Sprint 3 - Agility Design Document

November 24, 2020

By: Eric Brew, Estuar Calle, Ryan Berenger

Table of Contents

[1. Executive Summary 3](#_Toc57840818)

[1.1 Project Overview 3](#_Toc57840819)

[1.2 Purpose and Scope of this Specification 3](#_Toc57840820)

[2. Product/Service Description 3](#_Toc57840821)

[2.1 Product Context 3](#_Toc57840822)

[2.2 User Characteristics 3](#_Toc57840823)

[2.3 Assumptions 4](#_Toc57840824)

[2.4 Constraints 4](#_Toc57840825)

[2.5 Dependencies 5](#_Toc57840826)

[3. Requirements 5](#_Toc57840827)

[3.1 Functional Requirements 6](#_Toc57840828)

[3.2 Security 7](#_Toc57840829)

[3.2.1 Protection 7](#_Toc57840830)

[3.2.2 Authorization and Authentication 7](#_Toc57840831)

[3.3 Portability 7](#_Toc57840832)

[4. Requirements Confirmation/Stakeholder sign-off 8](#_Toc57840833)

[5. System Design 8](#_Toc57840834)

[5.1 Algorithm 8](#_Toc57840835)

[5.2 System Flow 10](#_Toc57840836)

[5.3 Software 11](#_Toc57840837)

[5.4 Hardware 11](#_Toc57840838)

[5.5 Test Plan 11](#_Toc57840839)

[5.6 Task List/Gantt Chart 12](#_Toc57840840)

[5.7 Staffing Plan 12](#_Toc57840841)

# Executive Summary

## Project Overview

This is a project where the group has to code a robot with *Sphero Edu* to be able to complete a series of steps for it to finish the track for an agility test. The product must go through a course filled with three obstacles, turns, a ramp, and knocking over pins. All the requirements must also be met chronologically when the product is on the course. An algorithm was made in order to break up the requirements and the coders to easily understand what is needed to be achieved with the course. Reports and measurements were made to capture if the robot properly executed the code given to it. External devices were used to communicate the code and to log the actions the product did. The intended audience for the product is people that are interested in coding and want to learn the basics of computer science.

## Purpose and Scope of this Specification

The purpose of these specifications is to show how the robot should work in different environments and what it can do. The intended audience, the people interested in learning how to code, should read these specifications to know how the project worked in scope and out of scope. The in scope and out of scope include all the following:

In scope

* Robot must complete the course by not knocking down obstacles and hitting the pins
* Robot should follow the code that is given

Out of Scope

* The robot will only work with the block coding on *Sphero Edu.*
* The robot only really works on a hard flat floor.

# Product/Service Description

General factors that can affect the performance of the product and its requirements can be detrimental for the robot to work properly. If the product has at least one factor preventing it from doing the necessary requirements, nothing would work properly or not work at all. This product is dependent on other hardware and users that missing an important piece to execute performance can lead to it not happening or working improperly. The user must always look into these issues before the robot performs the tasks that are being asked to do.

## Product Context

The product relates to other products by connecting to them for the code needed to perform actions. The product is not independent or self-contained due to it needing external devices to connect the code to it. Related systems can interface with the product due to how the same software is used with both the external devices and robot. The major components of the larger system are used for inputs, sensory data, timing, and output. The interconnections between external devices and the product is the communication for the code to be translated into actions that are needed to be followed. External interfaces for the product are USBs to connect it to other peripherals.

## User Characteristics

The general customer profiles for each type of user that will use the product include:

* College students that have minimum experience and technical expertise for them to follow and improve their skills in programming. They could test out the code they write if the product does what is intended or fails. This teaches the students that they would need to do trial and error to have the product perform efficiently as intended.
* Staffs that have experience in programming and technical expertise so they could give to students to act out what is being taught in class. The staff must know what to do first, so they teach students how the product functions and pass on the knowledge.
* Faculty that knows programming and are technical expertise’s that could store and take care of the robots and give out when they are needed for certain classes. The product is fragile and expensive so a person that knows how to use them and care for them properly is needed.
* Children that are eight and over with no technical expertise or programming experience can learn to program at a young age due to the robot and code being easy to use. This can help prepare them for a future in programming and is the starting point to learn.

## Assumptions

Assumptions the user has can affect the requirements that are being achieved. The user not having equipment available to them, like an external device or robot, could lead to the performance not being performed under those two circumstances. The robot cannot move without the user using an external device to command it and a robot is needed to act out the commands ordered by the external device. Battery life can affect performance by the LED lights flashing red when they are not supposed to and could stop in the middle of a command if the battery runs out. This can be fixed by charging the robot before any performance is done Outdated hardware and software can affect the requirements by the software and hardware not connecting to the robot. This can be changed upgrading the software and/or hardware to either the latest version or a compatible version with the robot. The environment can also affect requirements by the commands producing different results. If the robot is in a small room instead of a larger room, code made for the larger room would have to be scaled down to accommodate the requirements needed. These assumptions can be fixed properly if the user adjusts the situation properly.

## Constraints

Constraints to the product include the following:

* For audit trails and log files to be recorded properly, the product must be at a certain distance from the external device for both to communicate with each other in real time
* The programming for the product works in a block code program that is the same brand as the product
* Hardware limitations on the product include:
  + the robot only going up to a speed setting of 225
  + Has a limited battery life for about 1-2 hours
  + The type of transportation the robot has is rolling
* External devices must have enough storage to download information gathered by the product
* Operating systems and hardware older than these are not supported
  + iOS 10+
  + Android 5+
  + Fire 7 (2019)
  + Fire HD 8 (2015+)
  + Fire HD 10 (2015+)
  + Windows 10 (1709+)
  + macOS (10.12+)
  + Chrome OS (50+)
* Supported Browsers
  + Chrome
  + Safari
  + Firefox
  + Internet Explorer
* Parallel operation with an old system includes when the product’s software and external software is being updated through the internet
* The user must access the code at a regular basis to find if the code has any mistakes in it and see if the code made is secure
* The product reads the code that is critically of the application and performs what is needed so the user must be careful what they put in the code
* Distance away from campus causes lack of access to testing room sometimes
* The group had to work mostly virtual and most of the group would work in person on the course
* Scheduling the group had affected when each person would work on the project

## Dependencies

The product has dependencies that affect the requirements listed for the performance:

* The product requires firmware updates 1-2 times a year to fix bugs and improve performances.
* Code is needed to command the product before the needed requirements are reached.
* This new product requires a separate device to measure and command for the actions that are needed to be performed.
* The product should undergo many small tasks and test plans before focusing on doing a course.
* An algorithm should be tested and completed before any final code can be built for the endurance test.

# Requirements

To be able to complete this task you will need a robot that will be provided to you and it needs to be able to complete a course while not hitting obstacles, going over a ramp, and hitting a number of pins. The dimensions can be modified to accommodate space. The user will need a robot that will be coded by using *Sphero Edu* which uses block code which will be the code used to complete this project.

|  |  |
| --- | --- |
| Requirement | Priority |
| 1. Power | 1.This requirement is needed to make sure the robot is working properly when it is on the starting point and ending point so it would not stop in the middle of the course or go past and be able to turn on each curve of the track. |
| 2.Movement | 2. The value of the requirement is needed for the robot to move through the obstacle course to not hit the three obstacles. The functionality the requirement provides is to check the robot moving by itself and that it could do this requirement. The requirement is also modifiable since the dimensions can be changed to accommodate different settings. |
| 3.Orientation | 3.The requirement is important to the overall structure for the system because it is needed to turn each time the product hits a corner. This requirement improves the structure by having the robot go in one motion throughout the course without manually changing it for it to move on corners. |
| 4.Collision | 4. This requirement is needed for the robot to hit the items that are intended to be hit. The requirement should help the robot hit as many pins as possible. |
| 5. Speed | 5. This requirement is needed for the robot to go at a certain speed the course requires it to go. This should help the robot go at a speed to go over the ramp. |
| 6.Communication | 6. This requirement is essential for the whole group because it allows people to be on the same page and all have the same goal for the robot before it starts moving. The group that made the project were doing this in person and in a virtual setting. |
| 7.Portability | 7. The requirement is needed to transfer information between software or hardware that is being used with the robot. Other hardware, software, and the cloud can access the code and other information pertaining about the robot when a user ports it. The portability can help transfer information from the external devices to the robot and vice versa with the information it has. |

* How the system should work:
  + When the robot starts, go straight for 3 feet and 3 inches at 180 degrees, but avoid hitting the first obstacle. Next, the robot should turn 90 degrees and run straight for 3 feet and 3 inches at 270 degrees while avoiding the second obstacle. The robot should turn 90 degrees afterwards and go straight for 3 feet 10 inches and not hit the last obstacle at 184 degrees. The robot would turn 90 degrees and go straight for 7 feet 2 inches at 275 degrees and 270 degrees for the robot to adjust while going over the ramp in the process. The robot would turn 45 degrees. The next step would be the robot would go straight for 9 feet and 1 inch at 45 degrees resulting in the pins falling.

## Functional Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Req# | Requirement | Comments | Priority | Date Rvwd | SME Reviewed / Approved |
| AGILI\_01 | Power | The robot had full power when doing the course. | 1 | 12/2/2020 | Reviewed and Approved |
| AGILI \_02 | Movement | The robot moved the way it was intended to base on the code. | 2 | 12/2/2020 | Reviewed and Approved |
| AGILI\_03 | Orientation | The robot successfully rotated on the corners. | 3 | 12/2/2020 | Reviewed and Approved |
| AGILI\_04 | Collision | The robot knocked over only the pins. | 4 | 12/2/2020 | Reviewed and Approved |
| AGILI\_05 | Speed | The robot should have enough speed to move on the ramp. | 5 | 12/2/2020 | Reviewed and Approved |

## Security

### Protection

Protection from malicious or accidental access, modification, disclosure, destruction, or misuse is crucial for the user and the product. Examples include the following:

* Encryption on the external devices by having passwords, which would allow the user to unlock any device used alongside the product.
* When using the product with an external device, be on a stable internet connection due to how data could be collected if not used in a private connection
* Validate the code and the changes being made to it
* Log and record every action the robot takes in order to see if the code given was followed as intended to make sure nothing touched the code while the user was gone
* Keep historical data to make sure the code written was performed properly the last time the user used and to see if it carries out same actions again
* Transferring code and data to people must be used properly and share it to the ones the user is intended it to send it to
* On the SpheroEdu app, private the code that is being written so it would not be available to other users
* Look on the external device if the robot being connected is the one the user is currently using and see if any other devices are connected to the robot
* Back up the code or any data collected to the cloud or an external device in case a device used with the robot could be stolen

### Authorization and Authentication

The authorization and authentication used for the project were the following:

* Allowing the product to connect to the external device being used in order to continue
* Logging on the *SpheroEdu* on through a username or password to access the saved code
  + Highly recommended to not save username and password on any external device in case that device gets stolen
* Have username and/or passwords on the external devices to keep other people from directly changing the code on the *SpheroEdu* account given access to
* The owners of certain documents in this project gave full editing access only to other members of the group and is shared privately
  + Block Code
  + The project document
  + Flow Chart
* Users that own code must authorized what another user can and cannot do to the code on *SpheroEdu* like viewing or editing

## Portability

Since portability is a requirement, the attributes of the system that relate to the ease of porting the system to other devices and operating systems, examples include the following:

* The percentage of components with host dependent code was about 90 percent due to how the code can be changed at any time except for starting and ending.
* Percentage of code that is host dependent is about 80 percent of the code due to the host having to manually change the speed per seconds and angle to accommodate for the rectangular course. Starting and ending is not host dependent.
* The use of a portable language is that once you start working on the code you can save it and it remains on your laptop which you can bring places.
* The compiler that is used to translate the block code is *Javascript* for the code to transfer to many hardware and software
* The use of particular operating systems that had the code ported to each other were the phone software *iOS 14* and *Android 10*
* The need for environment independence helps show how the product operates the same with any hardware, software, and networks. The code should properly be ported and usable with any device that is supported with the *SpheroEdu* app and robot.

# Requirements Confirmation/Stakeholder sign-off

|  |  |  |
| --- | --- | --- |
| Meeting Date | Attendees (name and role) | Comments |
| 12/3/20 | Ryan- Project Manager/Coder  Eric- Tester/Coder | All functional requirements |

# System Design

The system design for the project is used to follow a course properly under the block code being written by coders. The staffing for the design of the project occurred when the course was shown so the robot can be in a set path for agility. Staff in the system design used and shared the block code to observe how the robot reacts to each command. The code went through trial and error to find a way to have the robot follow the course. An algorithm was made for the coders to follow and understand what actions the robot should do when it is on the course. The testing for the system design was done in person on the course provided by the class. Simple tasks were given to the robot to follow at first and moved to more complex tests to follow to course.

## Algorithm

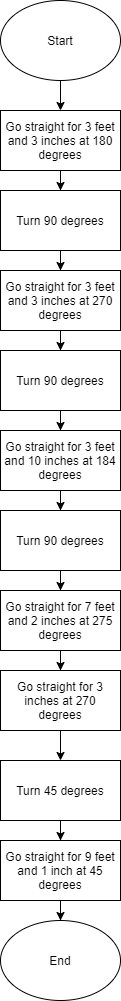
* Start
* Go straight for 3 feet and 3 inches at 180 degrees
* Turn 90 degrees
* Go straight for 3 feet 3 inches at 270 degrees
* Turn 90 degrees
* Go straight for 3 feet 10 inches at 184 degrees
* Turn 90 degrees
* Go straight for 7 feet 2 inches at 275 degrees
* Go straight for 3 inches at 270 degrees
* Turn 45 degrees
* Go straight for 9 feet 1 inch at 45 degrees
* End

A picture containing text, green, remote control

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generated

## System Flow



## Software

The software used to deploy this application for the robot is *Sphero Edu*, a block program that runs in J*avascript*, to perform the tasks mentioned in the code. The application programming interface includes blocks to have the robot change direction, move at certain speeds, speak phrases, and change light color. *Sphero Edu* is compatible with an app for mobile devices and a website on computers. Bluetooth is needed for communication between the device and robot to follow the block code and for the device to measure the movement. External software used to access the *Sphero Edu* app and website for this project were *Android 10*, *iOs 14*, *Windows 10, and MacOs 10*.

## Hardware

Hardware used for the endurance test is a clear *Sphero* brand robot that rolls in any direction, has a built-in voice box, and LED lights inside it that can change in any color. Other devices include iPhones, a Samsung Galaxy phone, Macbooks, and a Hewlett-Packard computer that are used to communicate with the robot. The two iPhones used were iPhone Xs with 3GB of RAM, 64GB of capacity, A11 Bionic chip with 64-bit architecture as the CPU, neutral engine, Embedded M11 motion coprocessor, can film videos at 4K resolution, built-in rechargeable lithium-ion battery with 13 hours’ worth of battery. The Galaxy phone used in this project was a Galaxy s10 with 8GB of RAM, 128 GB of capacity, Octa-core chipset as the CPU, and a resolution of QHD+. The two Macbooks used were Macbook Pros with 10 hours of battery life, 16GB of memory, 256 GB of storage, and 1.4GHz quad‑core 8th‑generation Intel Core i5, Turbo Boost up to 3.9GHz, with 128MB of eDRAM as the processor. The Hewlett-Packard computer used in the project was a ProBook 6560b with a7200 rpm SMART SATA II hard drive, 128 GB of storage, 6 hours of battery life, Intel Core i5-2520M Processor (2.50 GHz, 3MB L3 cache, 2 cores/4 threads, 35 W) up to 3.20 GHz with Intel Turbo Boost Technology, and a Mobile Intel HM65 Express Chipset.

## Test Plan

*Chart, line chart

Description automatically generated*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Reason for Test Case** | **Test Date** | **Expected Output** | **Observed Output** | **Staff Name** | **Pass/Fail** |
| Start Up Robot | 12/2/2020 | Move past first obstacle | Hit obstacle | Eric & Ryan | Fail |
| Move past first obstacle | 12/2/2020 | Don’t hit the obstacle | Moved straight and did not hit the obstacle | Eric & Ryan | Pass |
| Get past second obstacle | 12/2/2020 | Move in straight lines past second obstacle | Moved straight and did not hit the second obstacle | Eric & Ryan | Pass |
| Get past second obstacle with fixed angle and speed and rotate at the corner | 12/2/2020 | Move past first and second obstacles and turn at corners | Robot moved past first and second obstacles turned at corners with no issues | Eric & Ryan | Pass |
| Get robot past three obstacles | 12/2/2020 | Follow straight lines past obstacles and turn at the last corner | Moved past all three obstacles,  at the last corner but went farther on third | Eric & Ryan | Pass |
| Get robot over ramp | 12/2/2020 | The robot has to have enough speed to go over the ramp | The robot hit all the pins | Eric & Ryan | Pass |
| Hitting pins | 12/2/2020 | The robot has to hit all the pins | The robot changed colors at the end of the course | Eric & Ryan | Pass |

## Task List/Gantt Chart



## Staffing Plan

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Role | Responsibility | Reports To |
| Ryan | Project Manager /Coder | Make sure everything is done on time | Eric and Estuar |
| Eric | Tester/Coder | Testing the algorithms that the robot runs by | Ryan |
| Estuar | Documentation/ Algorithm | Documents and oversees the software | Ryan |