Sprint 3 Design Document

November 26, 2019

Use this Requirements Specification template to document the requirements for your product or service, including priority and approval (Must do).

This document will also serve as a System Design Document (How to) and will include sections detailing system flow, algorithms, staffing plan, software/hardware, and Test Plan

This document contains instructions and examples which are for the benefit of the person writing the document and should be removed before the document is finalized.

To regenerate the TOC, select all (CTL-A) and press F9.

Table of Contents

1. Executive Summary. 3

1.1 Project Overview.. 3

1.2 Purpose and Scope of this Specification. 3

2. Product/Service Description. 3

2.1 Product Context. 3

2.2 User Characteristics. 3

2.3 Assumptions. 3

2.4 Constraints. 3

2.5 Dependencies. 4

3. Requirements. 4

3.1 Functional Requirements. 5

3.2 Security. 5

3.2.1 Protection. 5

3.2.2 Authorization and Authentication. 6

3.3 Portability. 6

4. Requirements Confirmation/Stakeholder sign-off. 6

5. System Design. 6

5.1 Algorithm.. 6

5.2 System Flow.. 6

5.3 Software. 6

5.4 Hardware. 6

5.5 Test Plan. 7

5.6 Task List/Gantt Chart. 7

5.7 Staffing Plan. 7

# **1.** **Executive Summary**

## **1.1** **Project Overview**

Your robot will run the obstacle course. The course will start in a square. Then the robot will encounter 3 objects which it must avoid. Next, the robot will move to stage 2 square and go over the ramp. Then the robot will proceed to stage 3 square, knock over as many pins as possible and finish in the same square where it started. Points added for each obstacle the robot completes, for each obstacle avoided, for each pin the robot topples and for each square the robot stops in during its run.

## **1.2** **Purpose and Scope of this Specification**

This project is intended for educational purposes. The reason for the project is for students to become familiar with the principles of software design, and more specifically, how to work with the sphero robot.These small challenges allow students to test their skills in a low stress fun environment. Programming skills that are learned from this project can be applied to other projects that are on a larger scale.

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

## **2.1** **Product Context**

In order to test the limits of the sphero robot and the knowledge of the program, students will run the obstacle course set up by the professor.

## **2.2** **User Characteristics**

Students will be using this product to practice their coding, organizational, and cooperative skills. The drag-and-drop block coding of the sphero app allows beginner coders to tackle a project using the robot without being overwhelmed.

## **2.3** **Assumptions**

If the robot is not available for testing, the user would not be able to test the efficacy of their programming model. The unpredictable nature of the sphero’s movement makes creating models to predict its behavior extremely difficult. The spherical shape of the robot also creates a confounding variable in testing, where the simple physics of the robot make no two runs the same. The constraints of the obstacle course can also interfere with a theoretical model of how the robot will behave.

## **2.4** **Constraints**

Describe any items that will constrain the design options, including

· Physical shape of the robot

· Limited command structure

· Limited access to the robot

· Commuter students

· Time constraints

· Lack of a standard measurement of distance

· Lack of a standard measurement of speed

· The obstacle order

· Collision with objects changing the direction of sphero

## **2.5** **Dependencies**

List dependencies that affect the requirements.

· Requires a sphero app account

· Requires Bluetooth connectivity

# **3.** **Requirements**

## **3.1** **Functional Requirements**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Req # | Requirement | Comments | Priority | Date Rvwd | SME Reviewed / Approved |
| Sp-01 | Robot has to complete the obstacle course |  | 1 | 11/26 | Charles/Luke |
| Sp-02 | The robot must avoid all obstacles in its path successfully | Points added for each obstacle the  robot completes, for each obstacle avoided | 1 | 11/26 | Charles/Luke |
| Sp-03 | The robot will move to stage 2 square and go over the ramp | Points will be added for each pin that is knocked over | 2 | 11/26 | Charles/Luke |
| Sp-04 | The robot must proceed to the stage 3 square, knock over as many pins as possible, and finish in the same square where it started | The robot must go over the ramp in order to successfully complete the obstacle course | 3 | 11/26 | Charles/  Luke |

## **3.2** **Security**

### **3.2.1** **Protection**

Periodic checks on the physical integrity of the robot are paramount. Along with this, making sure the code stays up to date to the robot’s abilities.

## **3.3** **Portability**

JavaScript is a widely used coding language supported by most every modern device. Whether it is porting between computer operating systems, to mobile, or across robot models, the sphero app supports all modern operating systems and both iOS and Android.

# **4.** **Requirements Confirmation/Stakeholder sign-off**

|  |  |  |
| --- | --- | --- |
| Meeting Date | Attendees (name and role) | Comments |
| 11/5 | Brandon M., coder. Charles K., documenter. Luke M., planner/graphs. |  |

# **5.** **System Design**

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

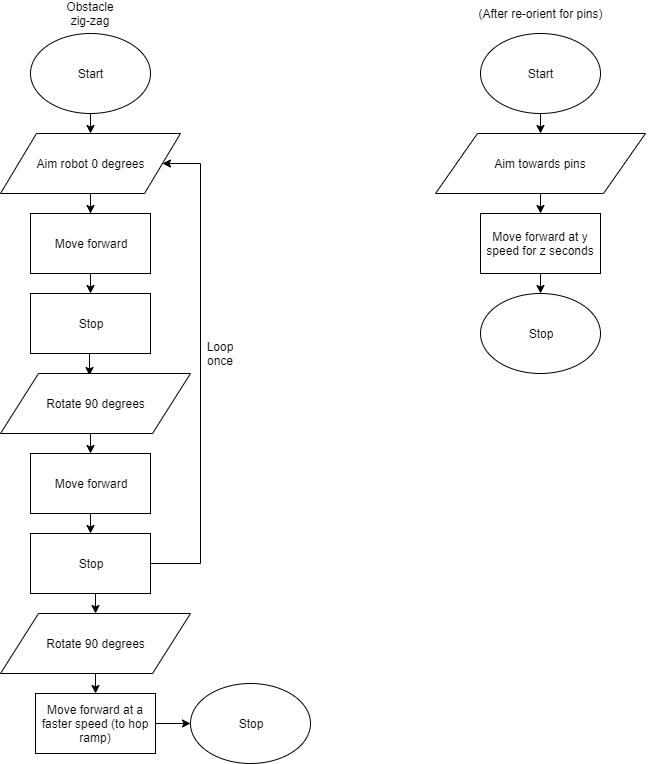
## **5.1** **Algorithm**

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

1. Start sphero and connect, start on ‘X’
2. Aim robot 0 degrees, move forward at y speed for z seconds
3. Stop
4. Rotate 90 degrees, move forward at y speed for z seconds
5. Stop, rotate to 0 degrees
6. Move forward at y speed for z seconds
7. Stop
8. Rotate 90 degrees, move forward at (faster) y speed for 2z seconds
9. Stop
10. Re-orient robot
11. Move forward at y speed for z seconds

## **5.2** **System Flow**

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

c

**5.3**

|  |  |
| --- | --- |
|  |  |
|  |  |

## 

## **Software**

JavaScript is used on the back end, but block code through the sphero app was sued to code and test.

## **5.4** **Hardware**

The sphero itself, two different laptops, and Bluetooth connectivity were used to perform the programs. Required for end users is any device that runs the sphero app, Bluetooth connectivity and a sphero unit are required.

## **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

5.5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Reason for Test Case** | **Test Date** | **Expected Output** | **Observed Output** | **Staff Name** | **Pass/Fail** |
| Decided on a set speed | 11/ 26 | Use a universal speed and only change timing | Only had to change times for the zig-zag. Speed 50. | Brandon | Pass |
| Test times for zig-zag | 11/26 | Stay on path with a universal time value of 2s per line | Stopped short of corners after 1st | “ | Fail |
| Test times for zig-zag T2 | 11/26 | Stay on path with an increased time on second corner (2.1s) | Stopped short on third still | “ | Fail |
| Test times for zig-zag T3 | 11/26 | Stay on path with an increased time on second and third corner (2.1, 2.5s) | Stayed relatively on path | “ | Pass |
| Test ramp jump | 11/26 | Go over ramp with sufficient speed and accuracy | Off path | “ | Fail |
| Test ramp jump T2 | 11/26 | Go over ramp with sufficient speed and accuracy | On path, sufficient speed, (162 for 2s) | “ | Pass |
| Re-orient and whero bowling | 11/26 | Knock down pins | Knocked down pins, different amount each time. (70, 2s) | “ | Pass |

## **5.6** **Task List/Gantt Chart**

## **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** |
| **Brandon Merola** | Coder | Code and write about testing in design doc | Team |
| **Charles Klehr** | Design Doc | Set up design doc | Team |
| **Luke Medley** | Planner/Graphs | Planning materials and flowchart | Team |