Sprint 1 - Endurance 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**

This project involves the development of an algorithm to control the movement of a Sphero Spark 2 robot to complete an endurance sprint as part of a triathlon. The Sphero Spark 2 robot will be used to navigate a rectangular course made with tape, which is 22 inches in length and 11.8 inches in width. The robot must follow the course and make turns at specific points to complete the endurance sprint.

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

The purpose of this specification is to outline the requirements and scope of the development of the algorithm for the Sphero Spark 2 robot for the classroom presentation. The intended audience for this specification includes the instructor and students. The scope of this specification includes the following:

**In scope**

Developing an algorithm to control the Sphero Spark 2 robot to complete an endurance sprint by navigating the rectangular course made with tape.

* The Sphero Spark 2 robot must follow the course and make turns at specific points.
* The algorithm must take into account factors such as degrees, speed, and time.
* The blocks to be used include roll, stop, and heading.

**Out of Scope**

* Physical construction or modification of the Sphero Spark 2 robot.
* Integration of the algorithm with other systems or devices.
* Testing or validation of the physical movement of the Sphero Spark 2 robot.

# **2. Product/Service Description**

Our product is a Sphero Spark 2 robot that will be used in an endurance sprint as part of a triathlon competition. The Sphero Spark 2 robot will be controlled by an algorithm to complete a full circumnavigation of a rectangular course made with tape. The course is 22 inches in length and 11.8 inches in width. The Sphero Spark 2 robot must follow the course and make turns at specific points to complete the full circumnavigation.

## **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.

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

| Req# | Requirement | Comments | Priority | Date Rvwd | SME Reviewed / Approved |
| --- | --- | --- | --- | --- | --- |
| ENDUR\_01 | The Main LED on the robot must be green before moving or speaking |  | 1 | 3/26/24 | Joseph |
| ENDUR\_02 | The Robot must Say Ready, Set, Go before moving |  | 1 | 3/26/24 | Joseph |
| ENDUR\_03 | The algorithm must control the robot’s movement within the specified rectangular course |  | 2 | 3/26/24 | Eddie |
| ENDUR\_04 | The algorithm must ensure the robot maintains a stable roll angle 0 degrees during the initial movement |  | 1 | 3/26/24 | Joseph |
| ENDUR\_05 | The algorithm must be able to make the robot turn at specific points along the course |  | 3 | 3/26/24 | Eddie |
| ENDUR\_06 | The robot must roll at a speed of 100 for a time of 9.6 seconds for the first straightaway |  | 1 | 3/26/24 | Eddie |
| ENDUR\_07 | The Robot must turn 90 degrees to the right at the end of the initial movement |  | 2 | 3/27/24 | Joseph |
| ENDUR\_08 | The robot must roll at a speed of 120 for 4.5 seconds during the first turn |  | 2 | 3/27/24 | Eddie |
| ENDUR\_09 | The Robot must turn 90 degrees to the right after completing the first turn |  | 2 | 3/27/24 | Eddie |
| ENDUR\_10 | The robot must roll at a speed of 100 for 9.8 seconds during the second straightaway |  | 1 | 3/27/24 | Joseph |
| ENDUR\_11 | The Robot must turn another 90 degrees to the right after completing the second turn |  | 2 | 3/27/24 | Eddie |
| ENDUR\_12 | The Robot must roll at a speed of 120 for 4.5 seconds during the second turn |  | 1 | 3/27/24 | Eddie |
| ENDUR\_13 | The Robot must stop and Say “I’m done and I need water” |  | 2 | 3/27/24 | Joseph |
| ENDUR\_14 | The Robots Main LED must be RED when it finishes its programs |  | 1 | 3/27/24 | Eddie |
| ENDUR\_15 | The Algorithm must enable the robot to complete the full circumnavigation of the rectangular course within the specified time limit |  | 2 | 3/27/24 | Eddie |

## 

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

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

To protect the system from malicious or accidental access, modification, disclosure, destruction, or misuse, the following factors will be implemented: -

* Encryption of sensitive data
* Activity logging and historical data sets to monitor system usage and detect any unauthorized access or modification
* Restrictions on intermodule communications to prevent unauthorized data transfer between modules
* Data integrity checks to ensure data has not been tampered with

### **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.

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

The percentage of components with host-dependent code is 77%. The percentage of code that is host dependent is 23%. The proven portable language used is Sphero Edu. We used a compiler language called Python. You need environment-independence to make sure the code runs smoothly.

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

| Meeting Date | Attendees (name and role) | Comments |
| --- | --- | --- |
| 03/19/24 | Eddie, Joseph | Started testing the sphero robot and how to get it to go in a square |
| 03/26/24 | Eddie, Joseph | Got the sphero robot to successfully go in a rectangle around the room. Also, worked on the flow chart and filled out the test table |
| 03/27/24 | Eddie, Joseph | Got together to film the robot video of it going around the room, and worked together on the System Design Document. |

# **5. System Design**

## **5.1** **Algorithm**

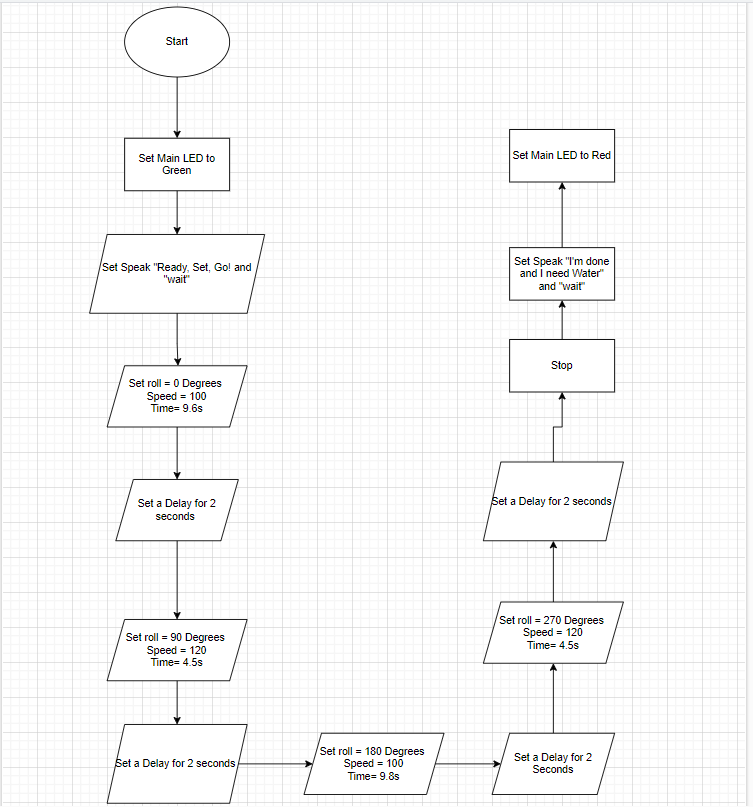
* Set Main Led to Green
* Set Speak to “Ready, Set, Go!”
* Set roll to 0
* Set speed to 100
* Set time to 9.6
* Set Delay to 2
* Change roll to 90
* Change speed to 120
* Change time to 4.5
* Delay stays the same
* Change roll to 180
* Change speed to 100
* Change time to 9.8
* Delay stays the same
* Change roll to 270
* Change speed to 120
* Change time to 4.5
* Delay stays the same
* When roll > 270 stop
* Change Speak to “I’m done and I need Water”
* Change Main Led to Red

## 

## 

## 

## **5.2** **System Flow**



## 

## **5.3** **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.

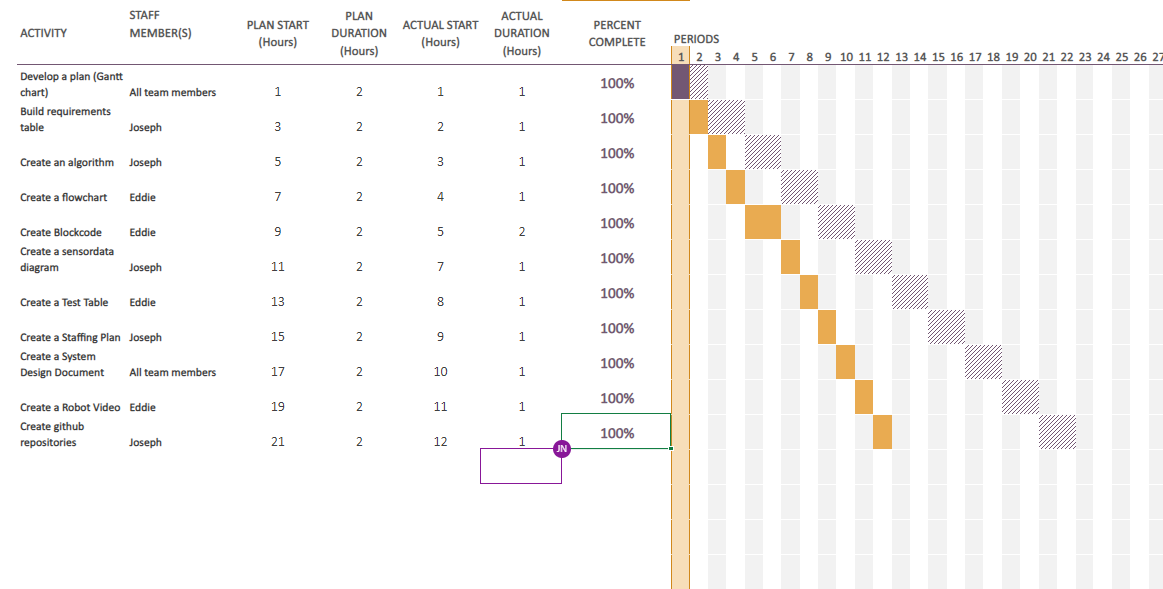
## **5.4** **Hardware**

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.

## **5.5** **Test Plan**

| **Reason for Test Case** | **Test Date** | **Expected Output** | **Observed Output** | **Staff Name** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| Sphero Bolt Robot to go straight | 3/19/24 | Set Degree to 0 so it goes in a straight line | Robot went straight | Eddie | Pass |
| What Speed the Robot should go and how long | 3/19/24 | Set Speed to 100 for 10 sec and it will stop at the tape | Robot went past the taped line | Eddie | Fail |
| What Speed the Robot should go and how long | 3/19/24 | Set Speed to 100 for 9.6 sec and it will stop at the tape | Robot stopped right on the tape | Eddie | Pass |
| Set a Delay for 2 sec and see it it affects the speed or anything at all | 3/26/24 | Set Delay for 2 sec and see if it affects the robot at all | Did not change much just changed how long it took for the next block to run | Eddie | Pass |
| Make the Robot turn and go into a straight line | 3/26/24 | Set Degree to 90 and set speed to 120 for 4.5 sec. Expect the robot to get close to the tape but not exactly on it | Robot went in a straight line and went past the line but now we can adjust the time to get on the tape | Eddie | Pass |
| Make the Robot turn the first corner | 3/26/24 | Set Degree to 180 and set speed to 100 for 9.8 sec. | Robot made its second turn | Joseph | Pass |
| Make the Robot go straight after its second turn | 3/26/24 | Set Degree to 270 and set speed to 120 for 4.5 sec | Robot went in a straight line to complete the rectangle | Joseph | Pass |
| Make the Robot stop at its starting position and speak | 3/26.24 | Tell the Robot to stop with a stop block | Robot stopped at starting position | Joseph | Pass |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## **5.6** **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 |
|  |  |  |  |