Sprint 3 - Agility Design Document

April 18th, 2024

Table of Contents

1. Executive Summary

1.1 Project Overview

This project is to showcase the new and improved functionality of the Sphero robots. This product is intended for anyone who is learning to code either as a profession or a hobby. In the following sections you will find the general makeup of the testing on the Sphero robots that have been conducted onsite, as well as external research into the making of these robots. There will also be a proposal of unique ways to use this software outside of learning moving forward in the future.

1.2 Purpose and Scope of this Specification

The purpose of this specification is to provide clear guidelines and requirements for the design and development of our team's Sphero robot project. It outlines the functionality, constraints, and considerations that need to be addressed during this project's lifecycle.

In scope

Functional Requirements: Ensure the robot completes the obstacle course.

Design Constraints: All programming is done through the Sphero Robot application.

Hardware/Software: This document explains the difference in using different software and the most compatible as well as the required hardware.

Testing: Document provides an in-depth test table with each test our team has run.

Out of Scope

Coding Application: This document does not go into depth with how the Sphero Application coding was developed.

Robot Design: This document does not go into the Sphero robots' design or how they are made.

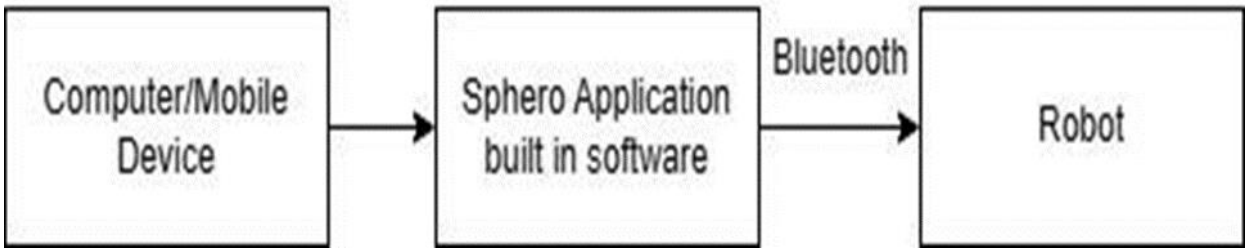
Sales Pitch: It is not our intention to sell the reader on anything, this document is simply research conducted by our team utilizing the Sphero Robots and their software.

2. Product/Service Description

To ensure proper functionality of the robots, the proper operating system is paramount. Due to the developments made by the Sphero team, the Robots work best with Mac OS but do offer operating capabilities with Windows or Linux OS. The Sphero program utilizes block coding for easy-to-use development which reduces development time exponentially.

2.1 Product Context

The Sphero robots utilize state of the art technology for easy-to-use coding with preset "blocks" that can be used to set the robots function as the user sees fit. The robots utilize similar but simplified software that the Roomba Vacuum or RC cars utilize. The Sphero robots are self-contained in that each component is designed to operate that single individual robot. Sphero does work with a variety of related systems including Computer/Mobile Devices via Bluetooth. These computer/mobile devices link directly to the robot and can control the robot as coding is done via the Sphero application and not onto the robot itself.



2.2 User Characteristics

Student:

Role: Student

Experience: Varies based upon users' knowledge.

Technical Expertise: Depends on if the student has experience/is familiar with operating a smartphone or laptop. Typically ranges from low to moderate.

Other Characteristics: Creativity and willingness to learn. Students may use the Sphero Robots to understand creating an algorithm and putting their on-paper tests into physical ones.

Faculty/Staff:

Role: Educator

Experience: Varies, but generally more experienced compared to students. Faculty may have some background in education, technology, or STEM fields.

Technical Expertise: Ranges from moderate to high. Faculty and staff may have experience with educational technology, programming, and curriculum development.

Other Characteristics: Interest in incorporating innovative teaching methods and technologies into their courses. Allows freedom for students to learn proper project management with little interference.

Other:

Role: Individuals either in STEM related fields or enjoy computing as a hobby.

Experience: Varies ranging from complete beginners to experienced individuals with a passion for technology.

Technical Expertise: Varies based upon individuals background, interest, and experience with programming or robotics.

Other Characteristics: Interest in technology and learning. Users may use the Sphero robots for recreational purposes or to learn more about programming and robotics.

2.3 Assumptions

Internet Connection- The user must ensure they are connected to the internet via Wi-Fi, hotspot, or cellular data to ensure proper Bluetooth connection to the robot.

Operating System- The Sphero Robots work best with Mac OS but are compatible with Windows, Linux, Android, and Chromebook as well with reduced functions.

Physical Environment- The robot is set to traverse a certain obstacle course and if the course were to change then the code would have to be adjusted to fit.

Age and Experience of User- The simple and easy to use application provides users with an efficient way to design the robots' functions, erasing many abstractions that can be present in coding.

2.4 Constraints

Old System Compatibility: While the Sphero application works best on Mac OS, it can be run on older systems with limited functions. For the best operation you must ensure the most up to date OS and application are downloaded.

Audit Functions: The original code is posted to Github for public viewing, but the SDD is embedded so that every change to the document found can be sourced back to who made the change.

Access and Security: Due to the block code and research being posted to Github, all this information is open source for anyone to use.

Importance of the Project: While not groundbreaking, our research is to show the simplicity of the Sphero application. As our team is new to the world of coding, this project shows how easy the Sphero robots can be to grasp the concept of coding.

Resource Limits: The Sphero Robots utilize a basic software that can be run on a variety of devices if it has the ability to download external applications and connect to wife/Bluetooth.

Design Standards: The Sphero Robots must be programmed within the Sphero application. The Sphero application utilizes block coding and presets that apply directly to Sphero robots. The built program can be used on any Sphero Robot assuming proper Wi-Fi connection to connect the robot via Bluetooth.

2.5 Dependencies

The Sphero Robots act based on the code provided to it via the Sphero application. To ensure proper functionality, the user must ensure the Sphero application is up to date. The proper program can either be downloaded via Github or the web or created from scratch based on the user's preference and tasking.

3. Requirements

To begin this project, all requirements must be described in adequate detail in order to satisfy further steps of the project. The following are the requirements in detail which satisfy the project guidelines:

Robot must start on starting square
Set robots heading in line with agility course
Robot must move along the zig-zag path, avoiding the 3 obstacles
Robot must gain speed to make the jump
Robot must pivot, spinning into the next part of the obstacle course
Robot must gain speed, knocking down as many pins as possible

Following these requirements, a designer may design a system to test and verify that they meet the project guidelines, as per what our group completed.

Requirement #	Priority	Input	Function	Output
1	**1**	Physically placing the robot on the starting square	Physically place robot on starting square	Robot begins on starting square
2	**1**	Robot must be able to move in line with course	Set robots heading inline with obstacle course start	Robot's code will operate starting on desired path
3	**1**	Code robot to move along zig-zag path	Robot must roll at 0° at x speed for x seconds	Robot moves on start of path

4	**1**	Code robot to spin, starting next leg of the zig-zag path.	Robot must spin 0° for x seconds	Robot spins, avoiding obstacle 1
5	**1**	Code robot to move along zig-zag path	Robot must roll at 90° at x speed for x seconds	Robot moves on second part of path
6	**1**	Code robot to spin, starting next leg of the zig-zag path.	Robot must spin 0° for x seconds	Robot spins, avoiding obstacle 2
7	**1**	Code robot to move along zig-zag path	Robot must roll at 0° at x speed for x seconds	Robot moves on third part of path
8	**1**	Code robot to spin, setting up to hit the jump	Robot must spin 0° for x seconds	Robot spins, avoiding obstacle 3
9	**1**	Code robot to gain speed at hit the jump	Robot must roll at 90° at x speed for x seconds	Robot increases speed to make the jump
10	**1**	Code robot to spin, setting inline with final stretch	Robot must spin at 0° for x seconds	Robot spins to get inline with the final stretch knocking down as many pins as possible.
11	**1**	Code robot to roll knocking down as many pins as possible	Robot must roll at 222° at x speed for x seconds,	Robot rolls down the path knocking down the pins at the end of the course.

3.1 Functional Requirements

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
AGILITY_01	Robot must start on provided square	Start	1	15APR24	Chris K
AGILITY_02	Robot's heading must start in line with course	Set heading so 0° is the start of the course	1	15APR24	Chris K
AGILITY_03	Robot must move in zig-zag pattern, avoiding the 3 obstacles	Set robots heading and speed in line with course, spinning the robot to execute turns.	1	15APR24	Chris K
AGILITY_04	Robot must make the jump	Increase robots speed to make it over the jump	1	15APR24	Chris K
AGILITY_05	Robot must pivot inline to last part of the obstacle course	Spin the robot after landing to ensure it moves on to the last leg	1	15APR24	Chris K
AGILITY_06	Robot must gain speed and knock down as many pins as possible	Increase speed to create the force necessary to knock down pins	1	15APR24	Chris K

3.2 Security

3.2.1 Protection

The use of GitHub, Google Docs, and Sphero.edu incorporates protective measures like encryption, activity logging, and data integrity checks. GitHub encrypts code repositories to prevent unauthorized access, while activity logging in both GitHub and Google Docs tracks changes for detection of unauthorized modifications. These measures collectively defend against malicious or accidental threats, bolstering system security.

3.2.2 Authorization and Authentication

To fortify security measures for our project, we utilized GitHub and Google Docs to enforce strict authorization and authentication protocols. GitHub served as our primary platform for version control and collaboration, allowing only authorized team members to modify our code repositories, ensuring maximum safety. Meanwhile, our System Design Document on Google Docs featured authentication features such as timestamps and user verification, providing clear accountability for any changes made. By integrating these measures, we bolstered security, safeguarding our data and ensuring the integrity of our project.

3.3 Portability

GitHub, Google Docs, and Sphero.edu are built with portability in mind. They use code that can easily work on different machines and operating systems. These platforms prioritize simple languages that are known to work well everywhere. Sphero.edu is especially good at working the same way no matter where it's used. The robots are made to be portable as they come with a carrying case, allowing for easy access to bring anywhere. This also means they can be used on different computers without any problems, making them flexible and easy to use across various setups.

4. Requirements Confirmation/Stakeholder sign-off

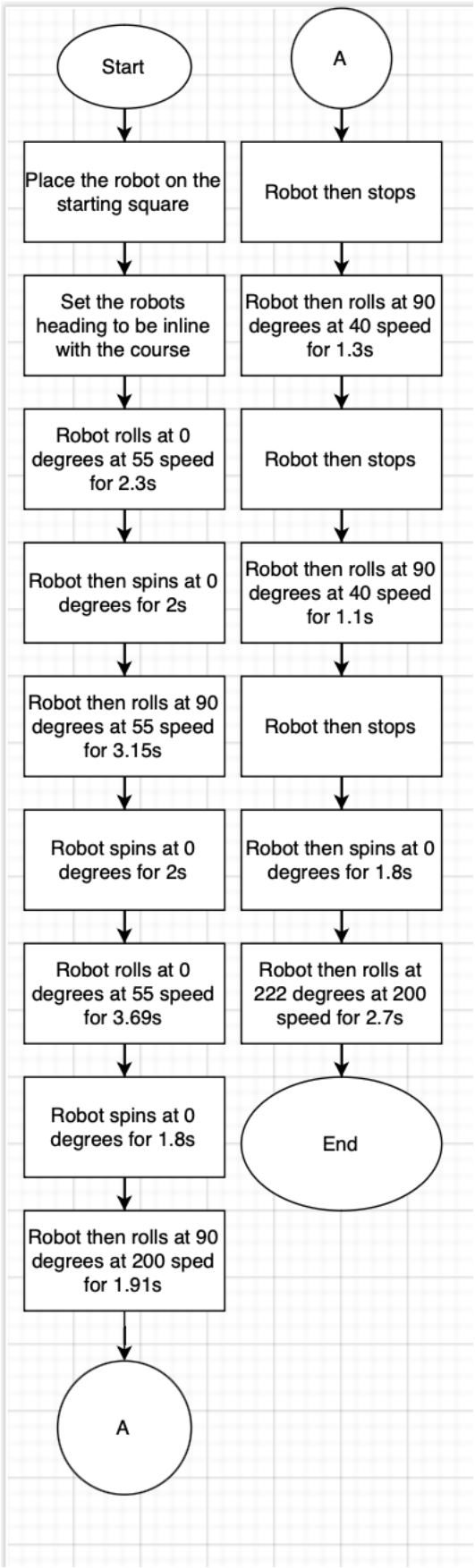
Meeting Date	Attendees (name and role)	Comments
04/15/24	Chris K and Aaron G	Confirmed all requirements, algorithm, and flowchart
04/16/24	Chris K and Aaron G	Confirmed code and recorded video

5. System Design

5.1 Algorithm

1.) Place robot on starting square
2.) Set robots heading inline with obstacle course
3.) Roll at 0° at 55 speed for 2.3s
4.) Spin 0° for 2s
5.) Roll 90° at 55 speed for 3.15s
6.) Spin at 0° for 2s
7.) Roll at 0° at 55 speed for 3.69s
8.) Spin at 0° for 1.8s
9.) Roll 90° at 200 speed for 1.91s
10.) Stop
11.) Roll 90° at 40 speed for 1.3s
12.) Stop
13.) Roll at 90° at 40 speed for 1.1s
14.) Stop
15.) Spin 0° for 1.8s
16.) Roll at 222° at 200 speed for 3.7s

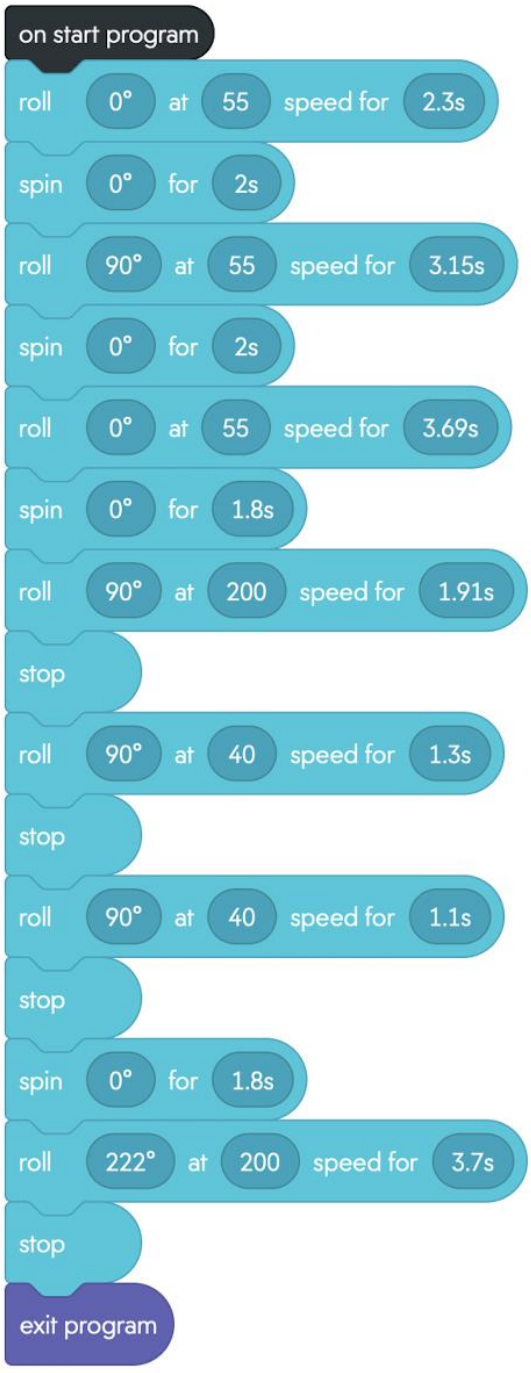
5.2 System Flow



5.3 Software

Mac Operating System for optimal results and feedback

Sphero Robot application for coding



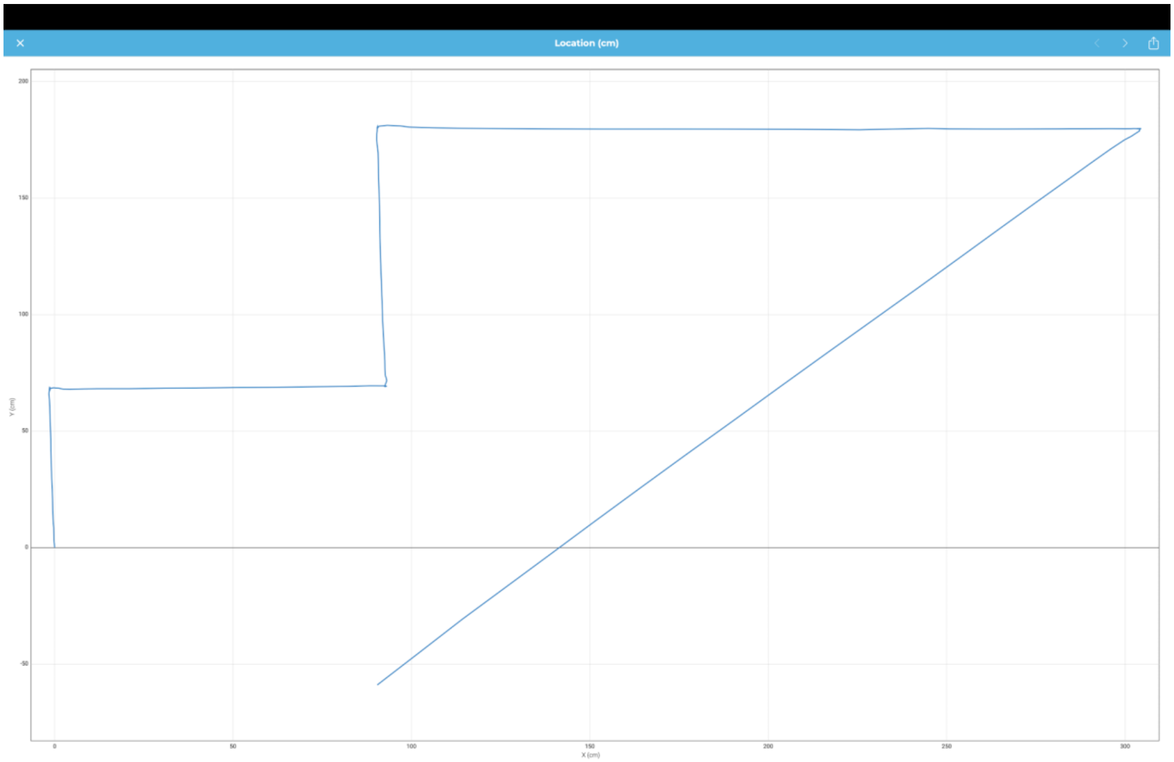
5.4 Hardware

Apple Mac computer for coding and accurate results following tests

Sphero robot

5.5 Test Plan

Reason for Test Case	Test Date	Expected Output	Observed Output	Staff Name	Pass/Fail
Ensure movement on first line and spins	16APR24	Moves along first line stops at correct time and spins inline to next point	Moves along first line and stops, spinning into second part of the course	Chris K	Pass
Ensure movement on the second line	16APR24	Moves along the second line and spins into next point	Moves along the second line and stops, spinning into second part of the course	Chris K	Pass
Ensure movement on third line	16APR24	Moves along the third line and spins into jump	Overshot the line signaling to the jump	Chris K	Fail
Ensure movement on third line	16APR24	Moves along the third line and spins into jump	Undershot the line signaling to the jump	Chris K	Fail
Ensure movement on third line	16APR24	Moves along the third line and spins into jump	Moves along the third line and stops, spinning into line with the jump	Chris K	Pass
Ensure has enough speed to hit the jump	16APR24	Moves along the line and hits the jump	Moves along the line and hits the jump	Chris K	Pass
Ensure robot is inline with the final stretch after the jump	16APR24	Moves and spins into line with the final obstacle	Landed and spun, falling short of the final line	Chris K	Fail
Added extra movement after landing the jump to move inline with final stretch	16APR24	Moves and spins into line with the final obstacle	Again fell short of the final line	Chris K	Fail
Added extra movement after landing the jump to move inline with final stretch	16APR24	Moves and spins into line with the final obstacle	Moves and spins into line with the final obstacle	Chris K	Pass
Moves swiftly down the final line knocking down pins	16APR24	Robot moves down line and knocks down pins	Robot missed left of the pins	Chris K	Fail
Adjusted heading on final sprint to knock down pins	16APR24	Robot moves down line and knocks down pins	Robot moves down line and knocks down pins	Chris K	Pass
Record Video of full sprint	16APR24	Robot completes full obstacle course with video proof of completion	Robot missed the pins to the right	Chris K	Fail
Record Video of full sprint	16APR24	Robot completes full obstacle course with video proof of completion	Robot completes full obstacle course with video proof of completion	Chris K	Pass



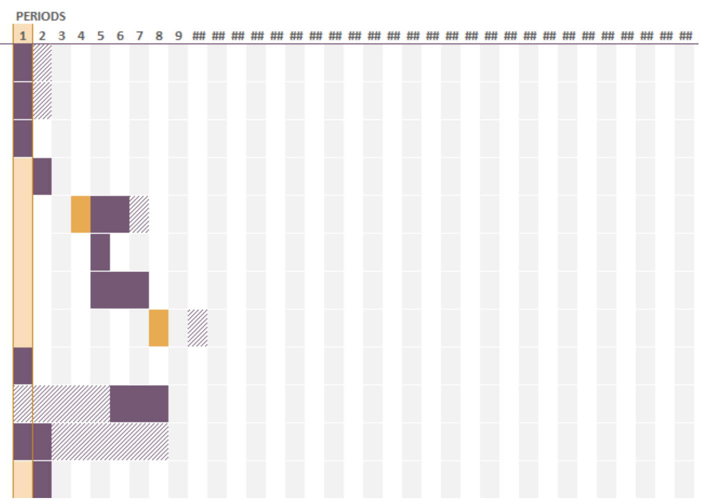
5.6 Task List/Gantt Chart

Sprint 3 - Agility

Select a period to highlight at right. A legend describing the charting follows.

Period Highlight: 1 | Plan Duration | Actual Start | % Complete | Actual (beyond plan) | % Complete (beyond plan

ACTIVITY	STAFF MEMBER(S)	PLAN START (Hours)	PLAN DURATION (Hours)	ACTUAL START (Hours)	ACTUAL DURATION (Hours)	PERCENT COMPLETE
Develop a plan (Gantt chart)	All team members	1	2	1	1	100%
Build requirements table	Chris Kenny	1	2	1	1	100%
Requirements Sign off Table	Chris Kenny	1	1	1	1	100%
Develop Flowchart	Aaron Guensch	2	1	2	1	100%
Create Block Code	All team members	5	3	4	3	100%
Adjust Sensor Data Diagram	Chris Kenny	5	1	5	1	100%
Document on Test Table	Aaron Guensch	5	3	5	3	100%
Film Robot Video	Aaron Guensch	10	1	8	1	100%
Staffing Plan	Chris Kenny	1	1	1	1	100%
Github Repository	All team members	1	8	6	3	100%
System Design Document (SDD)	All team members	1	8	1	2	100%
Algorithm	Chris Kenny	2	1	2	1	100%



5.7 Staffing Plan

Name	Role	Responsibility	Reports To
Chris Kenny	Leader/Developer	Develop, Ensure smooth operations, complete work	Professor Eckert
Aaron Guensch	Co-Leader/Developer	Develop and complete work	Chris
Skylyn	Quit	None	Themselves