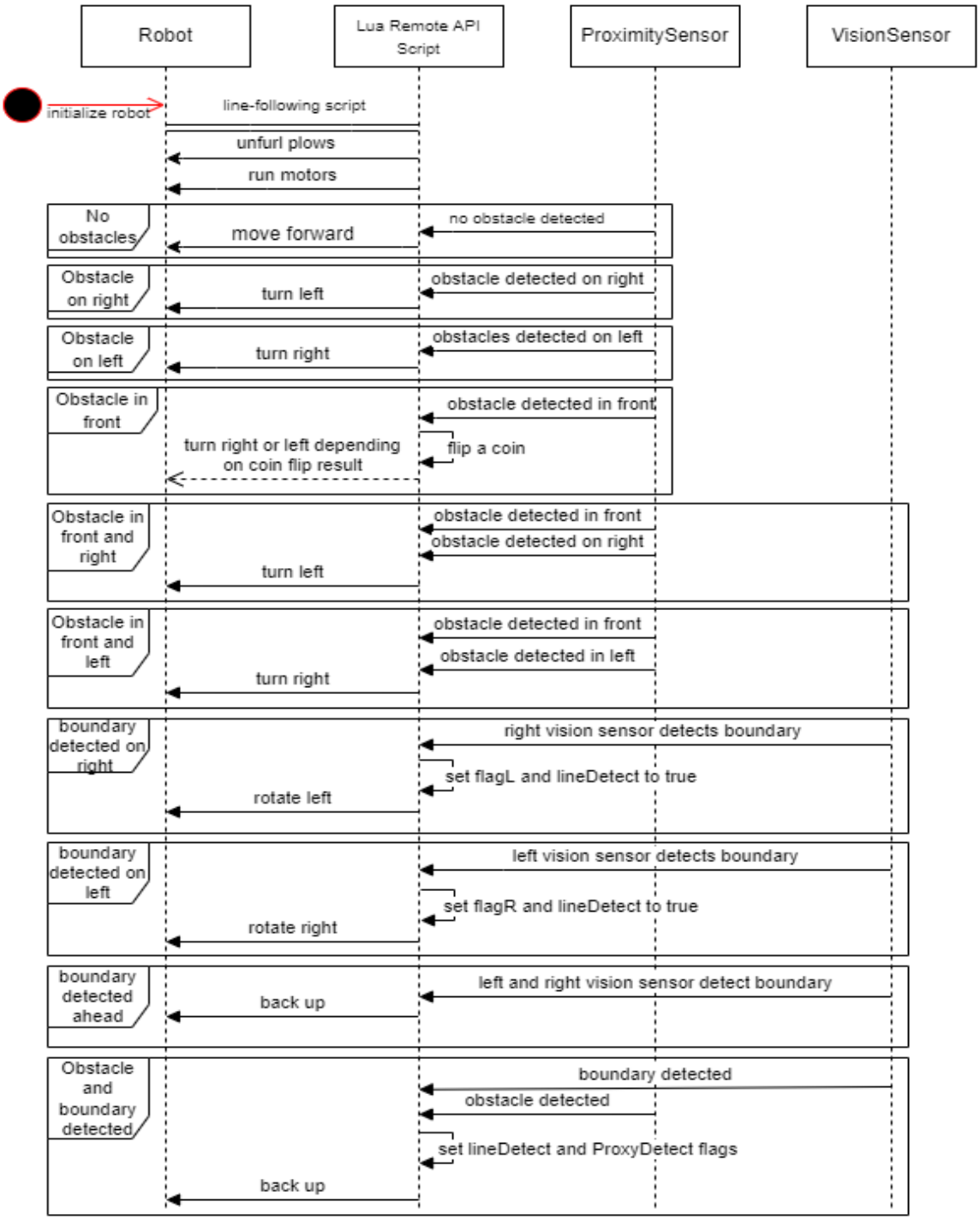


A potential solution being investigated for collision avoidance is Simultaneous Localization And Mapping (SLAM). Below is a generalized flowchart for using SLAM. The algorithm works by constantly updating a model of the world using sensor data, and using landmarks to localize itself in the map. This can be designed to be event driven and trigger certain actions. If the sensor data detects certain input, which will cause an interrupt.



Sequence Diagram

The following sequence diagram presents the possible cases for which the robot is designed to circumnavigate.



Project Budget

The project is expected to have a Budget at Completion of about $3600 total. $2800 of that is engineering costs and about $820 is materials costs. Over the course of the seven week project, four hours are budgeted for each of the four team members. A cost of $25 per engineering hour was used to get this number.

Materials cost breakdown can be seen in the table below.

Components	Number Required	Unit Cost ($)	Total Cost ($)	Supplier Link
Raspberry Pi 0 Kit (includes camera and two servos)	1	117	117	https://www.digikey.ca/en/products/detail/sparkfun-electronics/KIT-16327/12715013
IMU	1	14	14	https://www.sparkfun.com/products/10937
Proximity Sensor	1	4.5	4.5	https://www.sparkfun.com/products/15569
3D printer filament	1	35	35	https://www.amazon.ca/DURAMIC-3D-Filament-1-75mm-Black-1/dp/B095HRG913/ref=sr_1_2_sspa
Motors	2	325	650	https://www.robotis.us/dynamixel-xm430-w210-t/

Since the robot will not actually be manufactured the materials costs will not be spent. Currently two weeks of project work have been completed incurring an engineering time cost of $800. This is the current Planned Value of the project.

A spreadsheet containing the full cost calculations may be seen on the project GitHub.

Individual Contributions

Name	Contribution
Alex	- System architecture - LUA Script setup - Worked with team on coding tasks - Worked with team to complete proposal, progress report, and final report - Testing map simulation runs and optimizations
Will	- Modeled plow prototype 1 - 3D model setup in CoppeliaSim - Implemented obstacle avoidance in Lua script - Worked with team to complete proposal, progress report, and final report - Testing map simulation runs and optimizations
Huzaifa	- Modeled plow prototype 2 - 3D model setup in CoppeliaSim - Implemented map edge avoidance in Lua script - Worked with team to complete proposal, progress report, and final report - Testing map simulation runs and optimizations
Leenesh	- Handled project management and tasking - Worked to develop tests for the robot - Worked with team on coding tasks - Worked with team to complete proposal, progress report, and final report - Testing map simulation runs

GitHub Repository

All files relating to the project including this report have been uploaded to the following GitHub repository.

https://github.com/huzaifa73/SYSC4805/tree/main

Control Charts

The following control charts represent the data from a series of five trials conducted for every test map in order to determine overall performance of our snowplow. These graphs represent how well the collision avoidance and obstacle detection is on our robot.

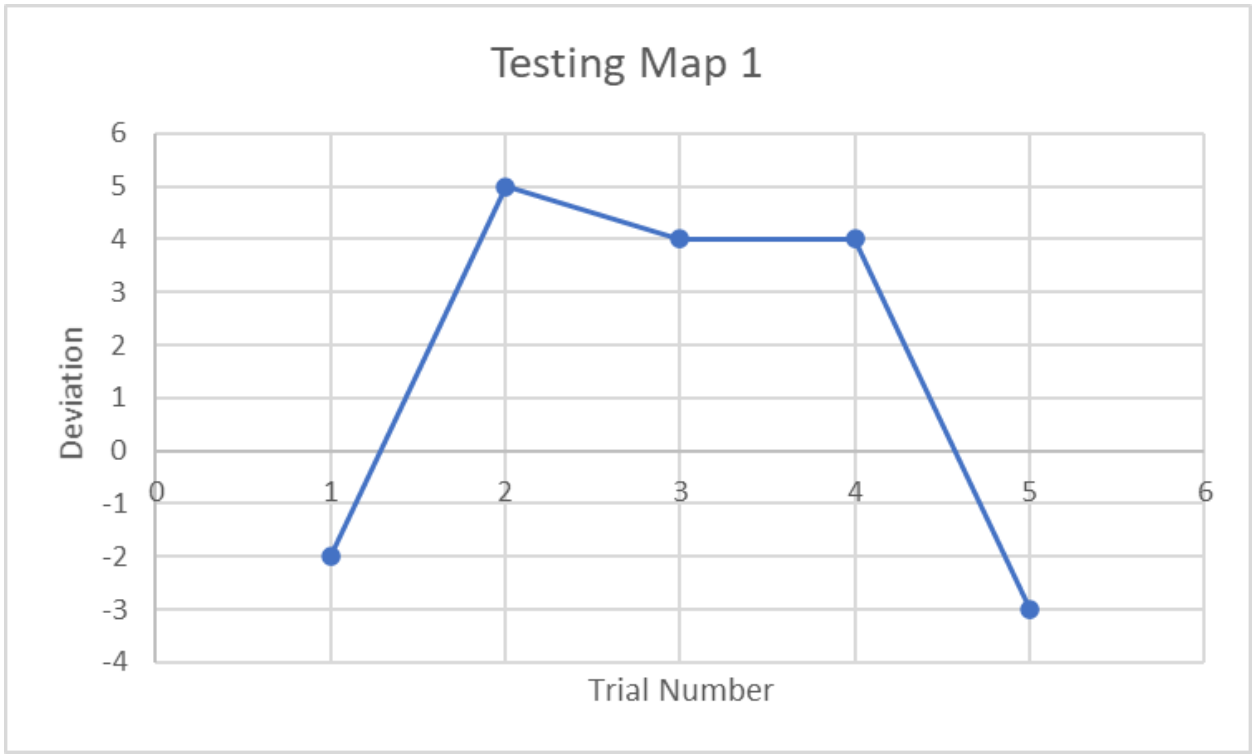


Fig 1) Results of our snowplow in Testing Map 1

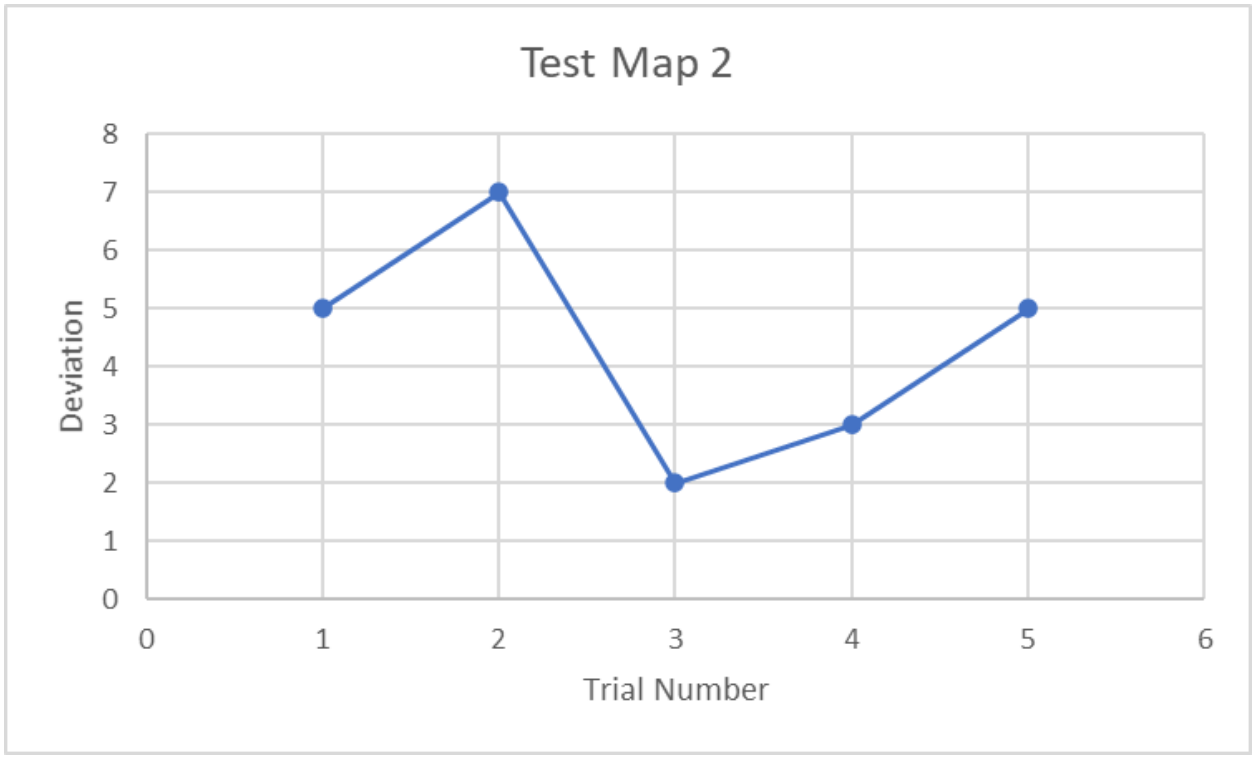


Fig 2) Results from our SnowPlow in Test Map 2

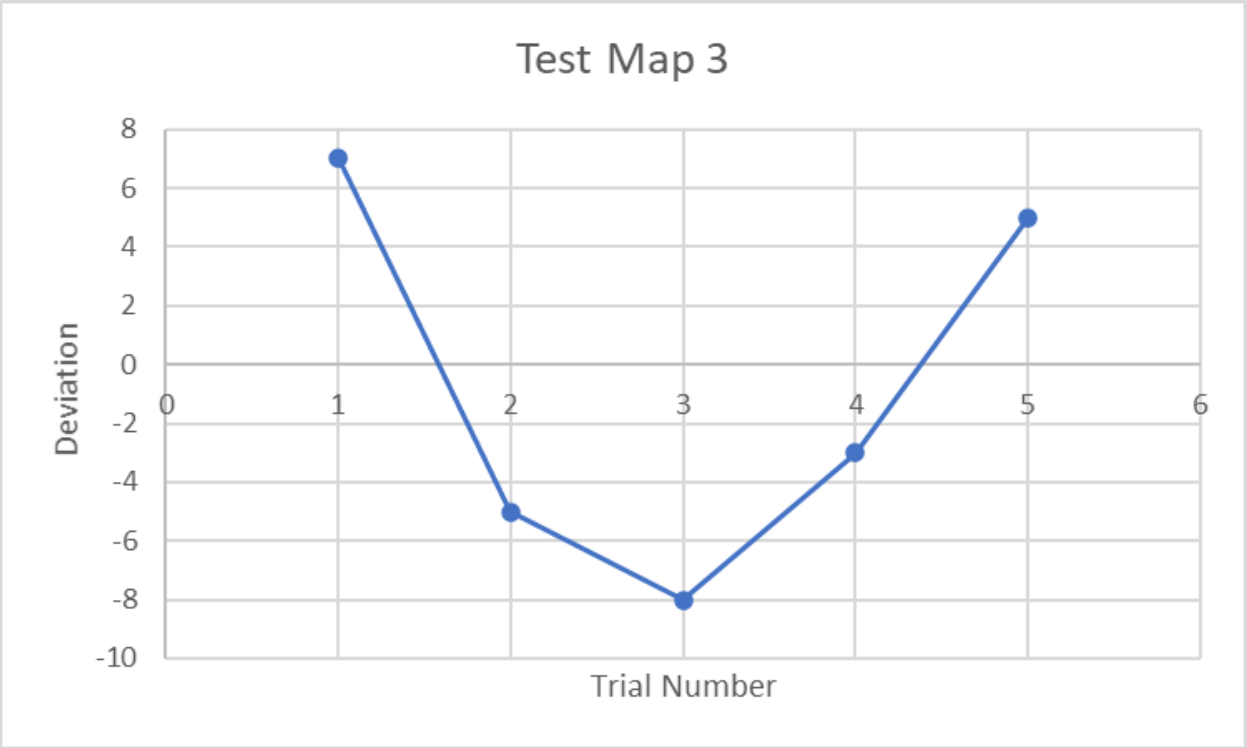


Fig 3) Results from our SnowPlow in Test Map 3

From these graphs, we can see that overall our snowplow performed well in the first 2 test maps, omitting a few anomalies. However, from the chart for test map 3, we see that overall performance of the snowplow dropped. This could be because each time the snowplow had trouble choosing a direction to go and would often be delayed, resulting in other stimuli walking into the robot from behind.

Testing Results

Training map 1

Bot leaves the enclosed area after 3:57. Approximately 61 Snow balls were removed before the bot left the area (snow had to be counted manually since there was not an automatic counted implemented on this training map). No collisions occurred before the bot left the area.

Training map 2

Bot leaves the enclosed area after 1:36. 433 Snow balls are remaining in enclosure at the time of failure. No collisions occurred. The total snow removed was 27 snowballs.

Training map 3

Bot collides with moving human at 4:03 resulting in plow damage rendering bot immobile. 389 Snow balls are remaining in enclosure at the time of failure. One collision occurred resulting in bot immobilized. The total snow removed was 103 snowballs. After penalties, 83 snowballs were removed.

Training map 4

The bot remained within the testing area for the full 5 minutes. After the time had expired there were 367 snowballs left in the enclosure. No collisions occurred. The total snowballs removed was 121.

Conclusion

The snowplow robot did not perform well in testing. The bot failed to meet requirement #3 by getting stuck after sustaining damage after a collision. It also failed to meet requirement #4 by leaving the map area.

Critical failures occurred on training maps one, two and three. On map one and two the bot failed to remain within the testing area. On the third training map the bot collides with a moving obstacle and sustains enough damage that it was not able to continue operation.

The failures resulting in leaving the testing area are likely a result of the method of controlling the robot's motors. During operation, the script first polls the proximity sensors followed by the vision sensors. This is not inherently an issue, however, after the proximity sensors are scanned the script will command the bot to move based on those readings. Only after the wheels are commanded does the script then check the vision sensors to detect the map edge. As a result, the bot can erroneously move forwards when it should be either turning or reversing while trying to remain inside the map. One way this could have been remedied would have been to poll both the proximity and vision sensors before deciding what command to give the wheels. This would prevent erroneous directions from one set of sensors.

The failure resulting from the collision with the moving obstacle is likely a result of the bot not reacting to the obstacle fast enough. The collision occurred in a head on scenario. This resulted in the minimum possible avoidance time for the bot. The only reason that this was a critical failure, however, was that the plow was knocked far enough out of alignment that the bot was no longer able to move. If the plow were designed in a more rugged manner then this would have only been a penalty instead of a total failure.

To see the full code listing of the script open the Commented_Script.txt file in the Final Report folder on the project GitHub.