Sprint 3 - Agility Design Document

December 3, 2020

Executive Summary

1.1 Project Overview

The product of the specifications is to make the Sphero Robot run an obstacle course with specific dimensions through the Sphero Edu block code. The robot will start in a square. First, avoid three objects. Then the robot will go over a ramp. Finally, the robot will knock over as many pins as possible.The intended audience of this product is Professor Gil Eckert, the examiner of this project.

Purpose and Scope of this Specification

Describe the purpose of this specification and its intended audience. Include a description of what is within the scope and what is outside of the scope of these specifications. For example:

Activities that fall within the boundaries of the scope statement are considered “in scope” and are accounted for in the schedule and budget. If an activity falls outside the boundaries, it is considered “out of scope” and is not planned for

In scope

-phase 1 meets requirements of endurance sprint

-audience - professor, and programmers

- purpose - using block code to control robot movements

Out of Scope

- meets the requirements of the agility sprint, right now we are only focusing on the agility sprint

- meets the requirements of the agility sprint, right now we are only focusing on the agility sprint

2. Product/Service Description

2.1 Product Context

This product relates to other products because it uses modern technology and ideas that other products use. This product is not independent and self-contained because it requires the user to interact with an interface and write code for the product to respond to therefore it is dependent on the user. The robot does interface with the Sphero Edu software as it reads code from the file and reacts to it. The user develops block code in the Sphero Edu program from a computer. Then the Sphero robot receives these instructions and follows through with the code that the user input into the computer.

2.2 User Characteristics

Create general customer profiles for each type of user who will be using the product. Profiles should include:

- **Students - Olivia Bellino, Connor Bennett, Ludrianna Bazile**
- **Staff - Gil Eckert**
- **Student experience - Students have a bit of experience using python and using spheroEDU**
- **Staff experience - Professor has much experience using python and much experience in spheroID**
- **Other general characteristics that may influence the product - Not being able to meet up with group members often.**

2.3 Assumptions

List any assumptions that affect the requirements, for example, equipment availability, user expertise, etc. For example, 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.

- **The robot is required to test the block code the users create**
- **Only one group member has the robot and is able to run tests when we are not there.**
- **Users must know how to operate and create the block coding**

2.4 Constraints

Describe any items that will constrain the design options, including

- access, management and security - **no constraints with access, management & security**
- criticality of the application - **no criticality of the application**
- system resource constraints - **limited access to robot course room**
- other design constraints - **robot course has design issues with mini crevasse in the floor that can easily make the robot go off course.**

2.5 Dependencies

List dependencies that affect the requirements. Examples:

- **Gantt chart must be created before the requirements are completed**
- **Block code must be finished before running the robot course**
- **Algorithms must be completed before the flowchart is worked on because the flowchart is based on the algorithm.**
- **WIFI**
- **The course room must be available**

3. Requirements

3.1 Functional Requirements

In the example below, the requirement numbering has a scheme - BR_LR_0## (BR for Business Requirement, LR for Labor Relations). For small projects simply BR-## would suffice. Keep in mind that if no prefix is used, the traceability matrix may be difficult to create (e.g., no differentiation between '02' as a business requirement vs. a test case)

The following table is an example format for requirements. Choose whatever format works best for your project.

For Example:

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
ENDUR_01	Avoid three objects	Robot follows the path without knocking anything over	1	12/1/2020	
ENDUR_02	Go over a ramp	Robot enters and exits ramp successfully	1	12/1/2020	
ENDUR_03	Knock over pins	Robot hits multiple pins down	1	12/1/2020	

3.2 Security

3.2.1 Protection

Specify the factors that will protect the system from malicious or accidental access, modification, disclosure, destruction, or misuse.

- **encryption**
- **activity logging, historical data sets**
- **restrictions on intermodule communications**
- **data integrity checks**

3.2.2 Authorization and Authentication

Specify the Authorization and Authentication factors. Consider using standard tools such as PubCookie.

Authentication confirms your identity to grant access to the system. Authorization determines whether you are authorized to access the resources. It is the process of validating user credentials to gain user access. Authentication factors required for authorization may vary, depending on the security level.

3.3 Portability

If portability is a requirement, specify attributes of the system that relate to the ease of porting the system to other host machines and/or operating systems. For example,

- Percentage of code that is host dependent- **75% dependent, The host who has the robot is able to create and share code with other group members.**
- Use of a proven portable language- **SpheroEDU is 100% proven portable language and is able to be shared with every group member.**
- Use of a particular operating system - **SpheroEDU**
- 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
11/23/2020	all group members	ran the program for the robot and got our final video of the robot running the agility sprint

5. System Design

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

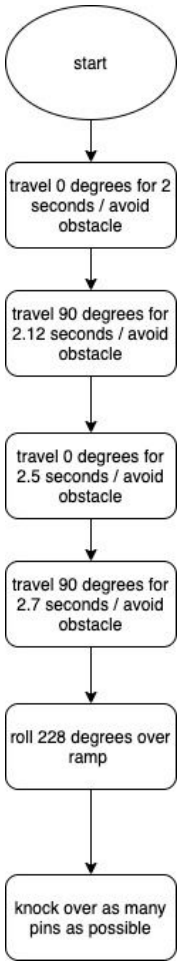
5.1 Algorithm

Develop and describe here the algorithm that will be used to provide the required performance of your software

- robot travels 0 degrees for 2 seconds and avoids the first obstacle
- robot travels 90 degrees for 2.12 seconds and avoids obstacle #2
- robot travels 0 degrees for 2.5 seconds and avoids obstacle 3
- robot travels 90 degrees for 2.7 seconds and avoids obstacle 4
- robot goes straight at high speed over ramp
- robot knocks over as many markers as it can

5.2 System Flow

Develop a flowchart (and show here) that accurately depicts how your software application will act to fulfill the algorithm



5.3 SOFTWARE

Describe software languages/platforms/api's used to develop and deploy this application

SpheroEDU
Microsoft Excel
Google Docs

5.3 Hardware

Describe hardware platforms that were used to develop, test and demonstrate this application

-Robot
-Laptops

5.4 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
Robot must move straight from the X	12/1/2020	Robot moves straight on the line without hitting the obstacle	Robot strayed from the line and hit the obstacle	Ludrianna	Fail
Robot must move straight from the X	12/1/2020	Robot moves straight on the line without hitting the obstacle	Robot moved on the line without hitting the obstacle	Ludrianna	Pass
Robot must turn 90 degrees and move straight	12/1/2020	Robot moves straight without hitting the obstacle	Robot moved on the line without hitting the obstacle	Connor	Pass
Robot must turn back -90 degrees and move straight	12/1/2020	Robot moves straight without hitting the obstacle	Robot moved on the line without hitting the obstacle	Connor	Pass
Robot must turn 90 degrees and pass over the ramp	12/2/2020	Robot passes over ramp and lands where in the correct spot	Robot overshot the ramp and did not land correctly	Olivia	Fail
Robot must turn 90 degrees and pass over the ramp	12/2/2020	Robot passes over ramp and lands where in the correct spot	Robot successfully scaled the ramp and landed in the correct spot	Olivia	Pass
Robot must turn to 225 degrees and knock down pins	12/2/2020	Robot knocks down pins	Robot strayed and missed pins	Connor	Fail
Robot must turn 225 and knock down pins	12/2/2020	Robot knocks down pins	Robot stayed on line and knocked down the pins	Connor	Pass

5.5 Task List/Gantt Chart

ACTIVITY	STAFF MEMBER(S)	PLAN START (Hours)	PLAN DURATION (Hours)	ACTUAL START (Hours)	ACTUAL DURATION (Hours)	PERCENT COMPLETE
Develop a plan (Gantt chart)	Connor Bennett	1	3	1	4	100%
Build requirements table	Ludrianna Bazile	1	2	1	3	100%
Build Staffing Plan	Olivia Bellino	1	1	1	1	100%
Build Flowchart	Olivia Bellino	1	3	1	6	100%
Build Block Code	All Team Members	1	6	1	8	100%
Test Sensor Data	All Team Members	1	3	1	4	100%
Film and Upload Robot Performance	Connor Bennett	1	1	1	1	100%
Receive and Examine Sensor Data Diagram	Ludrianna Bazile	1	4	1	5	100%
Create/Final Approval of Repository	Connor Bennett	1	1	1	2	100%
Coplete Systems Design Document	All Team Members	1	6	1	7	100%
Create Algorithm	Olivia Bellino	1	3	1	4	100%
Create Test Plan	Connor Bennett	1	5	1	6	100%

5.6 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
Olivia Bellino	project manager	Project managers play the lead role in planning, executing, monitoring, controlling and closing projects. They are accountable for the entire project scope, project team, resources, and the success or failure of the project.	Group manager
Connor Bennett	documentation	A Documentation Manager plans and directs documentation projects for timely delivery of documents, publications, and online content. May act as final approver or editor for projects. Additionally, a Documentation Manager may require an associate degree or its equivalent.	Project manager

Ludrianna Bazile	programmer tester	Organizing programs and activities in accordance with the mission and goals of the organization. Developing new programs to support the strategic direction of the organization. Creating and managing long-term goals. Developing a budget and operating plan for the program	Project manager