Sprint 2 - Accuracy Design Document

Table of Contents

1. EXECUTIVE SUMMARY

1.1 Project Overview

1.2 Purpose and Scope of this Specification

2. PRODUCT/SERVICE DESCRIPTION

2.1 Product Context

2.2 User Characteristics

2.3 Assumptions

2.4 Constraints

2.5 Dependencies

3. REQUIREMENTS

3.1 Functional Requirements

3.2 Security

3.2.1 Protection

3.2.2 Authorization and Authentication

3.3 Portability

4. REQUIREMENTS CONFIRMATION/STAKEHOLDER SIGN-OFF

5. SYSTEM DESIGN

5.1 Algorithm

5.2 System Flow

5.3 Software

5.4 Hardware

5.5 Test Plan

5.6 Task List/Gantt Chart

5.7 Staffing Plan

# **1. Executive Summary**

## **1.1** **Project Overview**

## The project at hand involves developing an algorithm using the Sphero Spark 2 and its Sphero Edu platform to successfully complete a triathlon course called "Accuracy." The course consists of a figure-eight path that must be traversed five times, with the robot staying within the provided path and starting and finishing in the same location. The intended audience for this project is anyone interested in robotics and programming, particularly those interested in using the Sphero Spark 2 and its associated platform.

## **1.2** **Purpose and Scope of this Specification**

* In scope, this document outlines the specific requirements for the algorithm to navigate the figure eight course with a path provided and complete it successfully five times, while staying within the given path, starting and finishing in the same square, and speaking and flashing multicolored lights for 5 seconds upon completion.
* Out of scope, this document does not cover the physical setup of the course or the hardware setup of the Sphero Spark 2 robot. Additionally, this document does not cover modifications to the algorithm to adapt to different course designs or variations of the "Accuracy" triathlon.

## **2. Product/Service Description**

The product being developed is an algorithm designed to successfully complete the Accuracy triathlon course using the Sphero Spark 2 and its Sphero Edu platform. The algorithm will be utilized by students or individuals interested in programming and robotics to demonstrate their skills in navigating a figure-eight course made of tape with a diameter of 5.2 inches. The product requirements are affected by several factors, including the capabilities of the Sphero Spark 2 and the Sphero Edu platform. The algorithm must be designed to work within the limitations and specifications of these products, such as the ability to stay within a path provided by the tape, the starting and finishing point, and the ability to flash multicolored lights. Additionally, the product requirements are influenced by the nature of the triathlon course itself. The course requires the robot to navigate around two circles with a diameter of 5.2 inches while staying within the provided path. The robot must complete the course five times and finish in the same location it started. The algorithm must be designed to ensure that the robot stays within the course path and completes the required number of rounds within the specified time frame. Overall, the product aims to provide an educational and engaging experience for individuals interested in programming and robotics. The algorithm developed will showcase the user's ability to program and navigate a robot through a challenging course, using the Sphero Spark 2 and Sphero Edu platform to achieve success.

* Assumptions: The triathlon course will be set up in a specific way, with clearly marked boundaries and checkpoints. Competitors will have access to the algorithm and will be able to use it on their own devices. Competitors will have basic knowledge of how to use their devices and the algorithm.
* Constraints: The algorithm must be compatible with the device and operating system being used by the competitors. The algorithm must not violate any rules or regulations set forth by the triathlon organizers. The algorithm must be secure and protect user data. The algorithm must be efficient and not require excessive amounts of system resources.
* Dependencies: The algorithm may depend on GPS data to accurately track a competitor's location. The algorithm may depend on network connectivity to download and update course data. The algorithm may depend on updates to the triathlon course layout and boundaries.

## 

## **2.1** **Product Context**

The Sphero Spark 2 robot is an independent and self-contained product. It does not interface with any other related systems.

## **2.2** **User Characteristics**

* The users of our product will be students and faculty members who will be participating in the triathlon competition. The general characteristics of our users are:
* Students/faculty/staff/other: Students and faculty members
* Experience: Familiarity with Sphero Edu
* Technical expertise: Basic understanding of how to control the robot using the algorithm provided
* Other general characteristics that may influence the product: None

## **2.3** **Assumptions**

The Sphero Spark 2 robot is already constructed and does not require any physical modification.

* Users have access to a computer or mobile device to control the robot using the algorithm provided.
* User has basic experience with Sphero Edu
* User has access to the accurate course based on the measurements of the rectangle, with a functional Sphero Spark 2

## **2.4** **Constraints**

* **Working with Old Systems:** Making sure the robot can talk to and work alongside older technology might limit what kind of tools and parts we can use.
* **Keeping Track of Things:** Recording what the robot does and keeping logs could slow it down or take up a lot of space, so we need to find a balance.
* **Who Can Use It and How Safe It Is:** We need to make sure only the right people can use the robot and that it's safe from hackers or other threats. This might affect how we build and program it.
* **How Important It Is:** If the robot needs to work all the time without any problems, we have to design it in a way that it won't break easily and can keep going even if something goes wrong.
* **Limits on What It Can Handle:** The robot can only do so much at once, like store information or think quickly. We need to make sure it doesn't try to do too much and slow down or stop working.
* **Following the Rules:** We have to follow certain rules and use specific tools when building the robot. This helps make sure it works well and can be fixed or updated easily in the future.

## **2.5** **Dependencies**

* The algorithm provided must be compatible with the Sphero Spark 2 robot and its capabilities
* Device must be compatible with Sphero Spark 2
* The Sphero Spark 2 robot must be fully charged and operational before use

# **3. Requirements**

* **Input Requirement 1:** The robot must receive commands from a mobile device via Bluetooth.
  + Function: Receive commands.
  + Data: Commands sent from the mobile device.
  + Output: None.
  + Location: Sphero Bolt Robot.
  + Verifiable: Ensure that the robot responds correctly to commands sent from the mobile device.
* **Input Requirement 2:** The robot must detect obstacles in its path using onboard sensors.
  + Function: Detect obstacles.
  + Data: Sensor readings.
  + Output: None.
  + Location: Sphero Bolt Robot.
  + Verifiable: Ensure that the robot can accurately detect obstacles and change its path accordingly.
* **Output Requirement 1:** The robot must display feedback on its LED matrix to indicate its status.
  + Function: Display feedback.
  + Data: Status information.
  + Output: LED matrix display.
  + Location: Sphero Bolt Robot.
  + Verifiable: Verify that the LED matrix displays the correct status information as intended.

**Priority 2 Requirements:**

* **Input Requirement 3:** The robot must be programmable using a compatible coding interface.
  + Function: Accept programming commands.
  + Data: Code instructions.
  + Output: None.
  + Location: Sphero Bolt Robot.
  + Verifiable: Ensure that the robot accepts and executes programming commands correctly.
* **Output Requirement 2:** The robot must emit sounds to provide auditory feedback to users.
  + Function: Emit sounds.
  + Data: Auditory feedback signals.
  + Output: Speaker.
  + Location: Sphero Bolt Robot.
  + Verifiable: Verify that the robot emits sounds in response to specific events or commands.

**Priority 3 Requirements:**

* **Input Requirement 4:** The robot should be able to receive firmware updates wirelessly.
  + Function: Receive firmware updates.
  + Data: Firmware update files.
  + Output: None.
  + Location: Sphero Bolt Robot.
  + Verifiable: Test the ability of the robot to successfully receive and apply firmware updates.
* **Output Requirement 3:** The robot may provide telemetry data to a connected mobile device for analysis.
  + Function: Provide telemetry data.
  + Data: Sensor readings, system status.
  + Output: Data stream.
  + Location: Sphero Bolt Robot to Mobile Device.
  + Verifiable: Ensure that the robot can transmit telemetry data accurately to the connected mobile device.
  + The algorithm must be able to navigate the Sphero Spark 2 along the figure-eight path provided for the "Accuracy" triathlon course.
  + The Sphero Spark 2 must remain within the provided path at all times during the traversal of the course.
  + The Sphero Spark 2 must start and finish in the same location
  + The algorithm must be able to detect and respond appropriately to obstacles on the course.
  + The algorithm must be able to adjust the speed and direction of the Sphero Spark 2 to account for changes in the course.
  + The algorithm must be able to navigate the course five times consecutively without interruption.
  + The algorithm must be developed using the Sphero Edu platform.
  + The algorithm must be compatible with the Sphero Spark 2 robot.
  + The algorithm must be able to run on a standard computer or mobile device.
  + The project must provide an engaging educational experience for those interested in robotics and programming.
  + The project must demonstrate the capabilities of the Sphero Spark 2 robot and its associated platform.
  + The project must be accessible to a broad audience with varying levels of robotics and programming experience.

**Priority Definitions:**

* Priority 1: The requirement is essential and must be fulfilled for the project to be considered successful.
* Priority 2: The requirement is important and should be fulfilled to enhance the functionality of the project.
* Priority 3: The requirement is desirable but not essential for the project to be considered successful.

## **3.1** **Functional Requirements**

| Req# | Requirement | Comments | Priority | Date Rvwd | SME Reviewed / Approved |
| --- | --- | --- | --- | --- | --- |
| ENDUR\_01 | The Main LED on the robot must be blue before moving or speaking |  | 1 | 4/8/24 | Joseph |
| ENDUR\_02 | The Robot must Say Ready, Set, Go before moving |  | 1 | 4/8/24 | Joseph |
| ENDUR\_03 | The algorithm must control the robot’s movement within the specified figure 8 course |  | 2 | 4/8/24 | Eddie |
| ENDUR\_04 | The algorithm must ensure the robot maintains a stable roll angle 0 degrees during the initial movement |  | 1 | 4/9/24 | Eddie |
| ENDUR\_05 | The algorithm must be able to make the robot spin at specific points along the course |  | 3 | 4/9/24 | Joseph |
| ENDUR\_06 | The robot must spin at a speed of 122 in a direction of 365 degrees for a time of 7 seconds and repeat this 5 times |  | 1 | 4/10/24 | Joseph |
| ENDUR\_07 | The robot must reset aim to spin in the opposite direction at -365 degrees for 7 seconds and repeat this 5 times |  | 2 | 4/10/24 | Eddie |
| ENDUR\_08 | The robot must then stop and say “I am the Winner” |  | 2 | 4/10/24 | Eddie |
| ENDUR\_09 | Wait and robot must fade the main light from Purple to Green for 3s, then fade from Light Blue to Yellow for 3s, and fade from Orange to Red for 3s |  | 2 | 4/10/24 | Joseph |

## 

## **3.2** **Security**

### **3.2.1** **Protection**

* Encryption: All sensitive data will be encrypted using industry-standard encryption algorithms.
* Activity logging: System activities will be logged to create historical data sets for auditing purposes.
* Restrictions on inter-module communications: Only authorized modules will be able to communicate with each other, and communication between modules will be restricted based on their permissions.
* Data integrity checks: Data integrity checks will be performed to ensure that data has not been tampered with or corrupted

### 

### **3.2.2** **Authorization and Authentication**

* To ensure proper authorization and authentication, standard authorization tools will be utilized.
* This will require users to log in with a username and password, and access to certain system features and data will be restricted based on user roles and permissions.
* Implementation of multi-factor authentication for additional security

## **3.3** **Portability**

* The following attributes of the system relate to the ease of porting the system to other host machines and/or operating systems:
* Percentage of components with host-dependent code: All components will be developed using platform-independent code to ensure ease of portability.
* Percentage of code that is host-dependent: All code will be developed to be platform-independent to ensure ease of portability.
* Use of a proven portable language: The system will be developed using a widely used, portable programming language.
* Use of a particular operating system: The system will be designed to run on multiple operating systems, including Windows, MacOS, and Linux.
* The need for environment independence: The product will be designed to operate the same regardless of operating systems, networks, development or production environments.

# 

# 

# **4. Requirements Confirmation/Stakeholder sign-off**

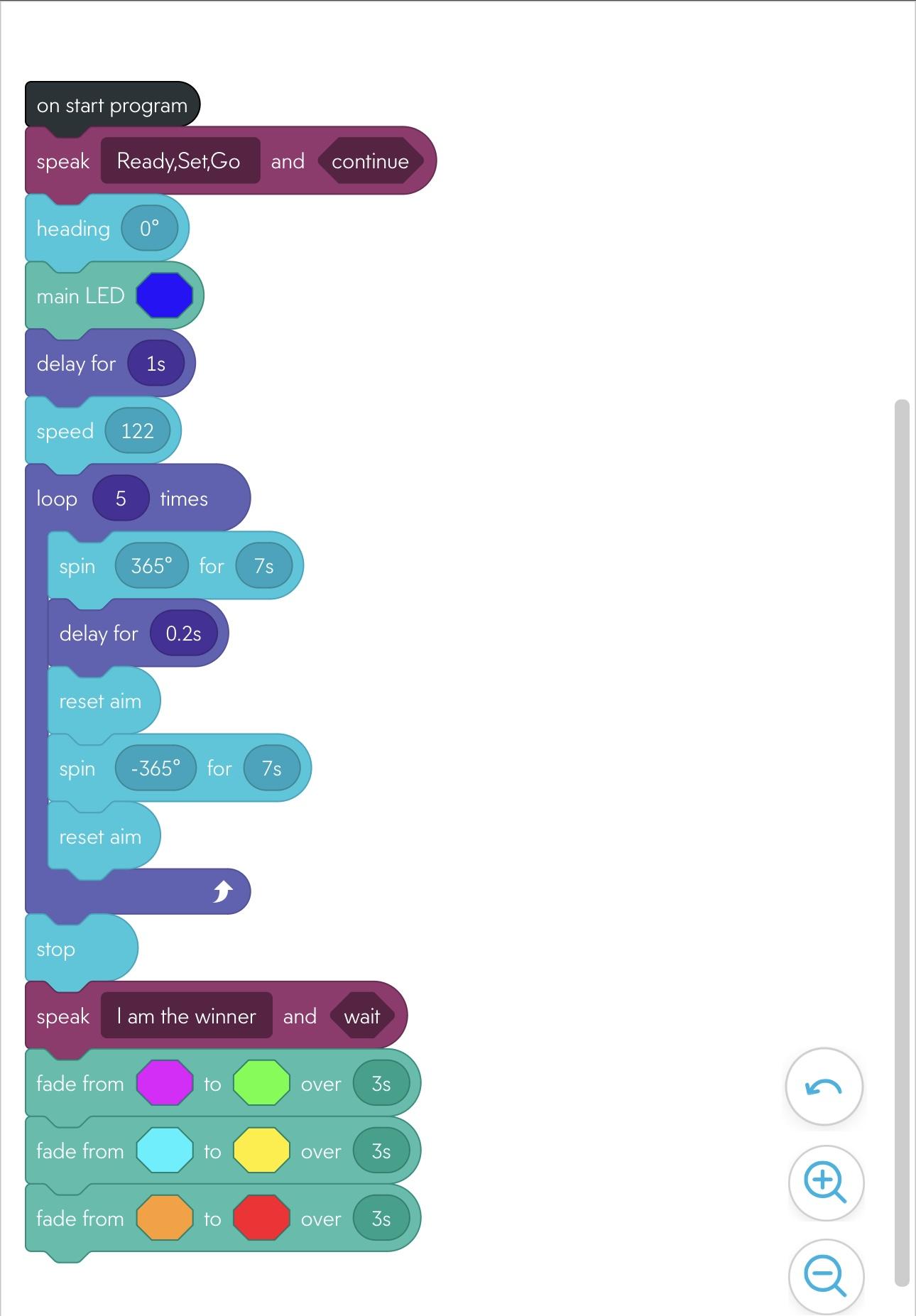
| Meeting Date | Attendees (name and role) | Comments |
| --- | --- | --- |
| 04/8/24 | Eddie, Joseph |  |
| 04/9/24 | Eddie, Joseph |  |
| 04/10/24 | Eddie, Joseph |  |

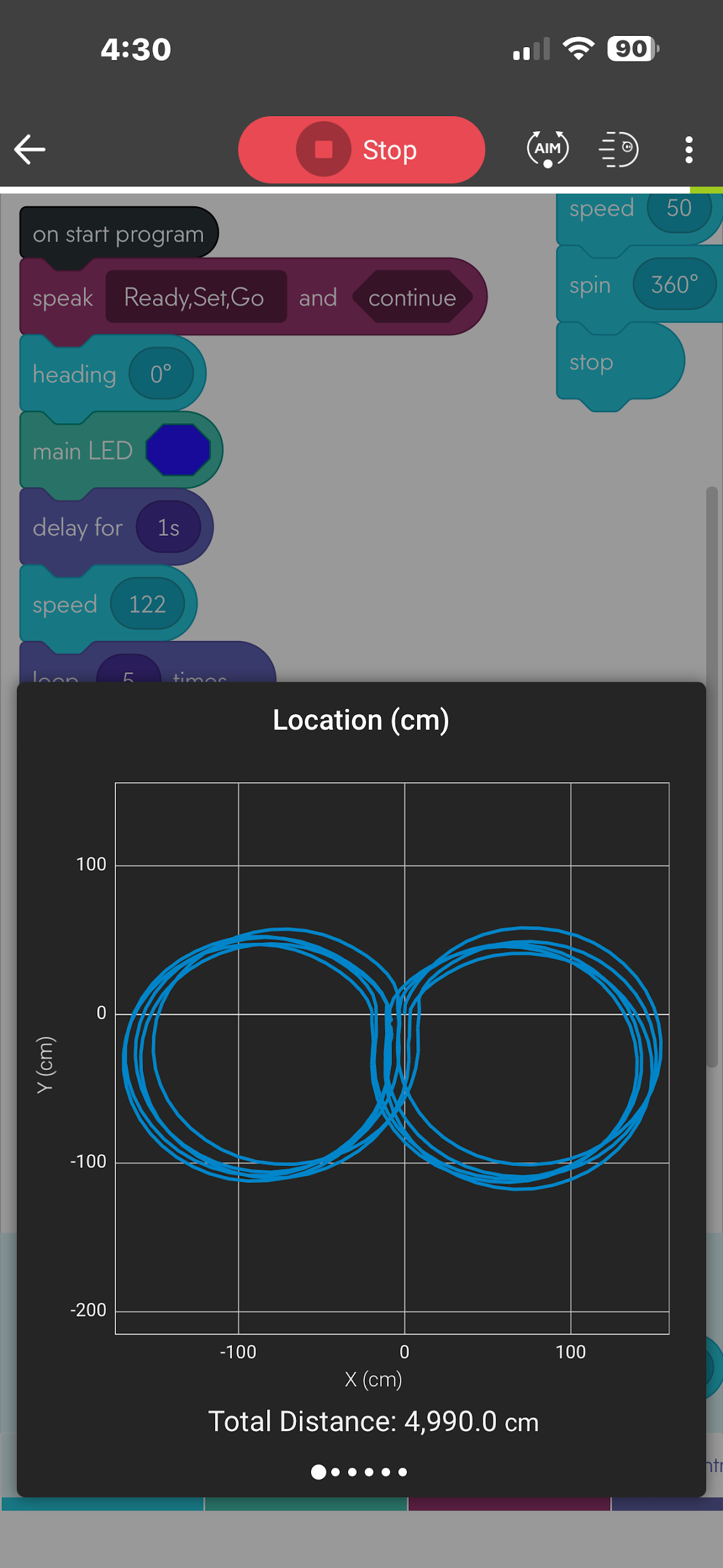
# **4. System Design**

## **4.1** **Algorithm**

**Program:**

* On start program
* Speak “Ready,Set,Go”, and Continue
* Heading 0 degrees
* Main LED is Blue
* Delay for 1s
* Speed set to 122
* (Loop 5 times
* Spin 365 degrees for 7s
* Delay for 0.2s
* Reset aim
* Spin -365 degrees for 7s
* Reset aim)
* Stop
* Speak “ I am the Winner” and Wait
* Fade from Purple to Green for 3s
* Fade from Light Blue to Yellow for 3s
* Fade from Orange to Red for 3s

****



* The program begins by starting up and initializing the Sphero Spark 2 robot.
* Once the robot is ready, the program speaks the phrase "Ready, Set, Go" to indicate the start of the race
* The robot then sets its heading to 0 degrees and turns on its main LED, which is blue.
* After a brief delay of 1 second, the robot sets its speed to 122 and enters a loop that will execute 5 times.
* Within each iteration of the loop, the robot spins 360 degrees clockwise for 7 seconds, pauses for another 0.2 seconds, and then spins 360 degrees counterclockwise for another 7 seconds. After the fifth iteration of the loop, the robot stops moving and speaks the phrase "I am the winner".
* Also there are two “reset aims” one after the first delay of 0.2s and one after the spin for -365 degrees for 7s.
* Finally, the robot fades its main LED from green to pink over a period of 3 seconds, then from blue to red over another 3 seconds, and finally from purple to light blue for another 3 seconds.
* Overall, this algorithm outlines the steps required for the Sphero Spark 2 robot to complete the figure-eight path of the "Accuracy" triathlon course, while also showcasing some additional functionality (such as speaking and LED fading) that may enhance the overall user experience.

## 

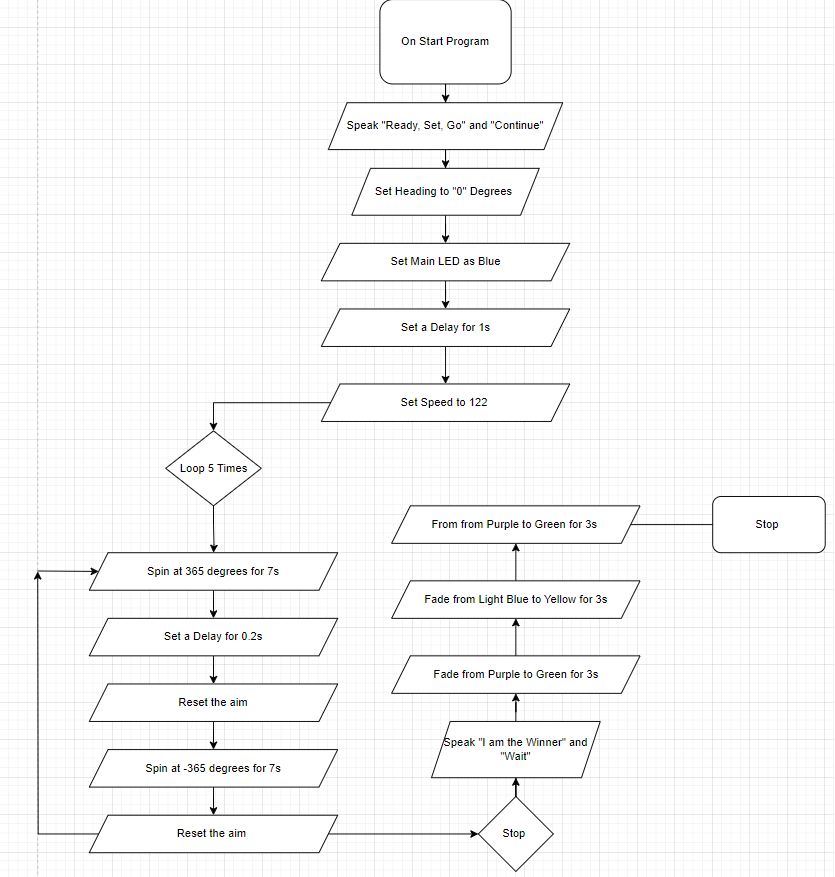
## 

## 

## 

## 

## **4.2** **System Flow(Flow Chart)**



## 

## **4.3** **Software**

We used Github to display our code and Sphero Edu to write the code in order to tell the robot what we needed it to do. Also use Block Code integrated with Sphero Edu powered by Javascript

## **4.4** **Hardware**

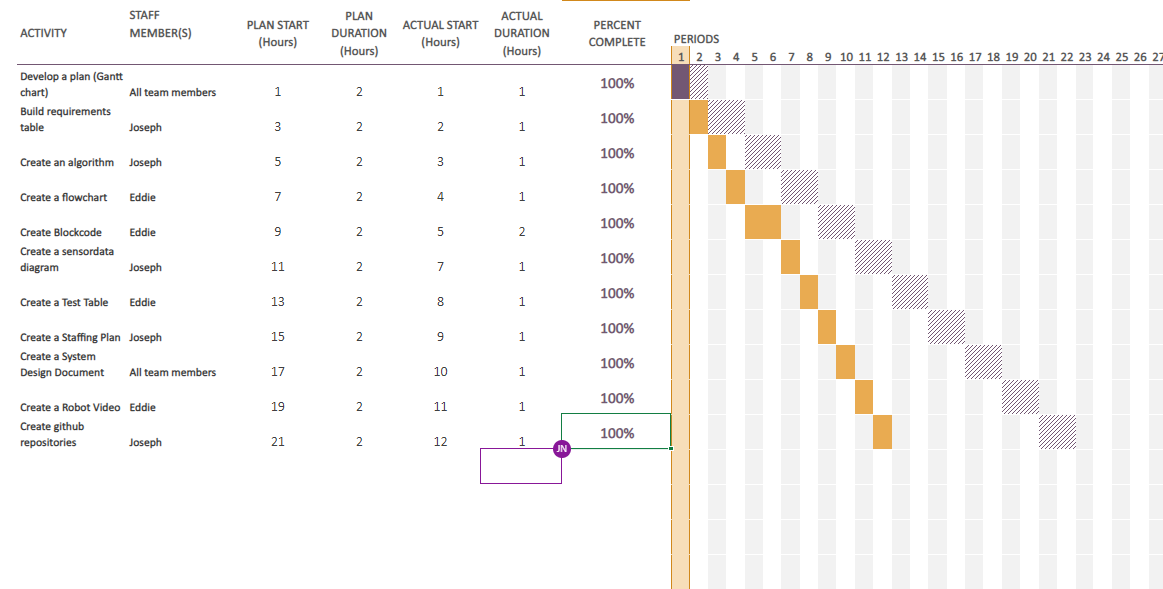
We used our laptops in order to develop, test and demonstrate a well written code in the application called Sphero Edu. It is an app that allows you to use block code to make the robot move.

## **4.5** **Test Plan**

| **Reason for Test Case** | **Test Date** | **Expected Output** | **Observed Output** | **Staff Name** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| Sphero Bolt Robot to spin in a figure eight | 4/8/24 | Set degree to 365 so it spins in a circle and then -365 so it makes the figure eight | Robot went off course | Eddie | Fail |
| Sphero Bolt Robot to spin in a figure eight | 4/9/24 | Set degree to 365 so it spins in a circle and then -365 so it makes the figure eight | Robot went in a figure eight | Eddie | Pass |
| What Speed the Robot should go and how long | 4/9/24 | Set Speed to 122 for 7 sec and it will go in a figure eight and then stop where it started | Robot stopped where it began | Eddie | Pass |
| See what a delay does to the robot | 4/10/24 | Set a Delay for 0.2 sec and see it it affects the speed or anything at all | It made the robot run smoother through the figure eight | Eddie | Pass |
| To see whether the robot can do a figure eight 5 times | 4/10/24 | Set degree to 365 so it spins in a circle and then -365 so it makes the figure eight and Set Speed to 122 for 7 sec and repeat 5 times with a 0.2 sec delay in between | The robot went in a figure eight 5 times | Eddie | Pass |
| Make the Robot stop at its starting position and speak | 4/10/24 | Tell the Robot to stop with a stop block | Robot stopped at starting position | Eddie | Pass |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## 

## **4.6** **Task List/Gantt Chart**



## **5.7** **Staffing Plan**

| Name | Role | Responsibility | Reports To |
| --- | --- | --- | --- |
| Joseph | Ceo | **Build requirements table, create an algorithm, create a staffing plan, create github repositories, create a sensor data diagram, create gantt chart, create system design document** | Gil Eckert |
| Eddie` | Manager | **Create a flowchart, create a test table, create gantt chart, create system design document, create a robot video, create blockcode** | Gil Eckert |
|  |  |  |  |