James Allan

Christopher Aromando

Robert Betlesky

Table of Contents for Sprint 3

1. Executive Summary

1.1 Project Overview

The third and final test for the Software engineering triathlon is the agility test. Main goals of this test was to collaboratively design, form, solve and organize a successful software code using our robot. Using Sphero Edu, our group's purpose was to generate a code where our robot starts with a green light and says ‘ready set go’ and stops with a red light and says “I’m done and I need water” similar to the first and second test. As our robot performs, it was critical that our robot followed a well-defined path that was successfully coded through Sphero.edu, where it did not collide with any objects , was able to display agility through sharp and quick turns and successfully launched off the ramp to knock over the pins. The third test required the most trials to make sure the robot completed the course accurately and efficient. Our group planned to present this to our instructor and other classmates and be able to answer any concerns or questions. The rest of this document will cover all aspects and steps our group took to successfully complete the final trial and provide other background insight.

1.2 Purpose and Scope of this Specification

Our purpose was to successfully complete the third trial and be as accurate as possible. Due to unforeseen circumstances, our intended audience will be different than described in the project instructions. Our in scope specifications included a test by one of our group members on the rug after the code was generated. The robot performed to the best of its ability, but often swayed back and forth or wasn't tight on turns due to the uneven surface. Therefore, we strongly believe our robot would've performed 100% accurate if we were in the classroom. Limited space, lack of solid surface, not having "pins" and collaborating online are all aspects that we could not control and round up our out of scope specifications.

2. Product/Service Description

Overall, the Sphero Edu performed extremely well and was simple to use. We experienced no issues while holding the robot in our possession. The product was easily accessible due to its bluetooth connection, very durable and had no technical malfunctions. Not being able to test the robot in the classroom severely affected aspects and requirements of the project. However, the robot's functionality was never an issue. The robot has many unique features such as the ability to change color and speak out loud. Overall, the product was definitely user-friendly and there would have been no issues if the robot was tested in the classroom.

2.1 Product Context

No members in our group had previous experience with any product in this market. However, upon research Sphero's product seems to be the most popular. For example, we googled "Classroom Robot" and Sphero's were the first images. This product is independent and self contained in some aspects. Once connected to bluetooth and a code is generated, the user may start the program and the robot will go in motion independently. We are not sure if competitors products need user interaction throughout every phase, but we agreed Sphero was a great product for this project. Accessing the product was also extremely simple and very few software or hardware was required. All a user needed was a bluetooth connection and to download Sphero Edu. Based on experience with Sphero’s product, we believe other products would not perform as well as Sphero’s.

2.2 User Characteristics

We could definitely see this product being used for a variety of courses to perform different objectives due to its accuracy and accessibility. Therefore, our main users are students and faculty. The experience between students who used the product definitely varies. Since we used the robot in a 100 level course, some students had no software programming background or came from completely different majors. The product's simplicity definitely allowed for this to be possible because there was no intimidation aspect. In our opinion, minimal expertise is required and the main characteristic a user needs is patience and willingness to learn.

2.3 Assumptions

Some assumptions that might affect the requirements include equipment availability (we assume it’d be easy to meet in class and test, but we cannot hand over the robot to others and let other group members test since quarantine), testing space (we assume that we might have space to test in class, but since quarantine, we all might not have enough space at home to test the robot), a specific operating system is assumed to be available; if the operating system is not available, the Requirements Specification would then have to change accordingly.

2.4 Constraints

Any items that may constrain the design options include operation on an old computer or phone system, system resource constraints (e.g., limits on disk space or other hardware limitations), testing space (some tests require a lot of space which we just do not have), and other design constraints (e.g., design or other standards, such as programming language or framework).

2.5 Dependencies

List dependencies that affect the requirements.

- Algorithm has to be completed before anything else
- Sphero code has to be written before Sphero code testing can be done
- Accuracy documentation is taken after the Sphero code testing
- Accuracy robot testing can be done after all the code has been written and tested
- Accuracy code adjustments can be adjusted after the Sphero robot testing
- Accuracy documentation has to be updated after Sphero code adjustments

3. Requirements

3.1 Functional Requirements

Req#	**Requirement**	**Description**	**Priority**	**Date Rvwd**	**SME Reviewed / Approved**
AGIL_01	Avoid Obstacle 1	Be able to take a 90 degree turn going right past an obstacle	High	4/20	Chris A.
AGIL_02	Avoid Obstacle 2	Be able to take a 90 degree turn right again past an obstacle without knocking it over	High	4/20	Chris A.
AGIL_03	Avoid Obstacle 3	Be able to take a 90 degree turn past the third obstacle continuing towards the ramp	High	4/20	Chris A.
AGIL_04	Launch Off Ramp with Speed	Be able to complete the square and go over the ramp accurately	Med	4/20	Chris A.
AGIL_05	Knock Over Pins (marker)	Successfully land the jump and knock over pins	Low	4/21	Chris A.

3.2 Security

3.2.1 Protection

Some of the factors that will protect the system from malicious or accidental access, modification, disclosure, or misuse. For example, encryption, activity logging historical data sets, restrictions on intermodule communications, data integrity checks.

3.2.2 Authorization and Authentication

The role of the Pubcookie login server is that of the trusted, central authentication service. It interacts directly with users. It verifies usernames and passwords with backend authentication services. It issues cookies to users to provide single sign-on functionality and to application servers to provide authentication information.

The role of the Pubcookie application server is that of authentication enforcer. It redirects users who haven't been authenticated to the login server. It verifies authentication information returned from the login server. It issues cookies to users to maintain authenticated application sessions and provides user authentication information to applications.

3.3 Portability

Attributes of the system that can help relate to the ease of porting the system to other host machines and/or operating systems include percentage of components with host-dependent code, percentage of code that is host dependent, use of a proven portable language, use of a particular compiler or language subset, use of a particular operating system, the need for environment-independence - the product must operate the same regardless of operating systems, networks, development or production environments.

4. Requirements Confirmation/Stakeholder sign-off

Include documentation of the approval or confirmation of the requirements here. For example:

Meeting Date	Attendees (name and role)	Comments
4/20/20	James A., Robert B., Chris A.	Brainstormed way to approach code
4/20/20	James A., Robert B., Chris A.	Assembled documentation and information from tests

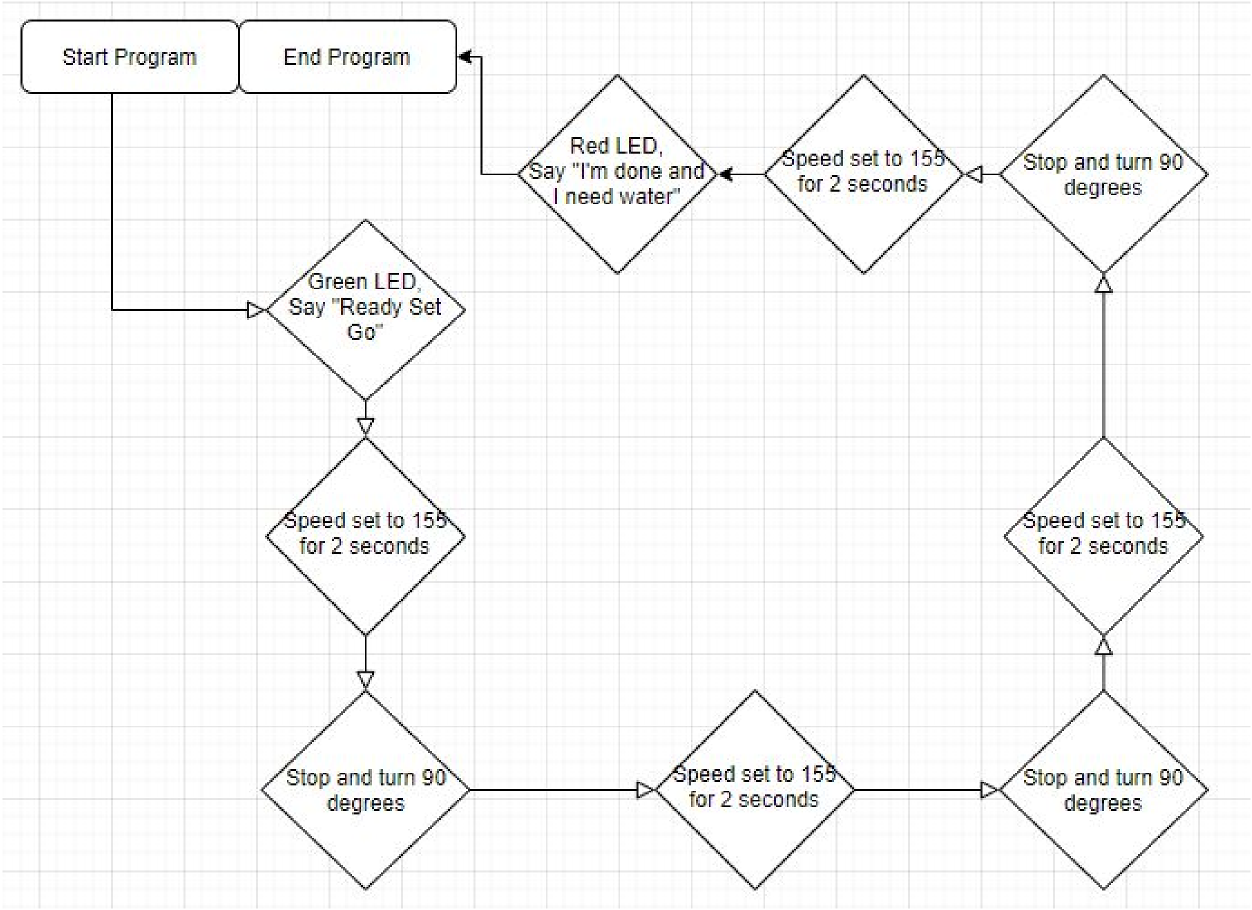
5. System Design

This section will provide all details concerning the technical design, staffing, coding, and testing the system

5.1 Algorithm

- Sphero will begin program
    - Enable green main LED and say “Ready set go”
    - Set speed to 150
    - Stop after 2 seconds, turn 90 degrees right
    - Set speed to 150
    - Stop after 2 seconds, turn 90 degrees right
    - Set speed to 150
    - Stop after 2 seconds, turn 90 degrees right
    - Set speed to max 255
    - Stop after 5 seconds
    - Sphero will enable red main LED and say “I’m done and I need water”
- Sphero end program

5.2 System Flow


Start Program
End Program
Red LED, Say "I'm done and I need water"
Speed set to 155 for 2 seconds
Stop and turn 90 degrees
Green LED, Say "Ready Set Go"
Speed set to 155 for 2 seconds
Speed set to 155 for 2 seconds
Stop and turn 90 degrees
Speed set to 155 for 2 seconds
Stop and turn 90 degrees

5.3 Software

Required Software:

- PC Operating System/Iphone iOS
- The SpheroBot uses block coding software available on the Sphero.Edu application.

5.4 Hardware

Required Hardware:

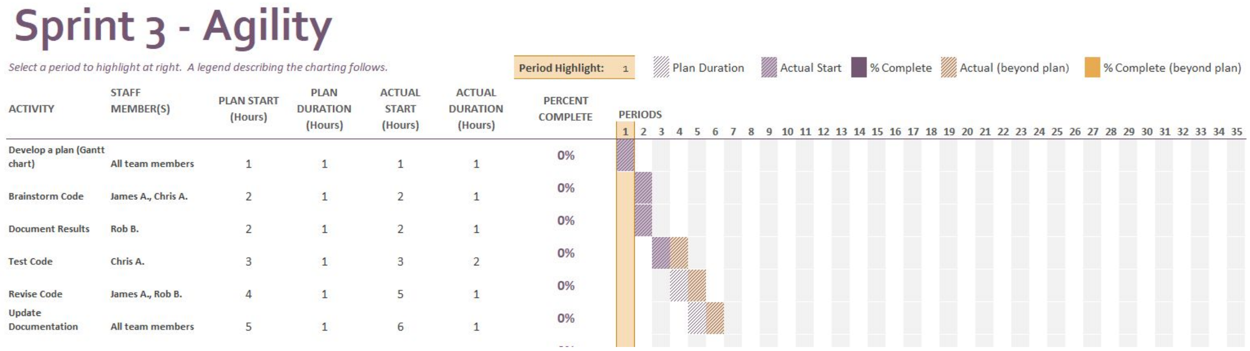
- SpheroBot
- Computer/Phone (Bluetooth Compatible)

5.5 Test Plan

Include a test plan showing all unit tests performed for this application, Include test rational, test date, staff member, pass/fail status

Reason for Test Case	Test Date	Expected Output	Observed Output	Staff Name	Pass/Fail
Speed Test	4/21	Find an appropriate speed to make turns at	Max speed is appropriate	CA	Pass
Turn Accuracy	4/21	Constant turns to form a square	Accurately performed desired turns	CA	Pass
Jump Accuracy	4/21	Maintain speed and jump off ramp to knock down pins	Robot successfully jumped and knocked down pins	CA	Pass

5.6 Task List/Gantt Chart



Sprint 3 - Agility

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

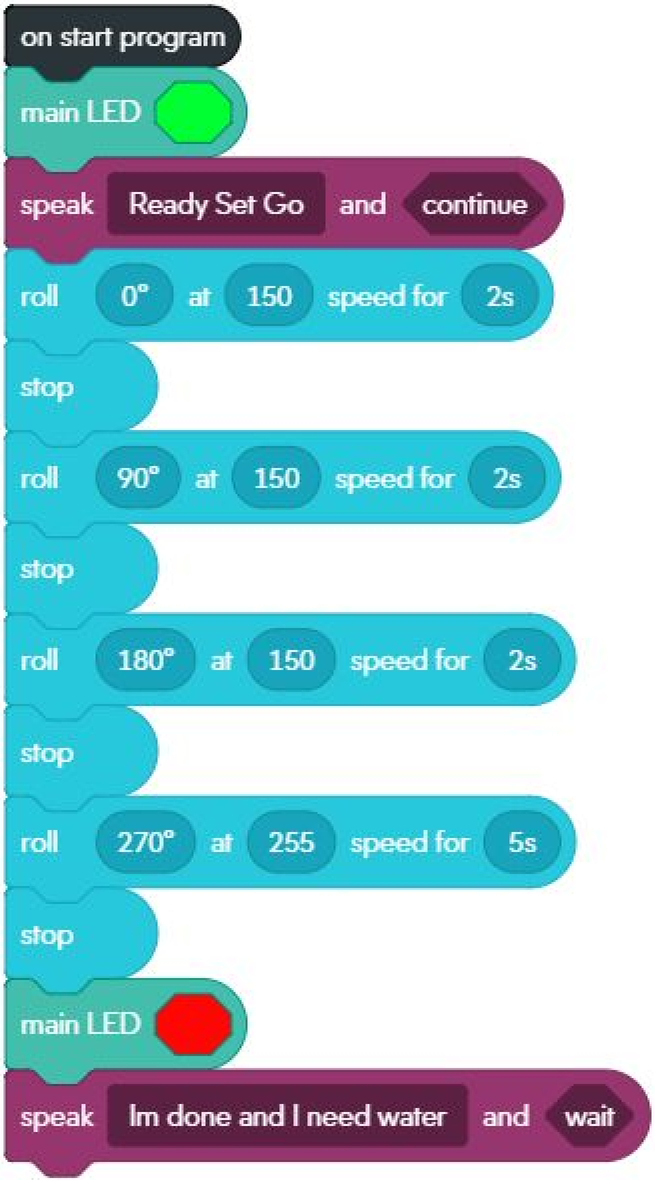
Period Highlight: 1

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	1	1	1	0%
Brainstorm Code	James A., Chris A.	2	1	2	1	0%
Document Results	Rob B.	2	1	2	1	0%
Test Code	Chris A.	3	1	3	2	0%
Revise Code	James A., Rob B.	4	1	5	1	0%
Update Documentation	All team members	5	1	6	1	0%

5.7 *Staffing Plan*

Insert a chart/table that depicts the roles and responsibilities of each team member that worked on this project

Name	Role	Responsibility	Reports To
James A.	Coding/Documentation	Develops shero code to be tested and inputs documentation.	Robert B.
Chris A.	Testing/Coding	Tests sphero code with robot and adjusts accordingly.	Robert B.
Robert B.	Documentation/Group Management	Inputs documentation and manages group time on different aspects.	Professor Gil Eckert



```
on start program
main LED 🟩
speak Ready Set Go and continue
roll 0° at 150 speed for 2s
stop
roll 90° at 150 speed for 2s
stop
roll 180° at 150 speed for 2s
stop
roll 270° at 255 speed for 5s
stop
main LED 🟥
speak Im done and I need water and wait
```