Sprint 3 - Agility Design Document

April 18, 2024

Table of Contents

1. Executive Summary

1.1 Project Overview

This project is for CS104 and is an agility run to test the Sphero robot for both accuracy and speed for completing a full obstacle course. The robot must accurately follow the blue lines, avoiding the obstacles. It must jump over the ramp, then proceed to knock down as many of the pins (markers) as possible.

1.2 Purpose and Scope of this Specification

Intended audience

- Computer Scientists
- Software Engineers
- Robotic Engineers
- Class of Computer Science

In Scope

This document addresses the parts of the Sphero Program

- User Interface
- Java program of robot
- Errors and fixes to Sphero
- Planning and roles of members involved

Out of Scope

This document addresses parts not mentioned on Project or Sphero.

- Practical uses for Sphero
- System requirements
- Background information on members
- Spheros Origins
- Gives us a good idea for our future sprints

2. Product/Service Description

This section describes the factors that affect the Sphero product and its requirements.

- **Public Use:** Sphero is out for the general public for young audiences to gain an understanding in coding, science, music, and the arts.
- **Internet Connection**: in order to pair with the Sphero and access Sphero EDU, proper internet connection is necessary, otherwise, nothing can work.
- **Activities:** Sphero has many activities for people of all ages through Sphero Edu app and littleBits Classroom. Challenges can be completed through STEAM as well.
- **Room Availability:** We had to be able to adapt to the times that the room was being used.

2.1 Product Context

The device used, Sphero Sprk, is a part of a broader system of different Sphero robots. The other robots include, the Sphero Bolt, Sphero Mini, Sphero RVR+, Sphero Indi, and many more. All of these robots are connected through the same program, Sphero EDU, where types of Spheros can be paired and coded.

2.2 User Characteristics

This section describes the scope of the users.

- Students: Students will have little to no prior experience in Sphero and will use challenge activities to help them learn on the Sphero EDU app. Many tests and understanding of code will be gained through application problems. Student range however is just as broad when the learning curve of Sphero is understood.
- Professor: The Professor will have much more insight on Sphero and its capabilities. He or she will be able to assist others in learning Sphero and can possibly gain insight as well from teaching.
- Staff: Staff members overseeing the product will have more insight than students but potentially less insight in the functionality of Sphero than the professor. As the staff work through the project they will be able to understand the product more.

2.3 Assumptions

- Sphero charged: It is assumed that the Sphero is properly charged, it will take longer to do the testing and coding phase if not and charging will be required.
- Sphero EDU program: It is assumed that the Sphero EDU program should run properly but flaws or bugs are still plausible. This can affect the testing and coding phase, as well as the overall performance in the agility run. Multiple tests may be needed to perform and record a video that is fit for the requirements of the project.

- Phone and Computer availability: It is assumed that phones and computers will be available at all times. If either were broken or dead, the project would not be able to be completed in any aspect for planning, testing, coding, or videotaping.

2.4 Constraints

- Sphero availability: Only one member of the group is able to have the Sphero at the same time. If one member has it while being unavailable, the other members can not perform tests.
- If the security of the system is breached the program may be compromised and setbacks will occur due to privacy and protection.
- System resource constraints: Computers may not have enough storage to download the necessary Sphero EDU app. This will result in delays.
- Sphero mapping and consistency: Sphero is known to have faulty mapping in terms of its direction and turning. Sometimes turning is factored into the time and slows down the overall process.

2.5 Dependencies

Dependencies are what the program relies on to function

- For the system to run, the algorithm and the flow chart must be designed for a full plan
- Sphero will require either a computer or phone to run the program

3. Requirements

There are a few requirements that we must meet for the project to be successful. The Sphero robot must start at the origin (priority 1). The robot must follow along the blue lines #1, #2, #3, #4, and #5, avoiding the obstacles #1, #2, and #3 (priority 1). The robot must successfully jump over the ramp (priority 1). The robot must knock down as many pins as possible when it reaches the end (priority 2). Extra points will be added for each pin knocked down and obstacles avoided.

3.1 Functional Requirements

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
AGILITY_01	Robot begins at starting point		1	4/17	Approved
AGILITY_02	Robot follows the blue line, avoiding obstacle #1, stopping at the end.		1	4/17	Approved
AGILITY_03	Robot continues across the second blue line avoiding obstacle #1 and #2, stopping at the end.		1	4/17	Approved
AGILITY_04	Robot continues across the third blue line avoiding obstacle #2 and obstacle #3, stopping at the end.		1	4/17	Approved
AGILITY_05	Robot continues across the fourth blue line, avoiding obstacle #3 and		1	4/17	Approved

	jumping over the ramp ending on the blue line.				
AGILITY_06	The robot turns and goes across the fifth blue line, knocking over as many pins as possible.		1	4/17	Approved
AGILITY_07	Robot ends		1	4/17	Approved

3.2 Security

3.2.1 Protection

There are many factors that protect the system from malicious or accidental access, modification, disclosure, destruction, or misuse.

- **Data protection and encryption**: All data is password protected and accounts are stored in a secure user database accessible only by administrators to Sphero EDU.
- **Privacy Policy**: No data is used for advertising. Data is only stored for app analytics. First and last names are optional .Sphero EDU does not use any cookies for marketing, tracking, or analytics purposes.
- **Age restrictions**: Users under 13 need parental permission to access Sphero EDU, children otherwise can not access Sphero EDU.
- **Data integrity checks**: Are implemented to maintain that everyone working on the program states who they are, what they add, and when. This ensures that no major errors are made, or at the very least are identifiable. It also ensures that the proper staff are working on the project and no malicious third party.

3.2.2 Authorization and Authentication

- When the project started, we were given Authorization by Professor Eckert to take the robot by signing it off.
- Github requires a sign before placing anything into the repository.

3.3 Portability

- **Host Dependent Code:** The code we do in SpheroEDU is 100% host dependent, but the app can be on every single device
- Our laptops are portable
- Our Sphero Robot is portable
- Code is only shared via GitHUB and pictures through iMessage
- The product works the same regardless of operating systems and devices with the same code

4. Requirements Confirmation/Stakeholder sign-off

Meeting Date	**Attendees (name and role)**	**Comments**
04/17/2024	Anthony S, Kooper K, Esha A	Discussed final design document roles, in addition to recording the video of the code implemented for sprint 3.

5. System Design

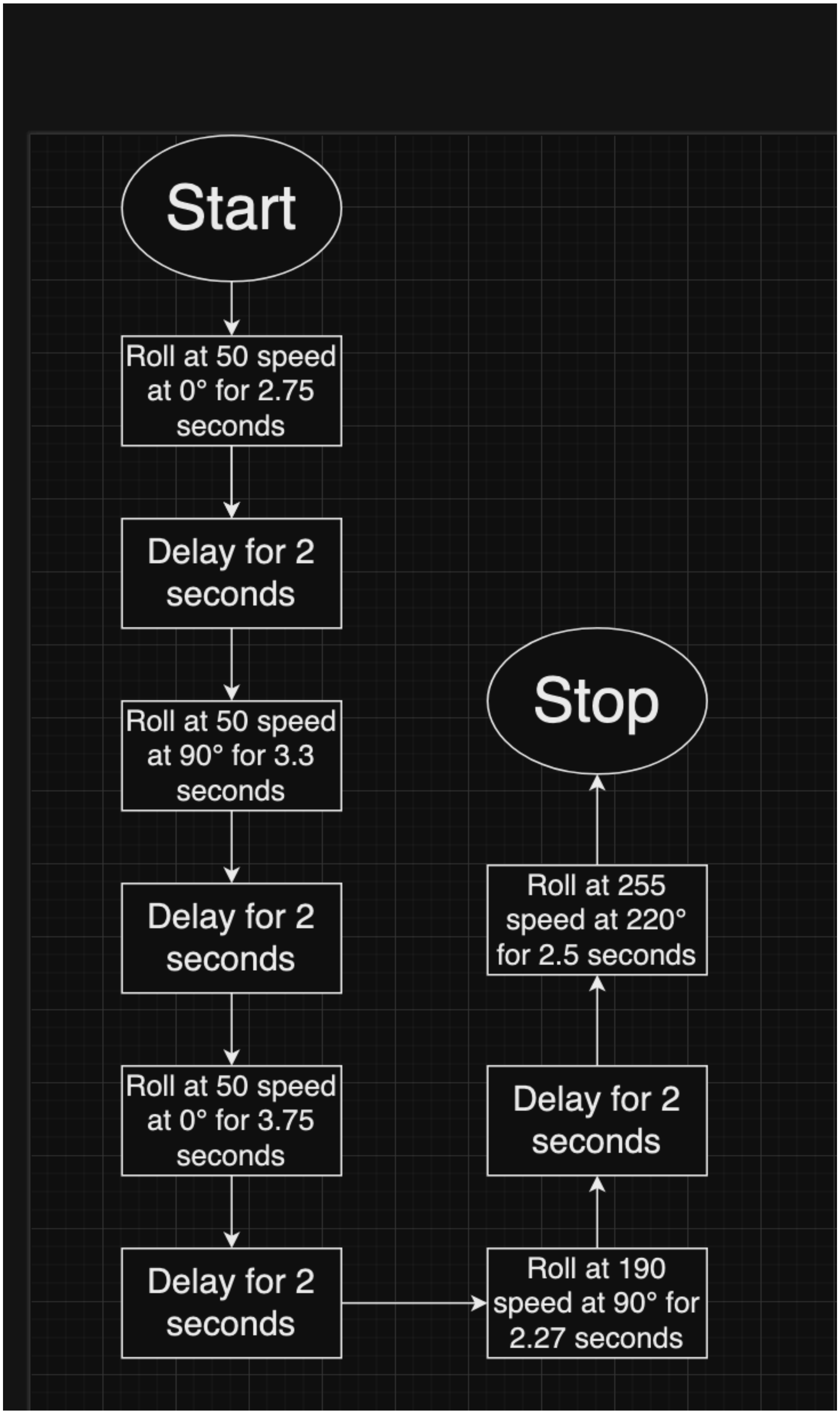
This section provides all of the details concerning the design and process of how the agility sprint program was created. It gives a sequence of the planning, algorithm, software, system flowchart, hardware, test plan, gantt chart, and staffing plan. All of these were developed and would not be possible if they had not been properly merged together.

5.1 Algorithm

This Algorithm is designed in terms of pseudocode on how the overall actual program will turn out. It suggests the speeds and time as , as it is unknown how fast and for how long it will take before actual tests.

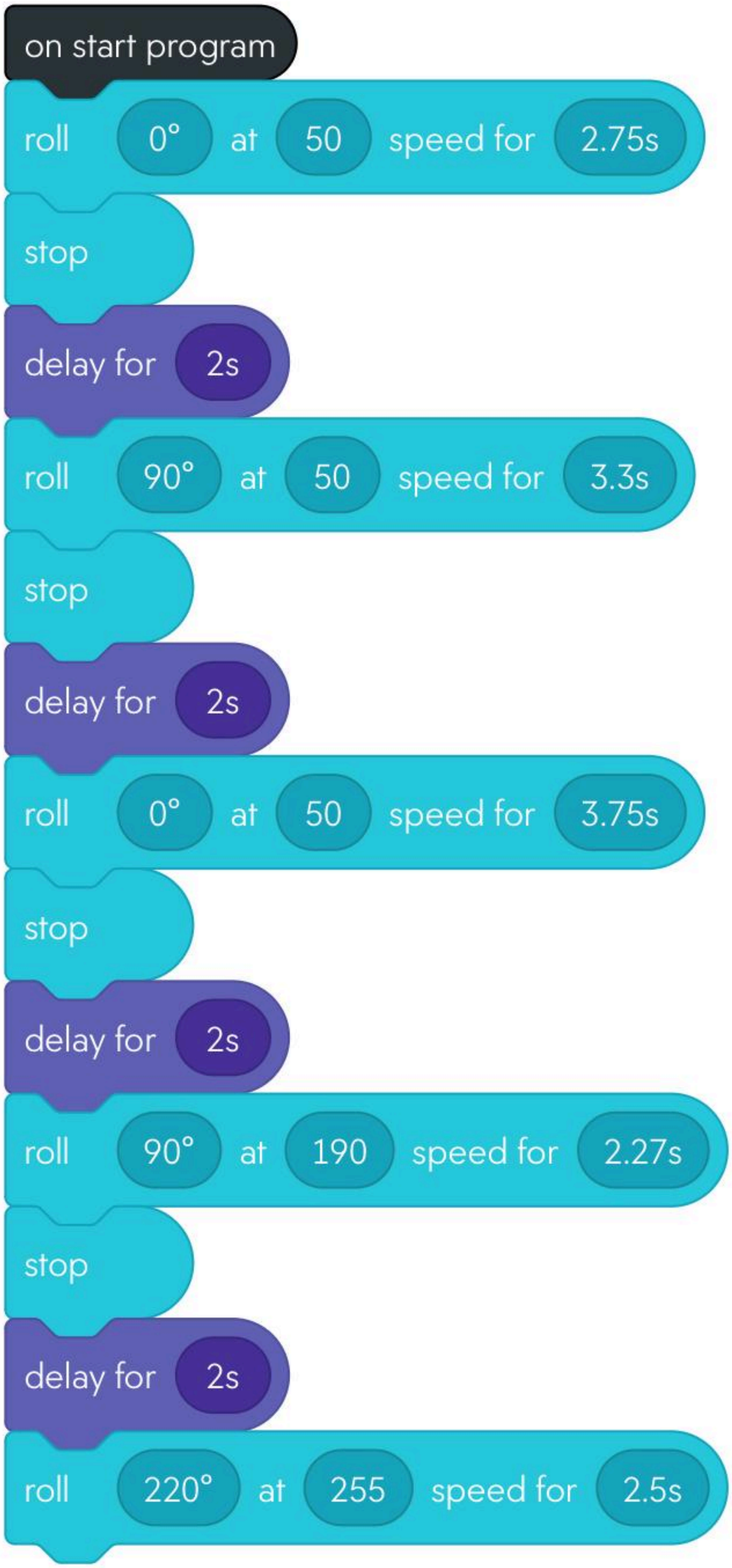
- 1. Robot Starts at the origin
- 2. Roll at speed x at y degrees for z seconds
- 3. Delays for 2 seconds
- 4. Roll at speed x at y degrees for z seconds
- 5. Delays for 2 second
- 6. Roll at x speed at y degrees for z seconds
- 7. Delays for 2 seconds
- 8. Roll at fastest speed at y degrees for z seconds
- 9. Delays for 2 seconds
- 9. Roll at x speed at y degrees for z seconds
- End Program

5.2 System Flow

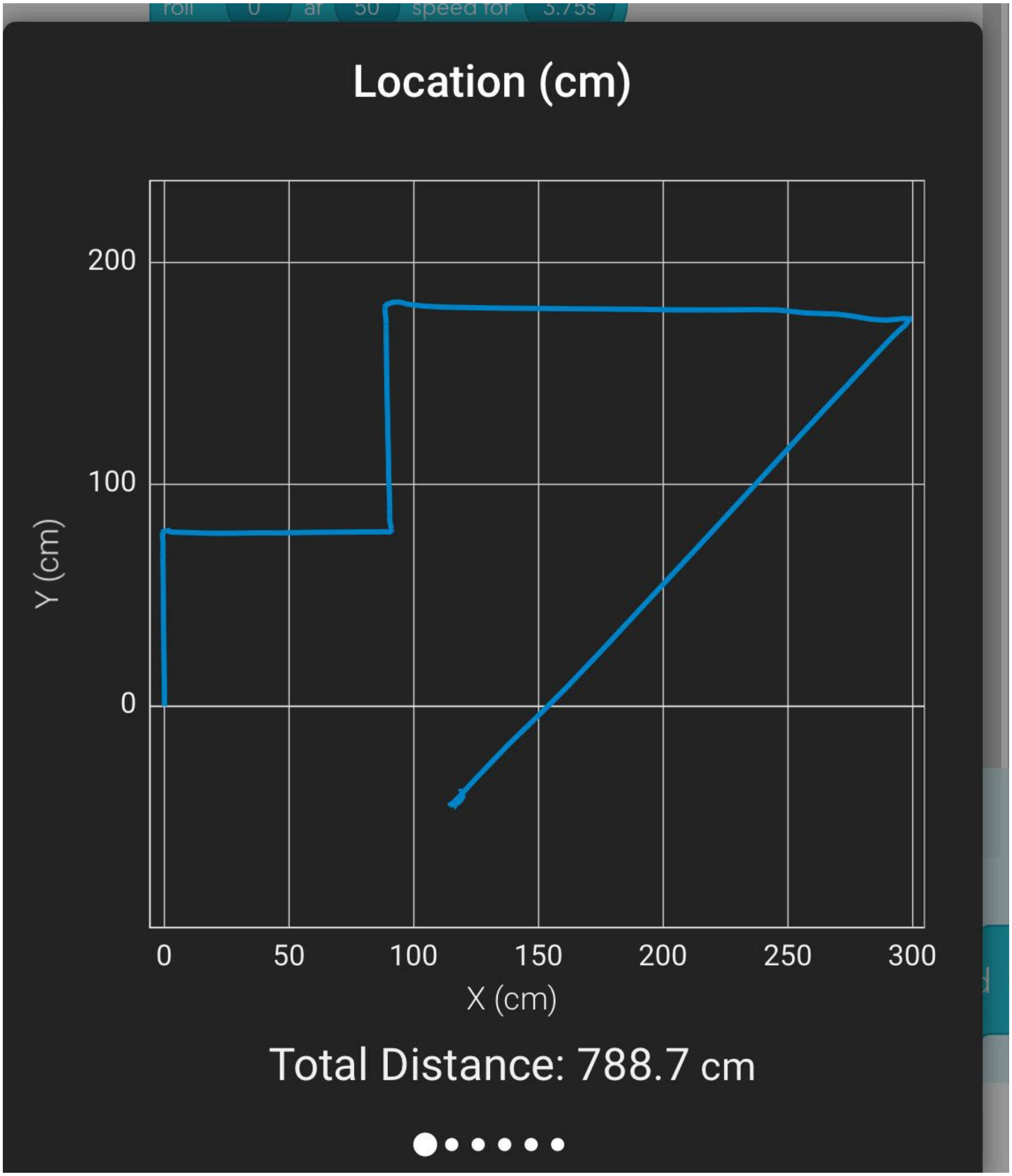

Start
Roll at 50 speed at 0° for 2.75 seconds
Delay for 2 seconds
Roll at 50 speed at 90° for 3.3 seconds
Delay for 2 seconds
Roll at 50 speed at 0° for 3.75 seconds
Delay for 2 seconds
Roll at 190 speed at 90° for 2.27 seconds
Delay for 2 seconds
Roll at 255 speed at 220° for 2.5 seconds
Stop

5.3 Software

- SpheroEDU is the coding language we used to program the output and performance for this sprint.
- We used Github to upload our system design document along with our video of the sprint.



```
on start program
roll 0° at 50 speed for 2.75s
stop
delay for 2s
roll 90° at 50 speed for 3.3s
stop
delay for 2s
roll 0° at 50 speed for 3.75s
stop
delay for 2s
roll 90° at 190 speed for 2.27s
stop
delay for 2s
roll 220° at 255 speed for 2.5s
```



Total Distance: 788.7 cm

5.4 Hardware

- A Macbook and Iphones were used for designing, testing, and running the program. Also speaks when the robot says “I am the winner.”
- SPRK Robot: Used in the accuracy run. Moves around the space to complete the trial. Turns red, orange, yellow, green, blue, purple in the experiment.
- Blue tape: the pathway of where the SPRK Robot should follow upon.

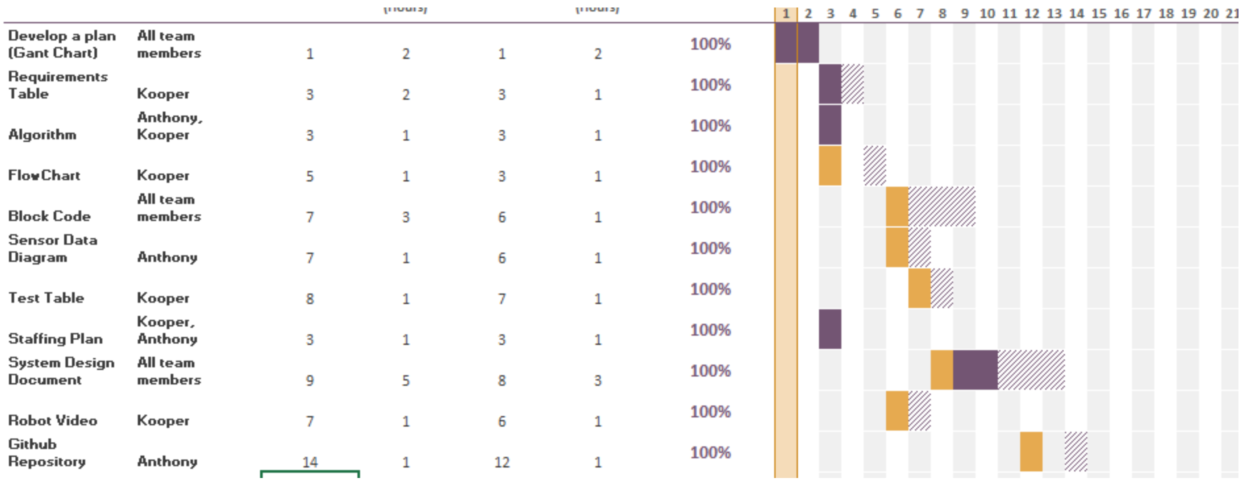
5.5 Test Plan

Reason for Test Case	Test Date	Expected Output	Observed Output	Staff Name	Pass/Fail
Robot moves across first blue line	4/17/24	Robot avoids obstacle #1, stopping at the end of line #1	Robot overshot the line.	Kooper, Esha, Anthony	Fail
Robot moves across first blue line	4/17/24	Robot avoids obstacle #1, stopping at the end of line #1	Robot went the wrong direction	Kooper, Esha, Anthony	Fail
Robot moves across first blue line	4/17/24	Robot avoids obstacle #1, stopping at the end of line #1	Robot produced the expected output	Kooper, Esha, Anthony	Pass
Robot moves across second blue line	4/17/24	Robot avoids obstacle #1 and obstacle #2	Robot hit obstacle #1 in turning	Kooper, Esha, Anthony	Fail

		stopping at the end of line #2			
Robot moves across second blue line	4/17/24	Robot avoids obstacle #1 and obstacle #2 stopping at the end of line #2	Robot produced the expected output	Kooper, Esha, Anthony	Pass
Robot moves across line #3	4/17/24	Robot avoids obstacle #2 and #3, stopping at the end of line #3	Robot moved at the wrong angle	Kooper, Esha, Anthony	Fail
Robot moves across line #3	4/17/24	Robot avoids obstacle #2 and #3, stopping at the end of line #3	Robot moved at the wrong angle, slightly overshot	Kooper, Esha, Anthony	Fail
Robot moves across line #3	4/17	Robot avoids obstacle #2 and #3, stopping at the end of line #3	Robot produced the expected output	Kooper, Esha, Anthony	Pass
Robot goes over the ramp on line #4	4/17/24	Robot jumps the ramp and ends at the end of line #4	Robot did not go fast and long enough, and ended on the ramp.	Kooper, Esha, Anthony	Fail
Robot goes over the ramp on line #4	4/17/24	Robot jumps the ramp and ends at the end of line #4	Robot went fast enough but continued for too	Kooper, Esha, Anthony	Fail

			long after the blue line		
Robot goes over the ramp on line #4	4/17/24	Robot jumps the ramp and ends at the end of line #4	Robot produced the expected output	Kooper, Esha, Anthony	Pass
Robot moves across line #5 and knocks over all pins	4/17/24	Robot knocks overall of the the pins after moving across line #5	Robot missed the pins	Kooper, Esha, Anthony	Fail
Robot moves across line #5 and knocks over all pins	4/17/24	Robot knocks overall of the the pins after moving across line #5	Robot produced the expected output and knocked over all pins	Kooper, Esha, Anthony	Pass

5.6 Task List/Gantt Chart



			(hours)		(hours)	
Develop a plan (Gant Chart)	All team members	1	2	1	2	100%
Requirements Table	Kooper	3	2	3	1	100%
Algorithm	Anthony, Kooper	3	1	3	1	100%
FlowChart	Kooper	5	1	3	1	100%
Block Code	All team members	7	3	6	1	100%
Sensor Data Diagram	Anthony	7	1	6	1	100%
Test Table	Kooper	8	1	7	1	100%
Staffing Plan	Kooper, Anthony	3	1	3	1	100%
System Design Document	All team members	9	5	8	3	100%
Robot Video	Kooper	7	1	6	1	100%
Github Repository	Anthony	14	1	12	1	100%

5.7 Staffing Plan

Name	Role	Responsibility	Reports To
Kooper K	Project Supervisor	Divides roles, works on all aspects of the project. Oversees progress of the project.	Prof. Eckert
Esha A	Project Editor	Edits the final SDD and make sure the project is structured properly.	Prof. Eckert, Kooper K, Anthony S
Anthony S	Management and editor	Edits the final SDD, tests the robot and gets the video for gitHUB repository.	Prof. Eckert