Snow Plow Final Report

Group 9: Team Deep Space Sparkle

Ahmad Mozan	101080351
Bailey Lyster	101115419
Cameron Maccoll	101120539
Christian Fisher	101071157

Table of Contents

Project Charter : Team Deep Space Sparkle

Objective

Group 9's objective was to design and create a snow plowing robot in Coppelia Sim that travels around a track while plowing snow, avoiding any obstacles it may encounter.

Overall Deliverables

Group 9 delivered a Coppelia model file of a robot that can detect the road, detect snowfall and detect obstacles in its way. Included on the robot were sensors, wheels, a plow and a body. The robot runs scripts that interpret the sensor data. The scripts utilizes real-time data to plan routes to optimize distance travelled, follow the road, maximize snow displaced, adapt and re-route when obstacles are detected, deploy a wider plow and drive the wheels. These scripts are included among the deliverables. Group 9 delivered a progress report at the end of the third week of work detailing our progress on our deliverables up to that point and any adjustments we have made if we encounter any issues. A demonstration of the robot performing the required task on a multitude of different maps was performed at the end of the development period. This final write up details our final product and our solution to the problem.

Scope

Requirements

Software:

1. The main language of development is Python in conjunction with Lua to set up the API.

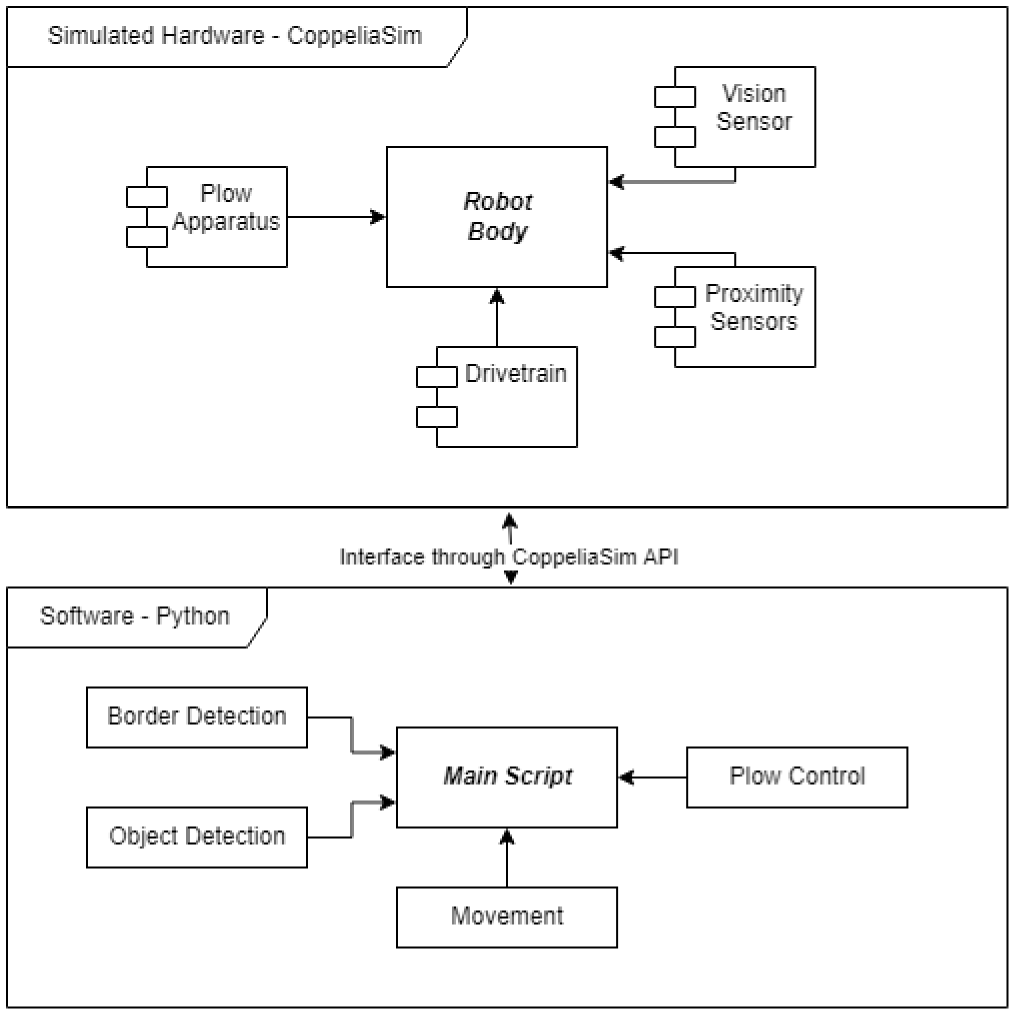
Functional:

2. Clear all snow from an area demarcated by a black path over the perimeter of a 12m x 12m grid.
3. Avoid all obstacles which could include both fixed and moving objects.

Physical Restrictions:

1. The robot must not exceed a space of 0.5m x 0.8 x 1.0m (W x L x H) at the starting rest position. The width may extend to 1.0m while the simulation is running.
2. The robot must not exceed a maximum speed of 2.0m/s at any point.
3. Any sensor used on the robot must have a real world equivalent.
4. The task must be completed within a 5 minute timeframe.

Overall Architecture



The architecture of this project consists of 3 discrete sections. These are the main code, the plow structure and the robot body.

The main code of the robot is driven by a while loop. This loop will first gather information about the robot's surroundings through the robot's sensors, then decide how to act upon this information, then finally take that action. The gathering of information about the robot's surroundings is done in a blocking mode, but these checks will not take a significant amount of time. Similarly the decision making phase does not take a significant amount of time. The actions however have the possibility of taking a non trivial amount of time. This is because actions such as rotating require a couple of seconds to complete when the action is attempted in a blocking mode.

The plow body consists of a back plate and two individual wings on the left and right side that can be manipulated independently of one another. Each wing is attached by a motor to do the main plate of the plow. Being able to individually control the shape and dimensions of the plow ensures the plow body can be the maximum possible size to increase the efficiency of snow clearing. Both wings of the plow extend to an angle of 130 degrees once the simulation starts. If the robot encounters an object, and decides to incrementally turn to avoid it, the plow can be retracted on one side to attempt to conserve the maximum amount of snow.

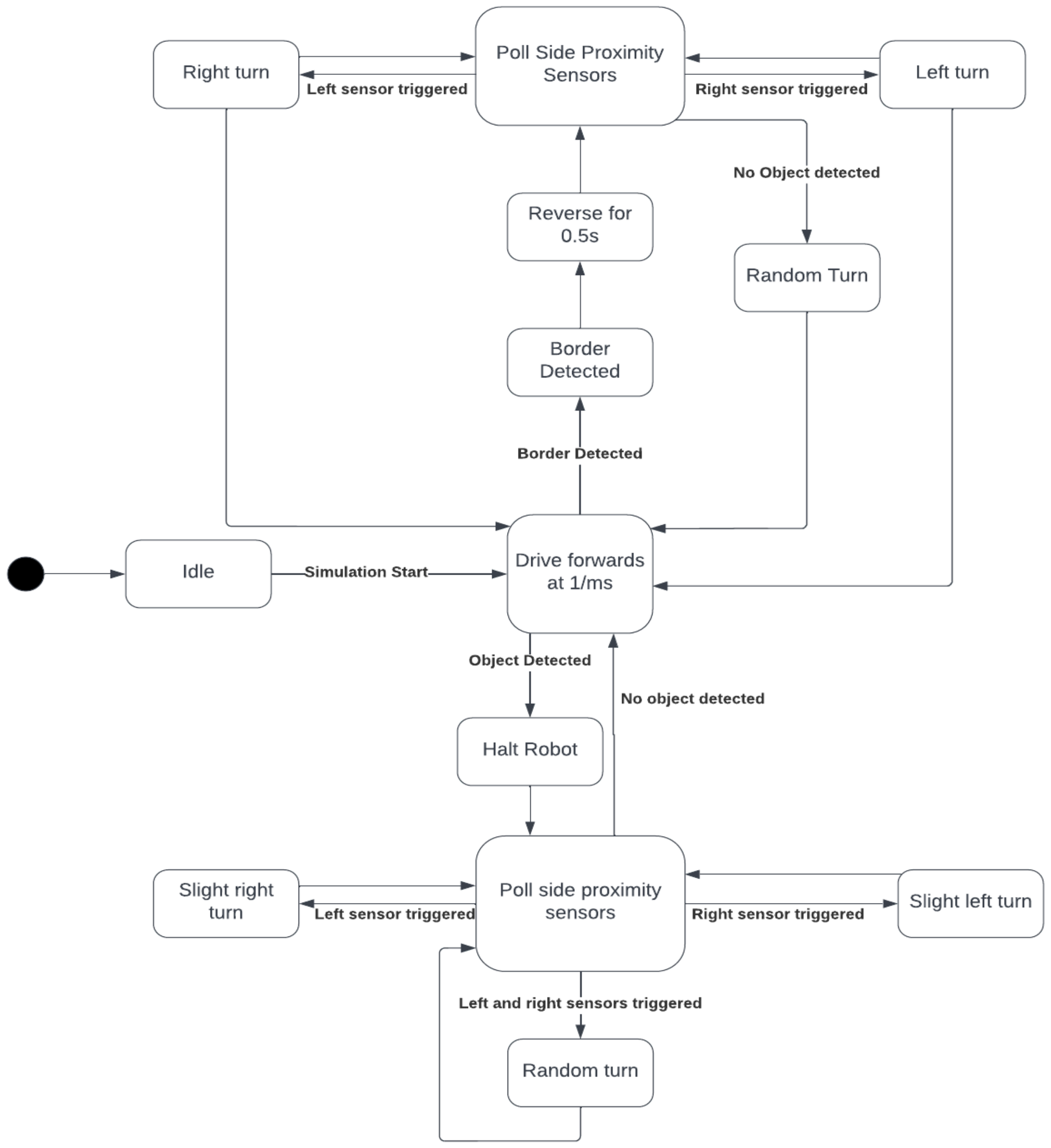
The robot body consists of a main body, 4 wheels, the plow, and a variety of sensors all tasked with sensing different information. A vision sensor is beneath the body which detects the black border surrounding the area, two ultrasonic sensors are placed above the plow for detecting obstacles in front of the robot, and one ultrasonic sensor is placed on each side of the body for detecting obstacles on the sides of the robot. The main body of the plow houses these sensors in the correct positions and ensures they have the proper field of view. Furthermore, the main robot body conforms to the specified size requirements of 0.5m x 0.8m x 1.0m.

System Triggers

Our system uses a time triggered system as the basis for its functionality, polling the sensors at regular intervals to determine if any action needs to be taken. The default behaviour of the bot consists of it driving around, following a pathing algorithm to optimize the amount of snow cleared. If the proximity sensors detect an obstacle, a separate process handler will be used to control the actions of the bot until the obstacle is no longer detected, at which point the bot will resume its default behaviour. The default pathing algorithm will drive the robot straight until it reaches a border or obstacle. The goal is for each successive pass to clear more and more snow from the track.

State Chart

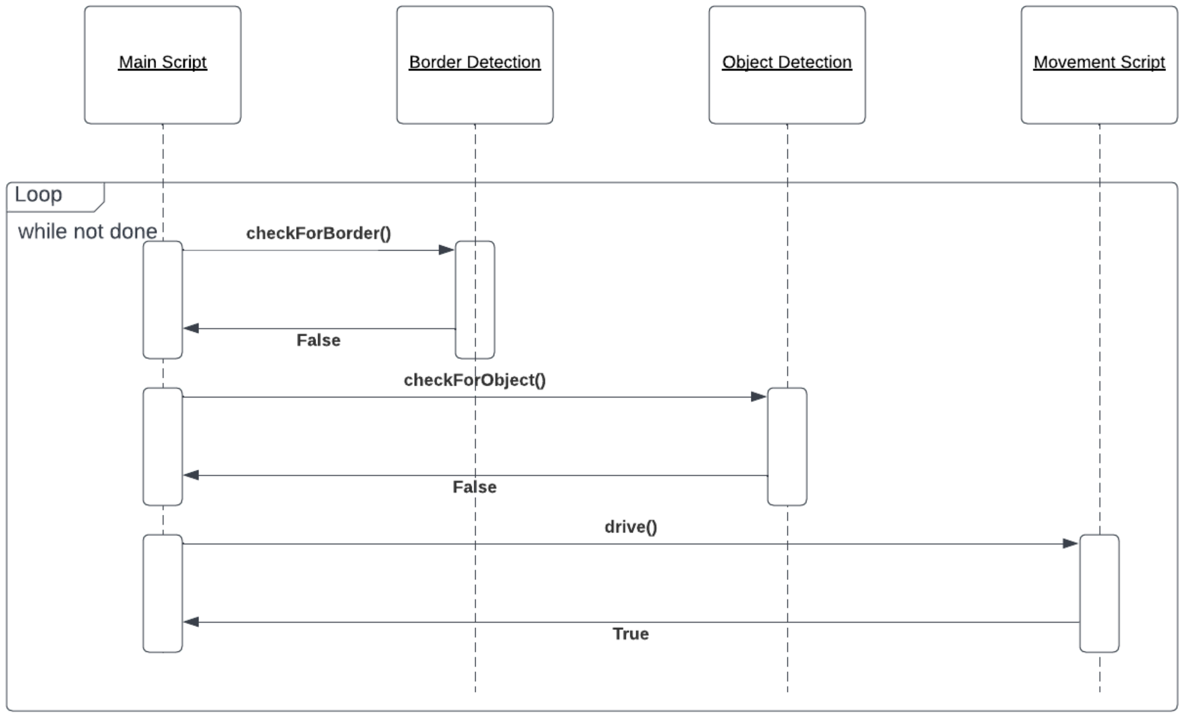
The state chart for the system is shown below. The system defaults to drive at approximately 1 meter per second forward, then if an object or border is detected the robot will handle it, then return to driving. The main difference between the handling of borders and the handling of objects is that when an object is detected there is a possibility of utilizing incremental turns to minimize snow loss, whereas when a border is detected incremental turning is not necessary.



Sequence Diagrams

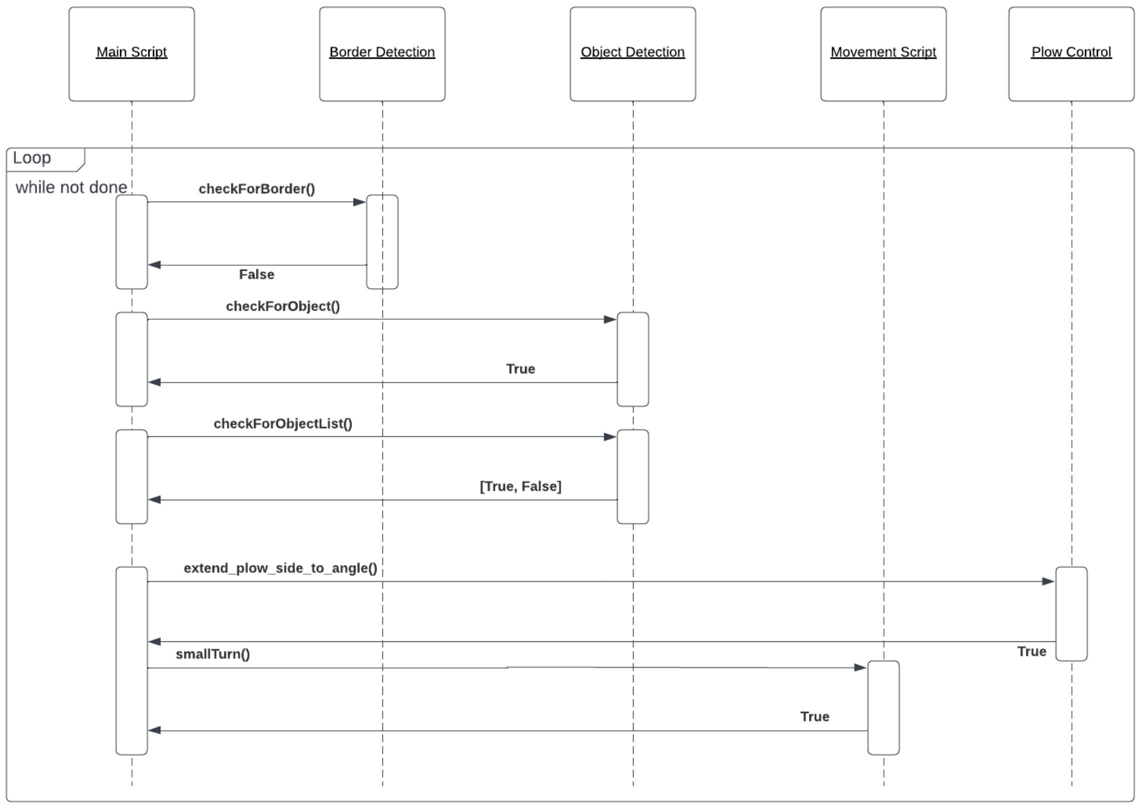
As seen in the Overall Architecture section, the system contains five discrete components. Each of these components are represented in the following sequence diagrams. Each diagram represents a different situation that the robot could encounter. These are the cases where nothing is detected, when an object is detected, and when a border is detected along with an object to the side.

Sequence Diagram 1: No Border or Object Detected



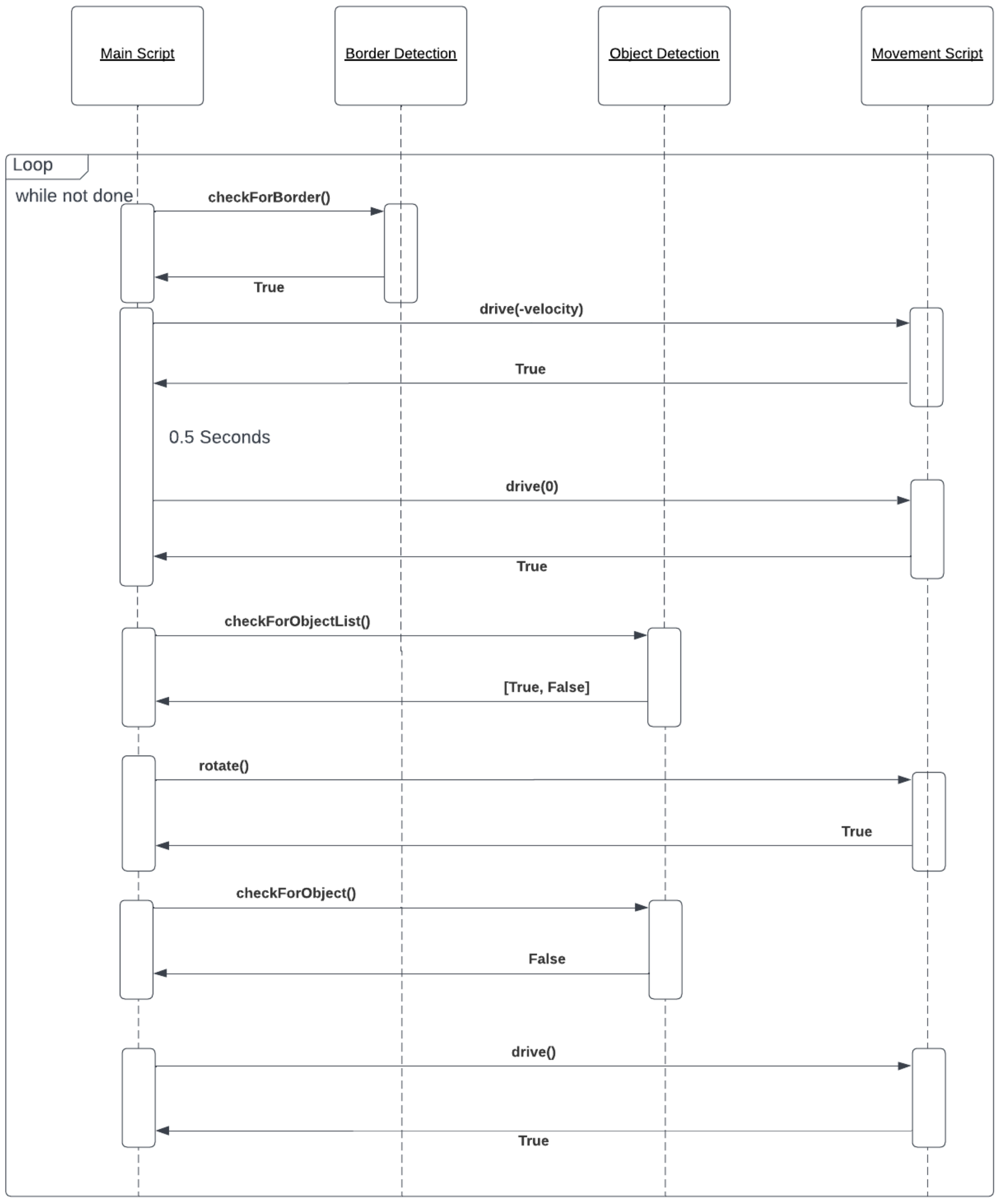
In this case, the main script will first check for a border and this check will return false. Next, the main script will check for objects in the path of the robot. This will also return false. In this scenario since there are no issues to handle, the main script will drive the robot forwards.

Sequence Diagram 2: Object Detected



Similarly to the first case, the main script will first check for a border, and this check will be false. Next, the main script will check for an object in the path of the robot. This check returns true. After this, the main script will halt the robot by setting the robot's velocity to 0. Next it will check the side proximity sensors using the checkForObjectList function. This function will return a list containing a boolean for each side proximity sensor. The boolean will be true if there is an object in that sensor's view. Using this information the main script will decide a direction to rotate. The plow will be angled to hold as much snow as possible, then an incremental turn will occur. This will loop until the front proximity sensors are clear.

Sequence Diagram 3: Border detected with side object



In this scenario the robot will encounter the border, and will have an object to its side as well. The first check the main script will perform is to check for a border. This check will return true, as in this scenario the robot is on the border. Once the border is detected, the main script will reverse the robot for 0.5 seconds, then halt. Next the main script will check the side proximity sensors. Based on the information returned, the main script will determine which direction to rotate, then do so. After rotating the border has been handled, so the main script will continue and check for objects, then drive forward as normal.

Testing

The following tests were run to ensure that the robot meets all requirements as stated above, and also performs as designed. This table of tests describes each test setup and how the execution occurs, and also gives a condition which must be true to consider the test a pass. These tests were done on all maps.

Tests:	Description	Pass Condition	Pass/Fail
1	Max Speed	<2.0 m/s	🟩
2	Border Detection	Robot able to detect border	🟩
3	Border Handling	Robot will detect border, reverse to designated area and then rotate	🟩
4	Non-White Border	Robot able to distinguish white tile as part of border	🟥 Robot is not able to detect the white starting area differently from the rest of the map floor, more sensors can alleviate this
5	Snow plowing	Robot able to push the snow using the plow	🟩
6	Snow plow dimensions	Snow plow meets dimension requirements	🟩
7	Extended plow dimensions	Extended plew meets dimension requirements	🟩
8	Plow moving	Plow able extend and contract	🟩
9	Object Detection	Robot able to detect objects	🟩
10	Avoiding objects	Robot able to avoid objects	🟩
11	Human detection	Robot able to detect Humans	🟩
12	Human Avoidance	Robot able to avoid Humans	🟥 Robot may get hit by the human and in some cases it turns into them
13	Determining human or object	Robot able to distinguish and avoid both Human and Objects	🟥 The robot does not distinguish between humans and objects
14	Corner border behaviour	Robot should be able to recognise and alter its path accordingly	🟩
15	Double object side by side	Robot able to detect both objects and still not hit them	🟩
16	Going maximum speed to border	Robot should be able while going at maximum speed to detect a border and turn	🟩

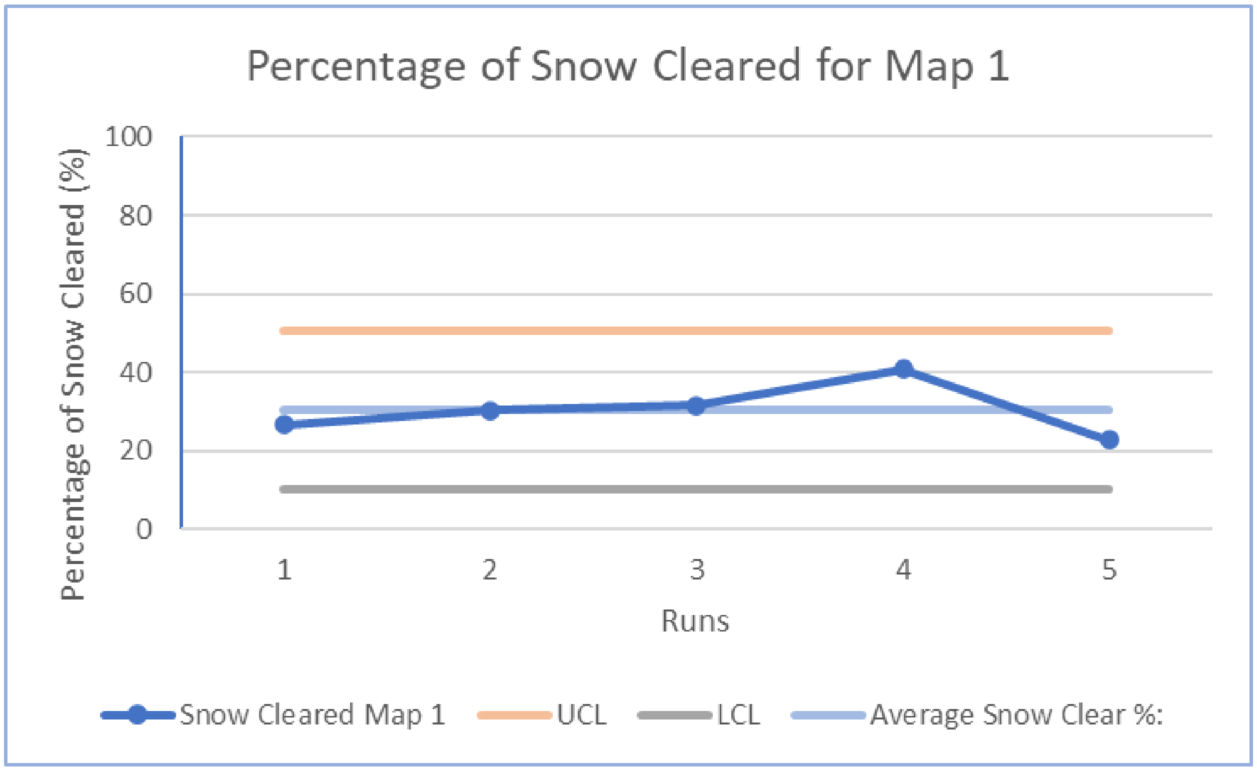
The following tests are done on maps 1-4. These tests include the average percentage clear and percentage hits on either an object or a human. These tests were run on each map a total of 5 times.

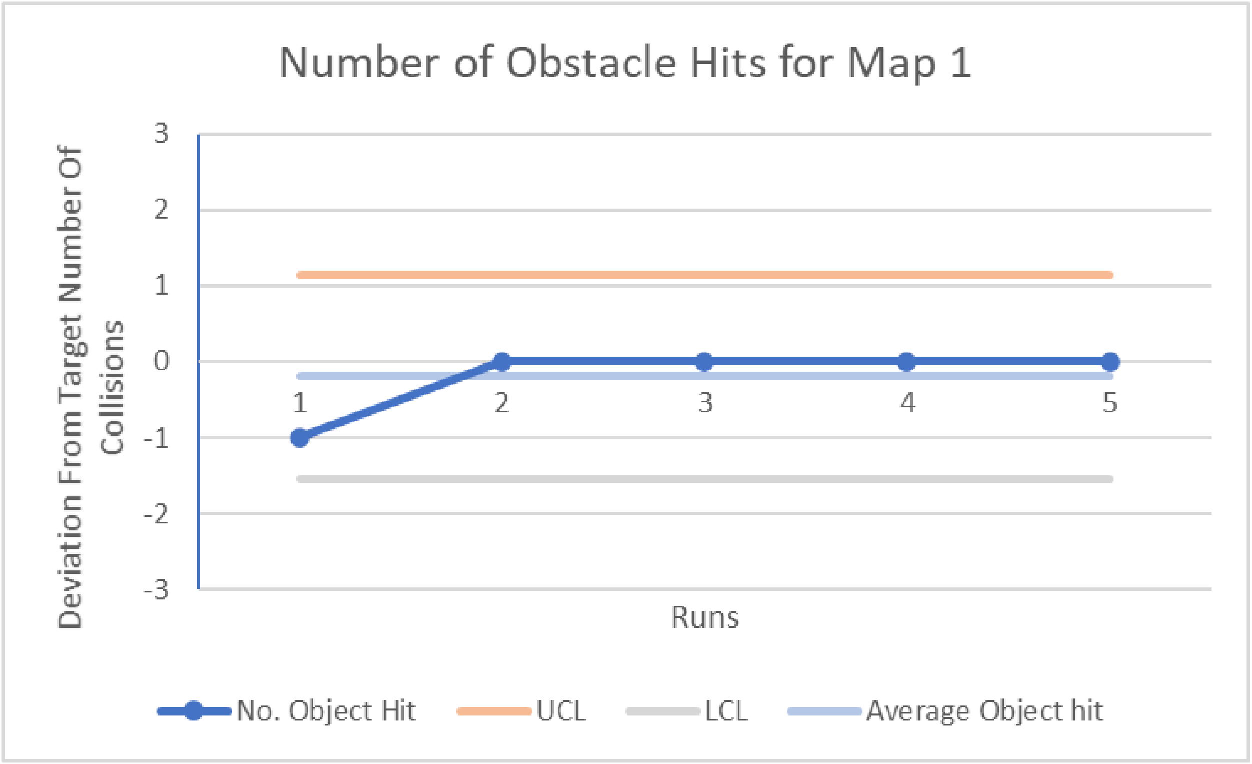
Map	Results	Values (Average)
1	Snow Clear percentage	30.4%
1	Objects hit	0.2
1	Human hit	0
2	Snow Clear percentage	33.78
2	Objects hit	0.4
2	Human hit	0
3	Snow Clear percentage	31.34
3	Objects hit	0.2
3	Human hit	0.6
4	Snow Clear percentage	27.14
4	Objects hit	0.2
4	Human hit	0

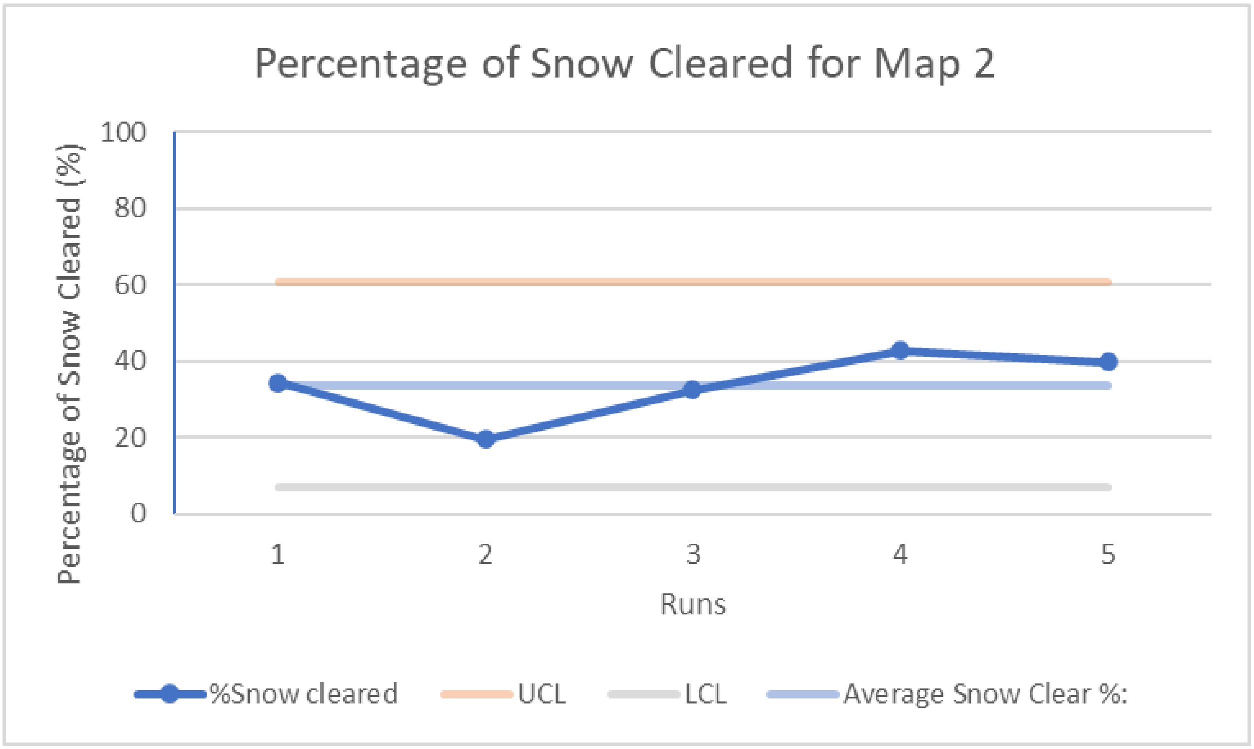
Based on the data seen above, it can be concluded that map 2 is the best on snow clearing for our plow robot. However, while this map does have the most snow cleared, it also has an average of 0.4 object hits per run which is higher than other maps. Map 3 has the second highest snow clear percentage but it has the highest obstacle hit rate between all runs. Due to the obstacle avoidance requirement being of higher priority than snow clearance, map 1 is our best map since it has hit an object only once and has cleared 30.4% of the snow. Meanwhile map 4 is our worst clear at 27.14%. More information on this data will be shown in the next section.

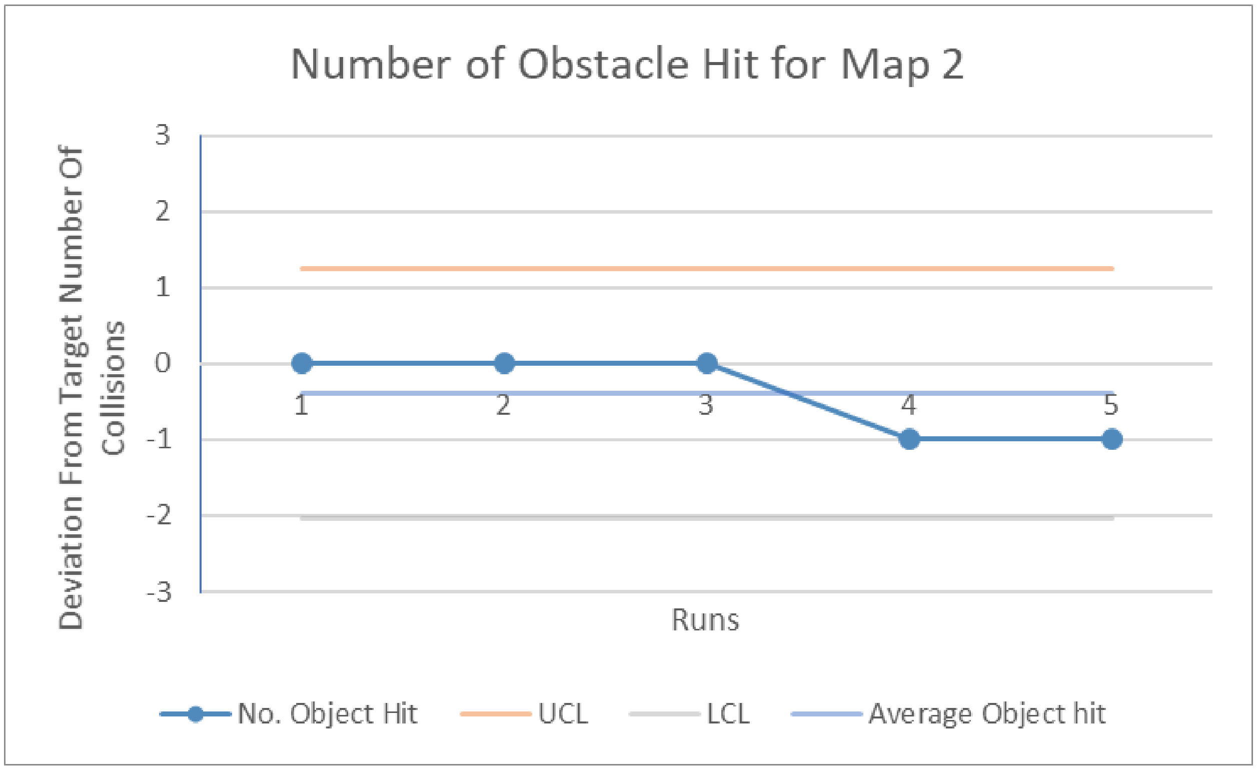
Control Charts

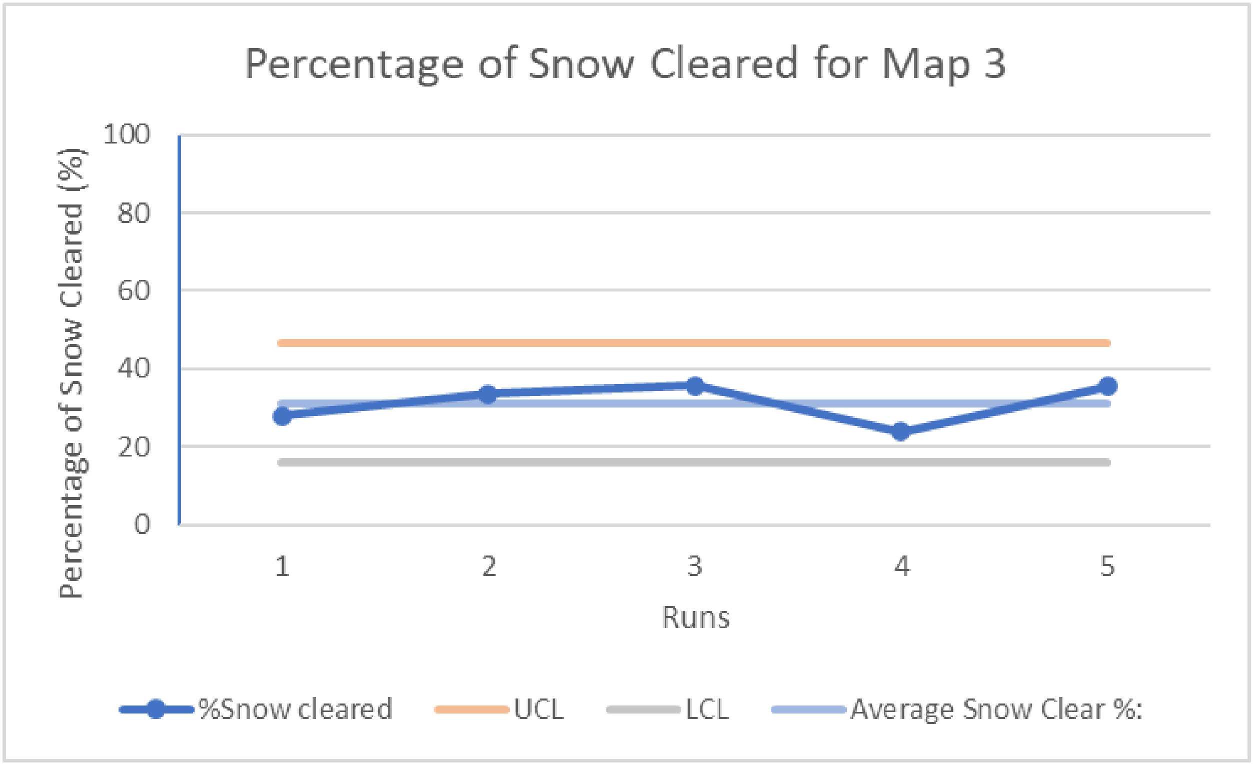
The following control charts document the results for our various requirements and the associated tests. For these charts, we found the standard deviation (sigma) for each set of data, then added 3*sigma to find the upper limit and subtracted 3*sigma to find the lower limit (LCL). For obstacle collision tests, the graphs are represented in deviation from the target value of zero collisions. The UCL and LCL were found using 3 sigma added or subtracted respectively.

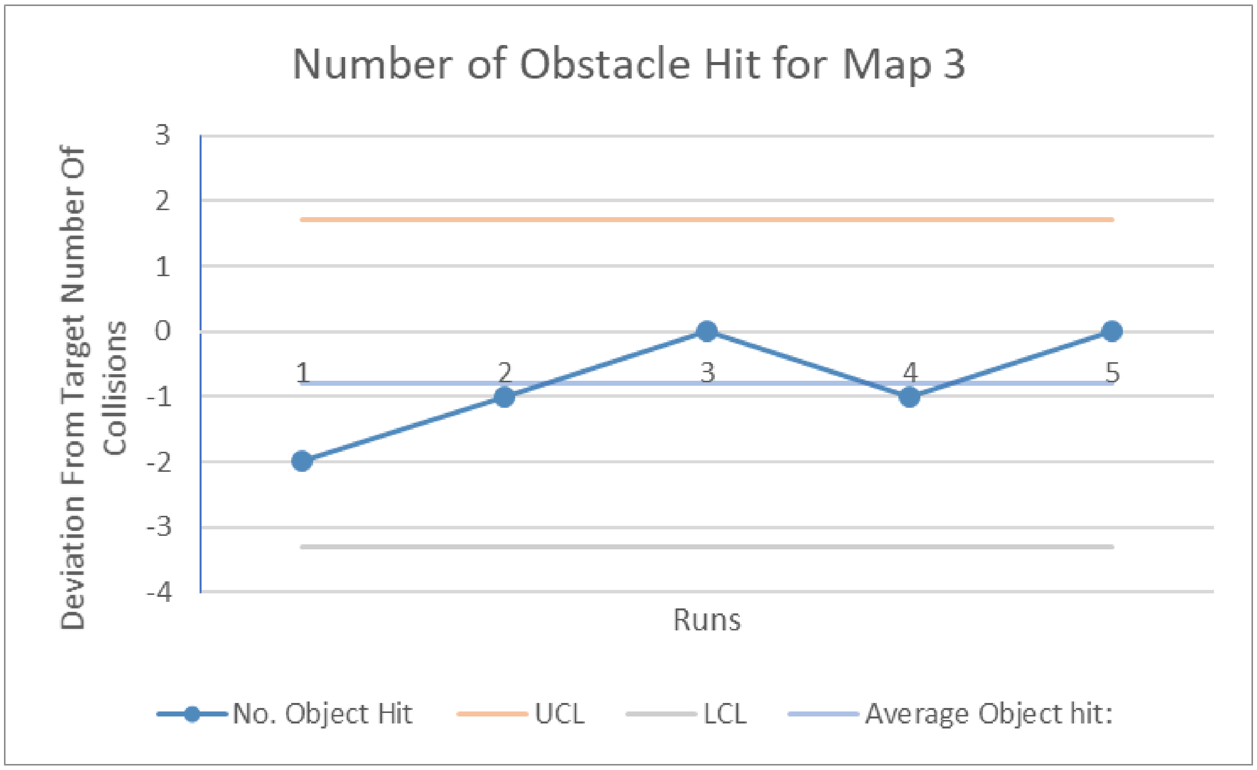

Percentage of Snow Cleared for Map 1
Percentage of Snow Cleared (%)
100
80
60
40
20
0
1
2
3
4
5
Runs
Snow Cleared Map 1
UCL
LCL
Average Snow Clear %:

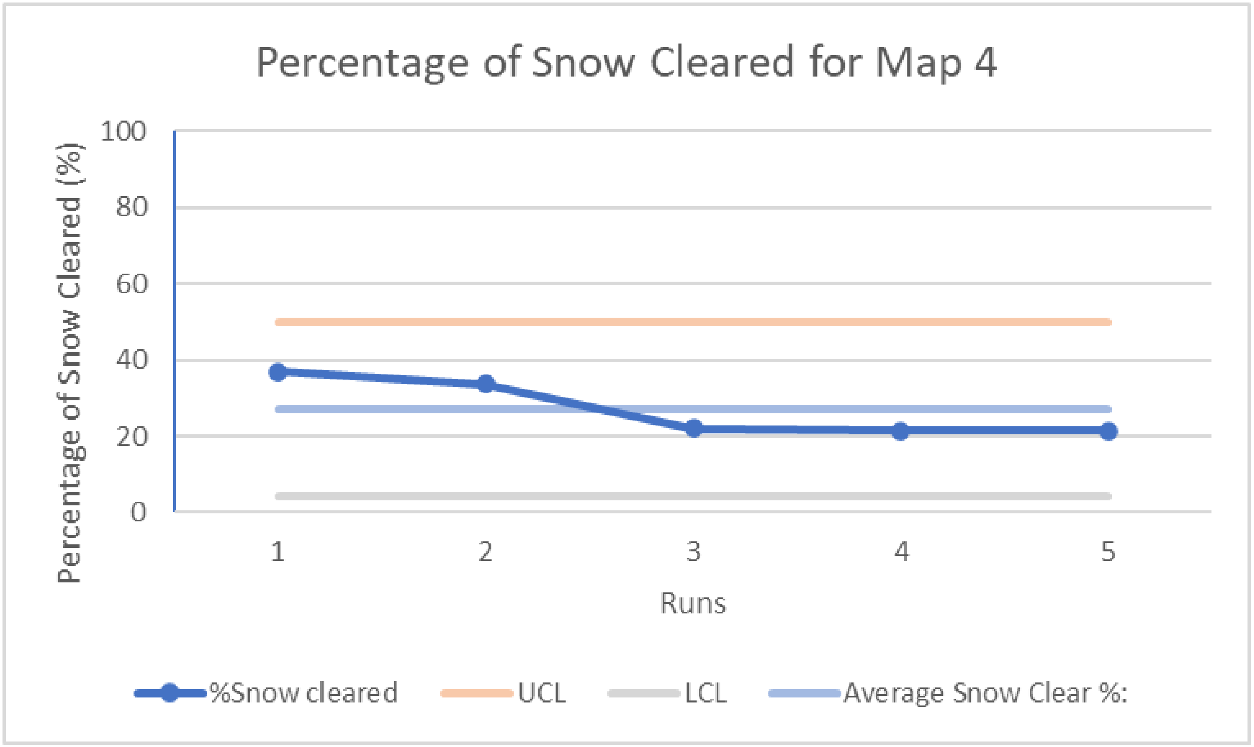

Number of Obstacle Hits for Map 1
Deviation From Target Number Of Collisions
3
2
1
0
-1
-2
-3
1
2
3
4
5
Runs
No. Object Hit
UCL
LCL
Average Object hit

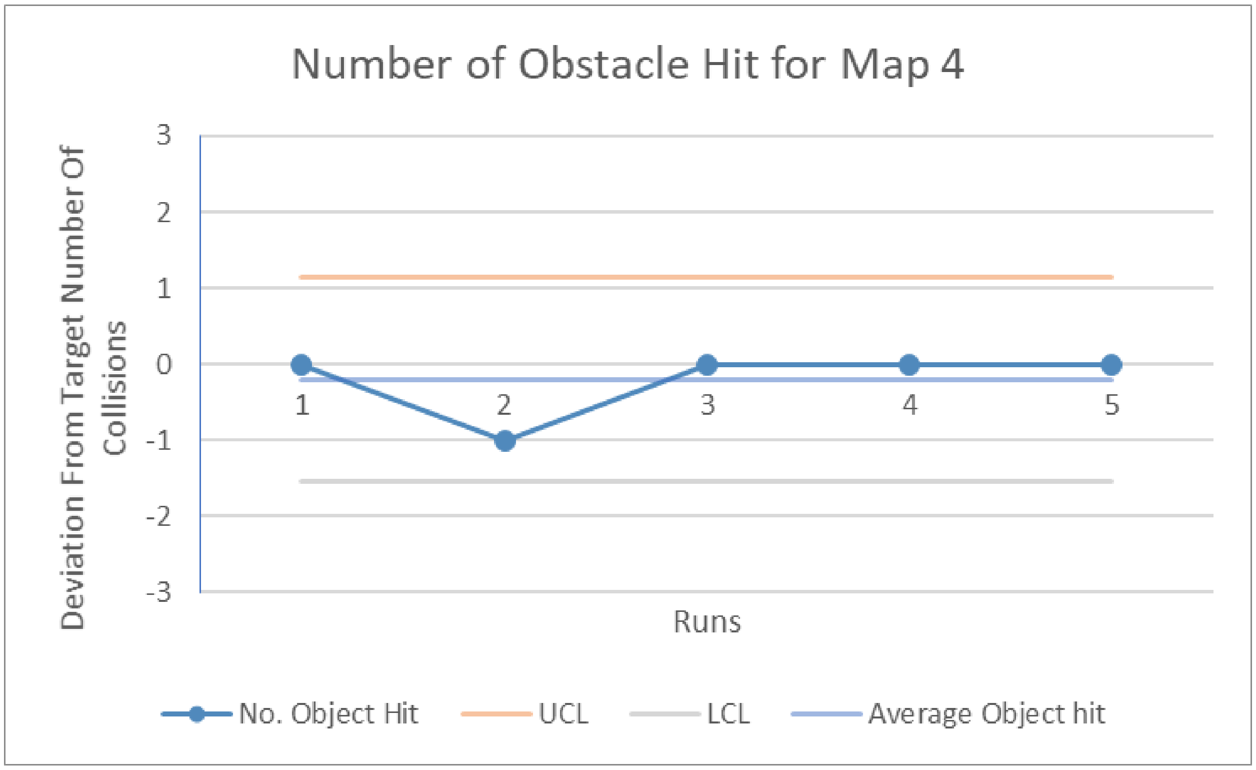

Percentage of Snow Cleared for Map 2
Percentage of Snow Cleared (%)
100
80
60
40
20
0
1
2
3
4
5
Runs
%Snow cleared
UCL
LCL
Average Snow Clear %:


Number of Obstacle Hit for Map 2
Deviation From Target Number Of Collisions
3
2
1
0
-1
-2
-3
1
2
3
4
5
Runs
No. Object Hit
UCL
LCL
Average Object hit


Percentage of Snow Cleared for Map 3
Percentage of Snow Cleared (%)
100
80
60
40
20
0
1
2
3
4
5
Runs
%Snow cleared
UCL
LCL
Average Snow Clear %:


Number of Obstacle Hit for Map 3
Deviation From Target Number Of Collisions
3
2
1
0
-1
-2
-3
-4
1
2
3
4
5
Runs
No. Object Hit
UCL
LCL
Average Object hit:

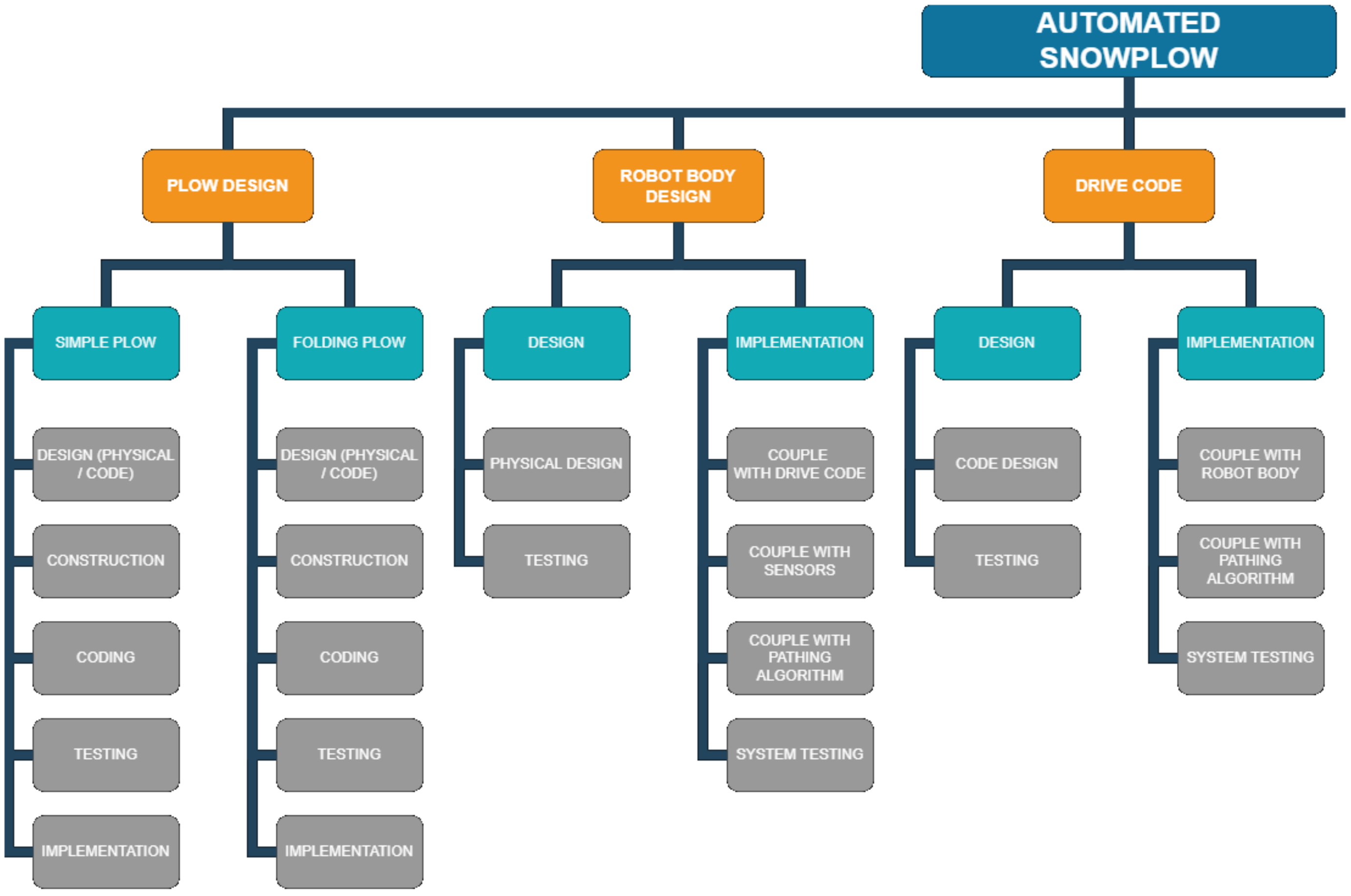

Percentage of Snow Cleared for Map 4
Percentage of Snow Cleared (%)
100
80
60
40
20
0
1
2
3
4
5
Runs
%Snow cleared
UCL
LCL
Average Snow Clear %:

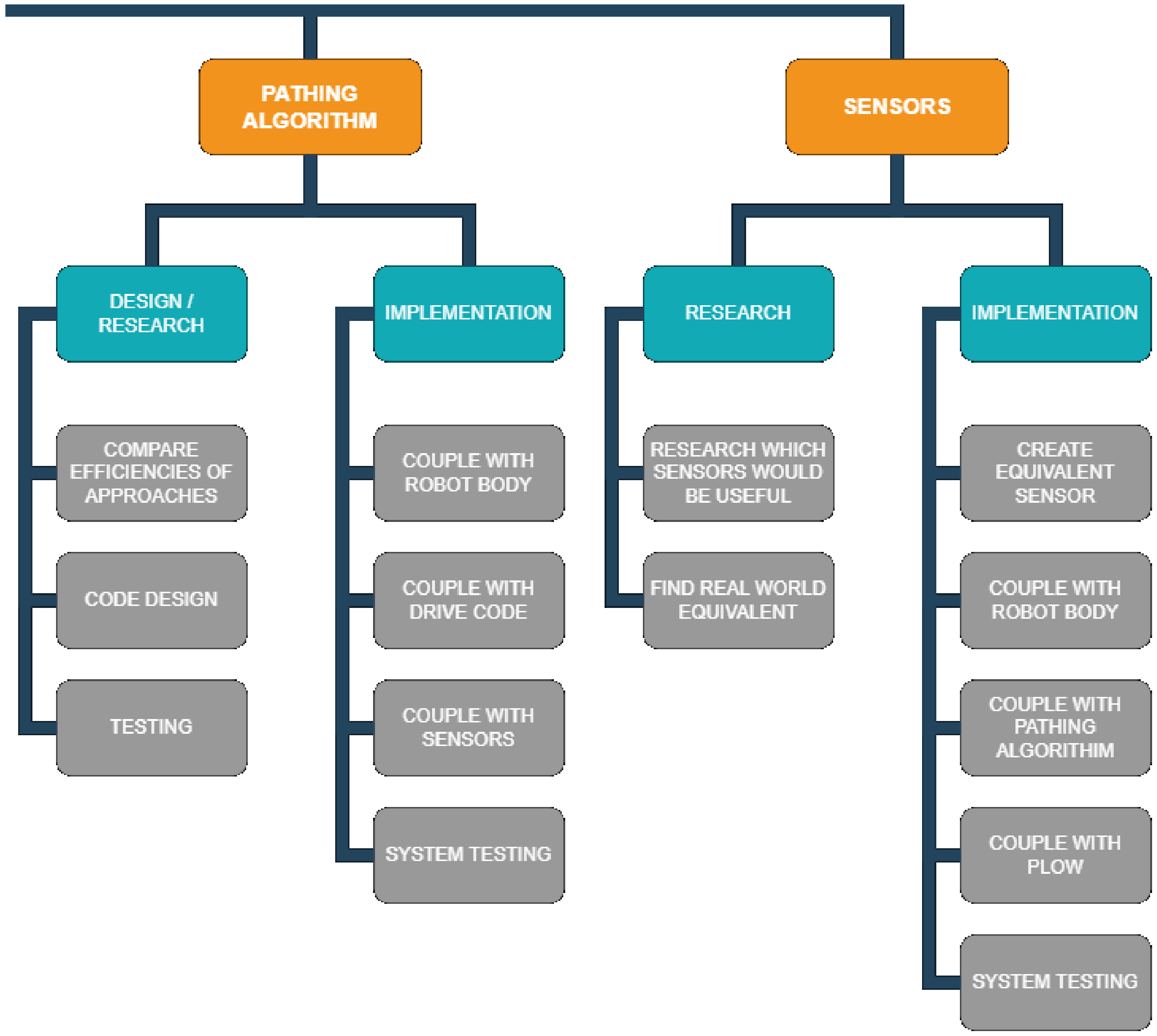

Number of Obstacle Hit for Map 4
Deviation From Target Number Of Collisions
3
2
1
0
-1
-2
-3
1
2
3
4
5
Runs
No. Object Hit
UCL
LCL
Average Object hit

Schedule

Work Breakdown Structure

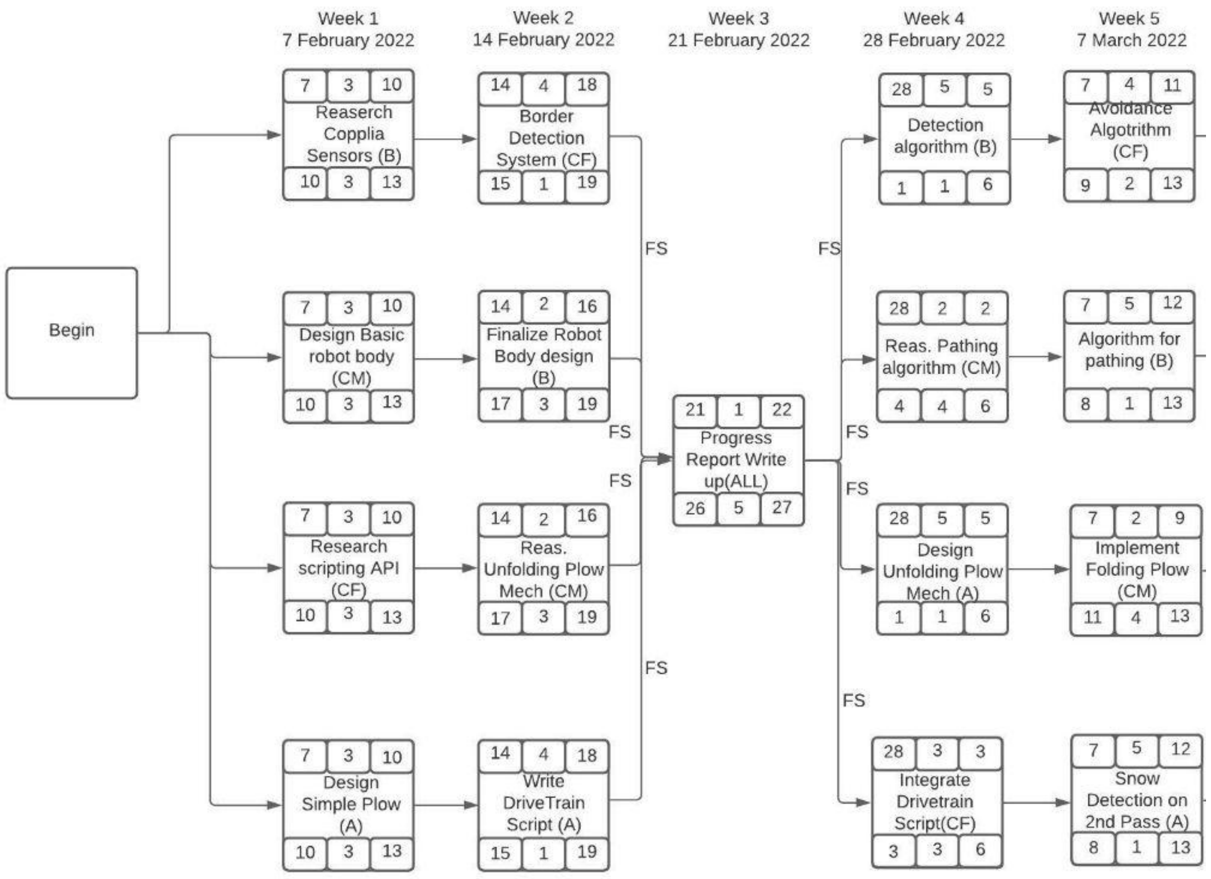
The following work breakdown structure was used to inform the breakdown of tasks for the project.

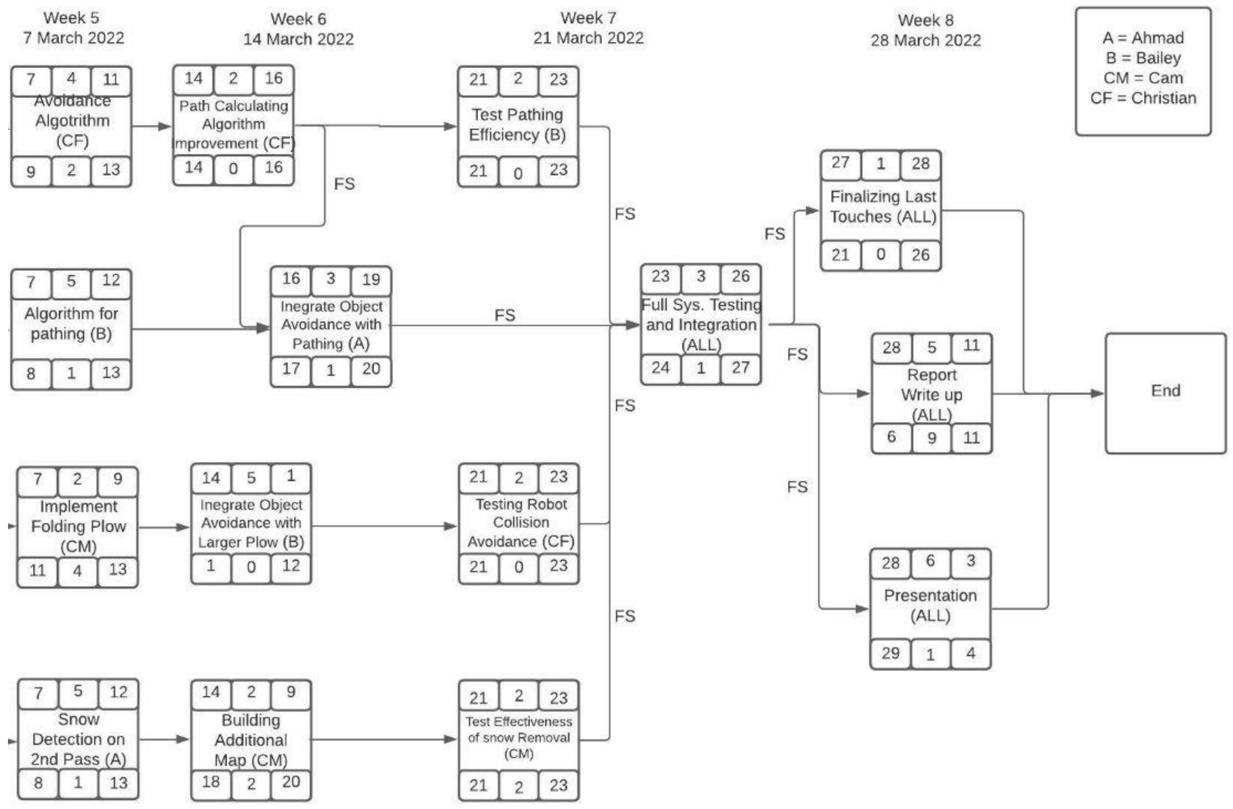



PATHING ALGORITHM
SENSORS
DESIGN / RESEARCH
COMPARE EFFICIENCIES OF APPROACHES
CODE DESIGN
TESTING
IMPLEMENTATION
COUPLE WITH ROBOT BODY
COUPLE WITH DRIVE CODE
COUPLE WITH SENSORS
SYSTEM TESTING
RESEARCH
RESEARCH WHICH SENSORS WOULD BE USEFUL
FIND REAL WORLD EQUIVALENT
IMPLEMENTATION
CREATE EQUIVALENT SENSOR
COUPLE WITH ROBOT BODY
COUPLE WITH PATHING ALGORITHIM
COUPLE WITH PLOW
SYSTEM TESTING

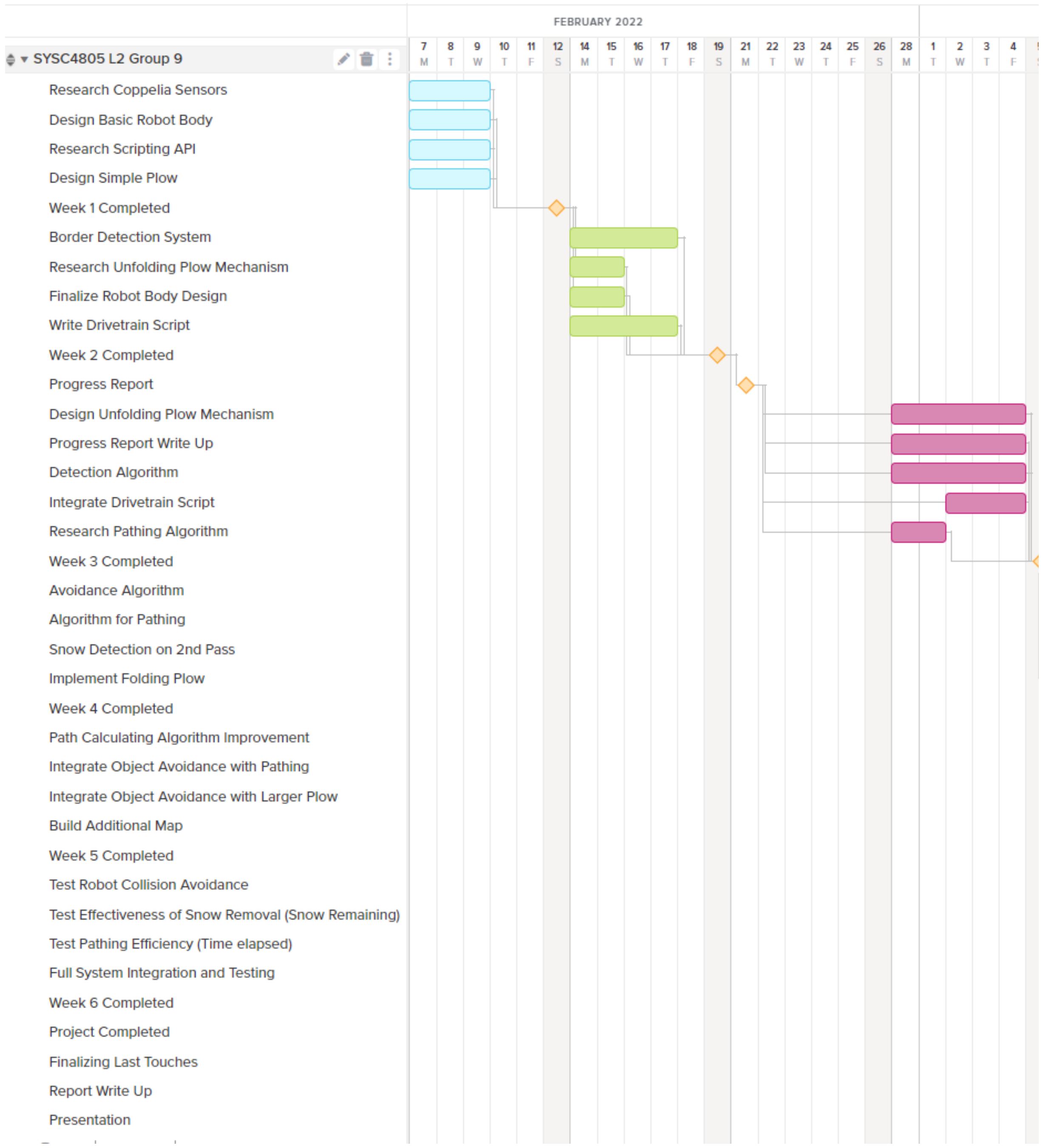
Schedule Network Diagram

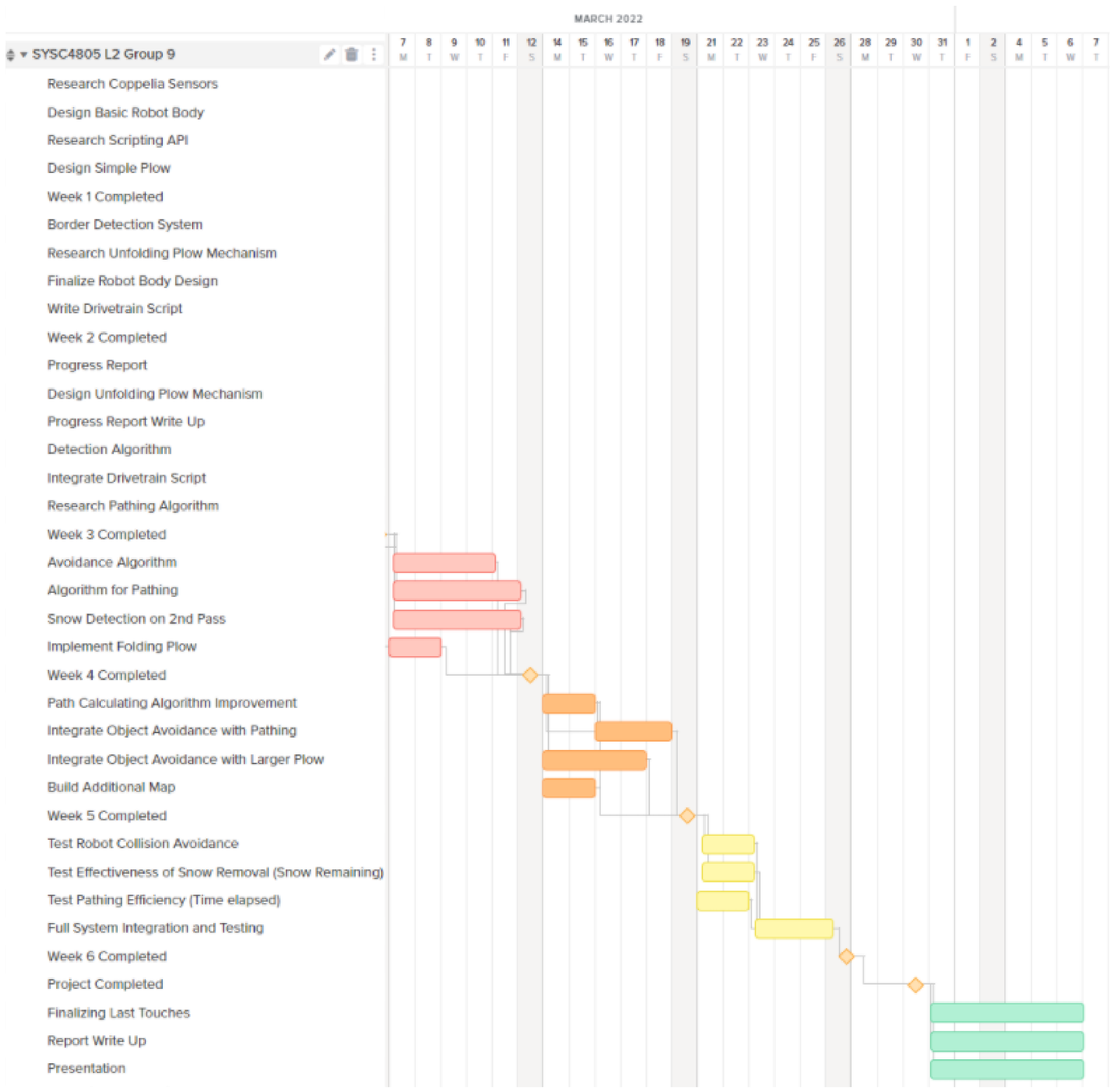
The following diagram shows the breakdown of each week's tasks along with the assigned group member. Each node is labeled with start and end dates along with the activities name.


Week 1
7 February 2022
Week 2
14 February 2022
Week 3
21 February 2022
Week 4
28 February 2022
Week 5
7 March 2022
Begin
7 3 10 Reaserch Copplia Sensors (B) 10 3 13
7 3 10 Design Basic robot body (CM) 10 3 13
7 3 10 Research scripting API (CF) 10 3 13
7 3 10 Design Simple Plow (A) 10 3 13
14 4 18 Border Detection System (CF) 15 1 19
14 2 16 Finalize Robot Body design (B) 17 3 19
14 2 16 Reas. Unfolding Plow Mech (CM) 17 3 19
14 4 18 Write DriveTrain Script (A) 15 1 19
21 1 22 Progress Report Write up(ALL) 26 5 27
28 5 5 Detection algorithm (B) 1 1 6
28 2 2 Reas. Pathing algorithm (CM) 4 4 6
28 5 5 Design Unfolding Plow Mech (A) 1 1 6
28 3 3 Integrate Drivetrain Script(CF) 3 3 6
7 4 11 Avoidance Algotrithm (CF) 9 2 13
7 5 12 Algorithm for pathing (B) 8 1 13
7 2 9 Implement Folding Plow (CM) 11 4 13
7 5 12 Snow Detection on 2nd Pass (A) 8 1 13
FS
FS
FS
FS
FS
FS
FS
FS


Week 5
7 March 2022
Week 6
14 March 2022
Week 7
21 March 2022
Week 8
28 March 2022
A = Ahmad
B = Bailey
CM = Cam
CF = Christian
7 4 11 Avoidance Algotrithm (CF) 9 2 13
14 2 16 Path Calculating Algorithm Improvement (CF) 14 0 16
21 2 23 Test Pathing Efficiency (B) 21 0 23
7 5 12 Algorithm for pathing (B) 8 1 13
16 3 19 Inegrate Object Avoidance with Pathing (A) 17 1 20
23 3 26 Full Sys. Testing and Integration (ALL) 24 1 27
27 1 28 Finalizing Last Touches (ALL) 21 0 26
28 5 11 Report Write up (ALL) 6 9 11
28 6 3 Presentation (ALL) 29 1 4
End
7 2 9 Implement Folding Plow (CM) 11 4 13
14 5 1 Inegrate Object Avoidance with Larger Plow (B) 1 0 12
21 2 23 Testing Robot Collision Avoidance (CF) 21 0 23
7 5 12 Snow Detection on 2nd Pass (A) 8 1 13
14 2 9 Building Additional Map (CM) 18 2 20
21 2 23 Test Effectiveness of snow Removal (CM) 21 2 23
FS
FS
FS
FS
FS
FS
FS
FS

Gantt Chart


FEBRUARY 2022
7 M
8 T
9 W
10 T
11 F
12 S
14 M
15 T
16 W
17 T
18 F
19 S
21 M
22 T
23 W
24 T
25 F
26 S
28 M
1 T
2 W
3 T
4 F
SYSC4805 L2 Group 9
Research Coppelia Sensors
Design Basic Robot Body
Research Scripting API
Design Simple Plow
Week 1 Completed
Border Detection System
Research Unfolding Plow Mechanism
Finalize Robot Body Design
Write Drivetrain Script
Week 2 Completed
Progress Report
Design Unfolding Plow Mechanism
Progress Report Write Up
Detection Algorithm
Integrate Drivetrain Script
Research Pathing Algorithm
Week 3 Completed
Avoidance Algorithm
Algorithm for Pathing
Snow Detection on 2nd Pass
Implement Folding Plow
Week 4 Completed
Path Calculating Algorithm Improvement
Integrate Object Avoidance with Pathing
Integrate Object Avoidance with Larger Plow
Build Additional Map
Week 5 Completed
Test Robot Collision Avoidance
Test Effectiveness of Snow Removal (Snow Remaining)
Test Pathing Efficiency (Time elapsed)
Full System Integration and Testing
Week 6 Completed
Project Completed
Finalizing Last Touches
Report Write Up
Presentation


MARCH 2022
SYSC4805 L2 Group 9
7 M
8 T
9 W
10 T
11 F
12 S
14 M
15 T
16 W
17 T
18 F
19 S
21 M
22 T
23 W
24 T
25 F
26 S
28 M
29 T
30 W
31 T
1 F
2 S
4 M
5 T
6 W
7 T
Research Coppelia Sensors
Design Basic Robot Body
Research Scripting API
Design Simple Plow
Week 1 Completed
Border Detection System
Research Unfolding Plow Mechanism
Finalize Robot Body Design
Write Drivetrain Script
Week 2 Completed
Progress Report
Design Unfolding Plow Mechanism
Progress Report Write Up
Detection Algorithm
Integrate Drivetrain Script
Research Pathing Algorithm
Week 3 Completed
Avoidance Algorithm
Algorithm for Pathing
Snow Detection on 2nd Pass
Implement Folding Plow
Week 4 Completed
Path Calculating Algorithm Improvement
Integrate Object Avoidance with Pathing
Integrate Object Avoidance with Larger Plow
Build Additional Map
Week 5 Completed
Test Robot Collision Avoidance
Test Effectiveness of Snow Removal (Snow Remaining)
Test Pathing Efficiency (Time elapsed)
Full System Integration and Testing
Week 6 Completed
Project Completed
Finalizing Last Touches
Report Write Up
Presentation

Human Resources

Responsibility Assignment Matrix

The following figure shows the breakdown of each activity as well as the assigned group member along with the approver for the activity.

- Responsible = R   Approver = A

Week Number	Activity	Ahmad Mozan	Bailey Lyster	Cam Maccoll	Christian Fisher
1	Design a simple plow prototype to investigate interaction with the environment.	R			A
1	Research Coppelia Sim sensors and their real world equivalents.		R	A	
1	Design basic robot body to fit dimensions given.	A		R	
1	Research scripting API		A		R
2	Implement border detection sensor.			A	R
2	Research unfolding plowhead mechanics.		A	R	
2	Finalize robot body design.	A	R		
2	Write a drivetrain script.	R			A
3 (Reading week)	Progress Report write up	RA	RA	RA	RA
4	Design unfolding plow mechanism	R		A	
4	Implement object detection algorithm	A	R		
4	Research pathing algorithms		A	R	
4	Integrate drivetrain script			A	R
5	Implement object avoidance algorithm.	A			R
5	Implementation of a folding plow.		A	R	

5	Building an algorithm for pathing		R	A	
5	Figuring out method of snow detection for round 2 of cleaning.	R			A
6	Integrate object avoidance with pathing algorithm.	R	A		
6	Integrate object avoidance with larger size of folding plow.		R	A	
6	Build additional map for stress testing robot functionalities			R	A
6	Path calculating improvements	A			R
7	Testing robot collision avoidance		A		R
7	Testing Effectiveness of snow removal			R	A
7	Testing pathing efficiency		R	A	
7	Full system integration and testing	RA	RA	RA	RA
8	Finalizing last touches	RA	RA	RA	RA
8	Presentation preparation	RA	RA	RA	RA
8	Project report write up	RA	RA	RA	RA

Project Budget

As for the team budget first we will calculate the budget needed for the robot to be built.

Item No.	Item	Price (In CAD after taxes)
1	3 Aluminum sheets to build body and plow [1]	60.00
2	4 Mxfans Small Square Lattice Shape Tire [2]	20.00
3	4 HC-SR04 Ultrasonic Module Ranging Sensor [3]	39.60
4	1 KEYESTUDIO 5MP 1080p Camera Module [4]	21.30
5	ARDUINO A000066 Uno R3 [5]	32.00
6	6 Gikfun DC 3V-6V 140 Motor 2000 RPM [6]	14.68
7	Gecoty Ni-MH Battery Pack [7]	27.12
8	Snow x 500	0.00

Since we have 4 engineers working to finish the project, who are paid $21 an hour, working 10 hours a week for a duration of 7 weeks would yield a labour cost of $9996 CAD. Therefore, the average cost for this robot to be built is $214.7 CAD after taxes. Therefore, the final project budget is $10210.7 CAD.

Comparing the project budget to the original planned value of $10240 we had in our proposal this means that we went under budget (AV < EV). The project budget decreased from $244 CAD to $214.7 CAD, which was due to the motor being swapped to a different model which saved on some costs.

References

[1]
https://www.homedepot.ca/product/paulin-8-x-24-x-0-025-inch-aluminum-sheet-metal/1000126786
[2]
https://www.amazon.ca/Mxfans-Front-Road-Racing-Wheel/dp/B07H9BLMDF/ref=sr_1_12?crid=2PN8PH562WJJV&keywords=small%2Bplastic%2Bcar%2Bwheels&qid=1646075533&s=toys&sprefix=small%2Bplastic%2Bcar%2Bwheels%2Ctoys%2C60&sr=1-12
[3]
https://www.amazon.ca/HC-SR04-Ultrasonic-Ranging-Arduino-MEGA2560/dp/B09PZDFP94/ref=sr_1_6?crid=158NNVF4H4MFK&keywords=HC-SR04+Module&qid=1646076298&sprefix=hc-sr04+module%2Caps%2C207&sr=8-6
[4]
https://www.amazon.ca/Keyestudio-Camera-Module-5MP-Raspberry/dp/B073RCXGQS/ref=sr_1_7?keywords=rpi+camera&qid=1646076338&sprefix=rpi+camera%2Caps%2C64&sr=8-7
[5]
https://www.amazon.ca/ARDUINO-A000066-Uno-DIP-1-5/dp/B008GRTSV6/ref=sr_1_7?crid=LPGX7O7G3B72&keywords=arduino&qid=1646076518&sprefix=arduino%2Caps%2C76&sr=8-7
[6]
https://www.amazon.ca/Gikfun-3V-6V-Electric-Arduino-EK2153/dp/B01JG8DIYQ/ref=sr_1_33?crid=LCC9UREUXW7Z&keywords=dc+motor&qid=1649709416&sprefix=dc+motor%2Caps%2C62&sr=8-33
[7]
https://www.amazon.ca/Gecoty-Rechargeable-Lighting-Security-Facilities/dp/B073SX3QDF/ref=sr_1_11?crid=31L2FEVT1N4Q4&keywords=small+toy+car+battery&qid=1646076782&sprefix=small+toy+car+battery%2Caps%2C62&sr=8-11