Sprint 3 - Agility 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 "Agility." The course consists of a obstacle course that must be traversed without knocking any of the obstacles over, with the robot staying within the provided path and starting and finishing knocking over the pins. 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 obstacle course with a path provided and complete it successfully while also knocking over the “pins”, while staying within the given path, starting and finishing at the end of the course, 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 "Agility" triathlon.

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

The product being developed is an algorithm designed to successfully complete the Agility 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 obstacle 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 an obstacle course with a diameter of 5.2 inches while staying within the provided path. The robot must complete the course without knocking anything over and finish knocking over all the pins at the end. 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.

**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 Robot must Say Ready, Set, Go before moving |  | 1 | 4/16 | Eddie |
| ENDUR\_02 | The algorithm must control the robot’s movement within the specified obstacle course |  | 2 | 4/16 | Joseph |
| ENDUR\_03 | The algorithm must ensure the robot maintains a stable roll angle 0 degrees during the initial movement for 4 seconds at speed 33 |  | 1 | 4/16 | Eddie |
| ENDUR\_04 | The algorithm must be able to make the robot roll through specific points along the course without hitting any obstacles |  | 2 | 4/16 | Joseph |
| ENDUR\_05 | The robot must roll at a speed of 33 in a direction of 90 degrees for a time of 4.9 seconds |  | 2 | 4/17 | Eddie |
| ENDUR\_06 | The algorithm must ensure the robot maintains a stable roll angle 0 degrees for 5.5 seconds at speed 33 |  | 1 | 4/17 | Joseph |
| ENDUR\_07 | The robot must roll at a speed of 200 in a direction of 90 degrees for a time of 2.5 seconds |  | 1 | 4/17 | Eddie |
| ENDUR\_08 | The robot must reset aim to roll 135 degrees at a speed of 135 for 5 seconds |  | 1 | 4/18 | Joseph |
| ENDUR\_09 | The robot must then stop and say “I am the Winner” |  | 1 | 4/18 | Eddie |
| ENDUR\_10 | Continue 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 then change main led to blue |  | 1 | 4/18 | 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 |
| --- | --- | --- |
| 04/15/24 | Eddie, Joseph |  |
| 04/16/24 | Eddie, Joseph |  |
| 04/17/24 | Eddie, Joseph |  |

# **4. System Design**

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

* On start program
* Speak “Ready,Set,Go”, and Continue
* Roll for 0 degrees at a speed of 33 for 4s
* Delay for 1s
* Roll for 90 degrees at a speed of 33 for 4.9s
* Delay for 1s
* Roll for 0 degrees at a speed of 33 for 5.5s
* Delay for 1s
* Roll for 90 degrees at a speed of 200 for 2.5s
* Delay for 2s
* Reset aim
* Roll for 135 degrees at a speed of 135 for 5s
* Stop
* Speak “ I am the Winner” and Continue
* Fade from Purple to Green for 3s
* Fade from Light Blue to Yellow for 3s
* Fade from Orange to Red for 3s
* Main Led is Blue

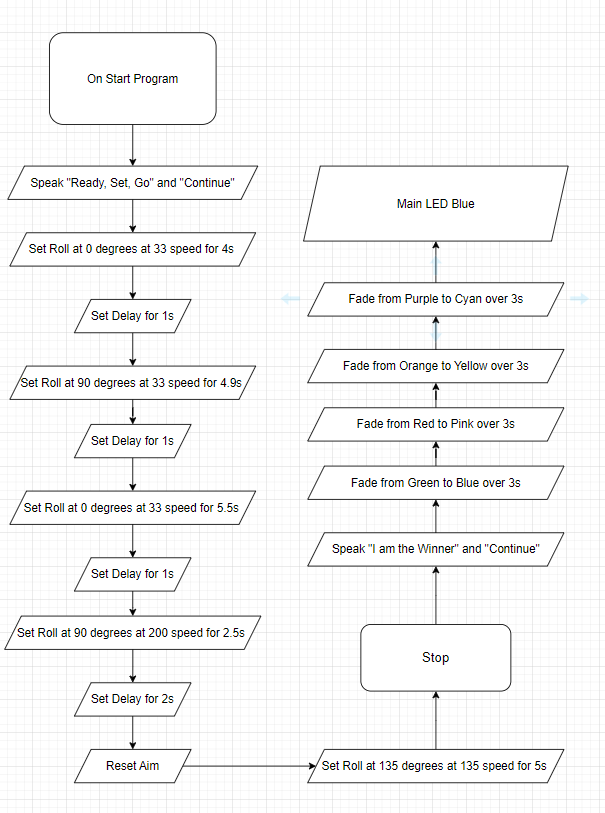
## 

## 

## 

## 

## **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**

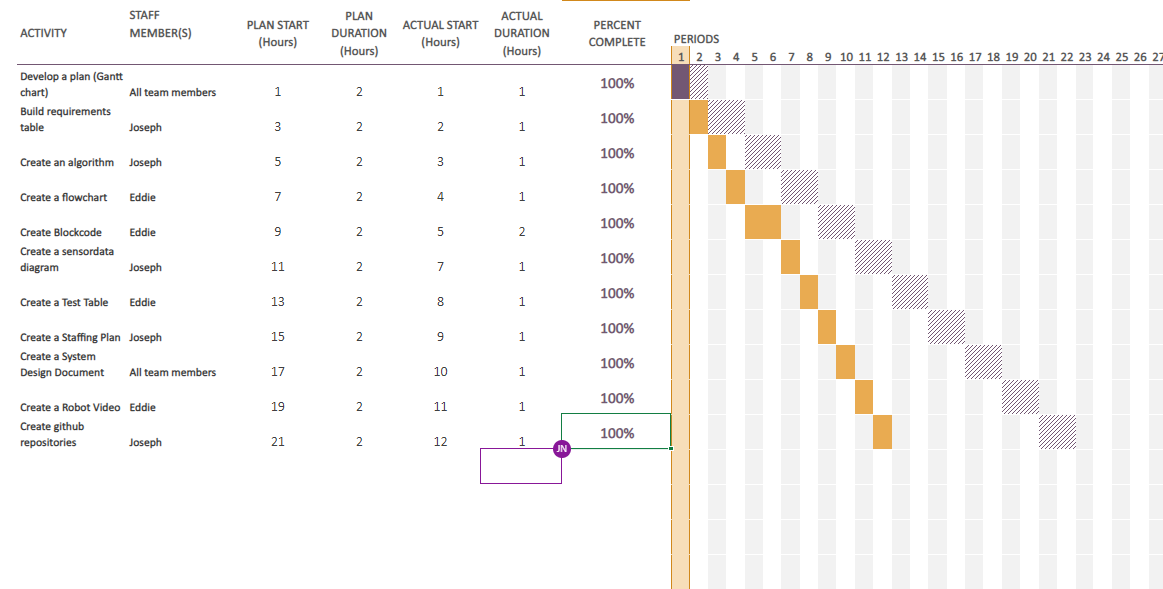
We used our laptops in order to develop, test and demonstrate a well written code in the application called Sphero Edu. It is an app that allows you to use block code to make the robot move.

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

| **Reason for Test Case** | **Test Date** | **Expected Output** | **Observed Output** | **Staff Name** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| Sphero Bolt Robot to go in a zigzag while avoiding obstacles | 4/16 | - Roll for 0 degrees at a speed of 33 for 4s  - Delay for 1s  - Roll for 90 degrees at a speed of 33 for 4.9s  - Delay for 1s  - Roll for 0 degrees at a speed of 33 for 5.5s  - Delay for 1s | Robot went in a zigzag | Eddie | Pass |
| Sphero Bolt Robot to go over a ramp | 4/16 | Goes fast and jumps the ramp | Robot missed the ramp | Joseph | Fail |
| Sphero Bolt Robot to go over a ramp | 4/16 | Goes fast and jumps the ramp | Robot missed the ramp | Eddie | Fail |
| Sphero Bolt Robot to go over a ramp | 4/16 | - Roll for 90 degrees at a speed of 200 for 2.5s | Robot successfully jumped the ramp | Joseph | Pass |
| Sphero Bolt Robot knocks over all ten markers | 4/16 | Robot knocks over the markers | Robot went off course missing all markers | Eddie | Fail |
| Sphero Bolt Robot knocks over all ten markers | 4/16 | Robot knocks over the markers | Robot went off course missing all markers | Joseph | Pass |
| Sphero Bolt Robot knocks over all ten markers | 4/17 | Robot knocks over the markers | Robot went off course missing all markers | Eddie | Fail |
| Sphero Bolt Robot knocks over all ten markers | 4/18 | - Reset aim  - Roll for 135 degrees at a speed of 135 for 5s | Robot knocked down all ten markers | Joseph | 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 |
|  |  |  |  |