# Sprint 1 - Endurance Design Document

## November 10, 2021

## By Bianca, Noah, and Bajon

# 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 ....... 4
3.2. SECURITY ....... 5
*3.2.1. Protection ....... 5*
*3.2.2. Authorization and Authentication ....... 5*
3.3. PORTABILITY ....... 5

**5. SYSTEM DESIGN ....... 5**

5.1. ALGORITHM ....... 6
5.2. SYSTEM FLOW ....... 6
5.3. SOFTWARE ....... 6
5.4. HARDWARE ....... 6
5.5. TEST PLAN ....... 6
5.6. TASK LIST/GANTT CHART ....... 7
5.7. STAFFING PLAN ....... 7

# 1. Executive Summary

### *1.1. Project Overview*

This is a project meant to test the functions of a robot based off our own programming in the Sphero Edu app. It should be able to cover the length of the classroom track accurately and in moderate time. It also must complete a series of different functions such as change lights, and speak several phrases that indicate the start and finish of the program. In the System Design Document you will find the steps taken to create this program to the best of our ability.

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

The purpose of this project is to test our capability of creating a program that allows the robot to carry out the required tasks while also collaborating with others to maximize efficiency. It will also show various capabilities of the robot itself. The intended audience of this project includes those beginning with programming, and those curious of the robot's capabilities when given the correct code.

**In scope:**

- The robot will travel the rectangular shape of the room
- It will return to it's starting location
- The robot will traverse the course without interfering with any outside objects

**Out of Scope:**

- This robot will only work in the conditions of the room provided
- It will only be able to complete what the block code is able to provide

# 2. Product/Service Description

In this section, describe the general factors that affect the product and its requirements. This section should contain background information, not state specific requirements (provide the reasons why certain specific requirements are later specified).

### *2.1. Product Context*

How does this product relate to other products? Is it independent and self-contained? Does it interface with a variety of related systems? Describe these relationships or use a diagram to show the major components of the larger system, interconnections, and external interfaces.

### *2.2. User Characteristics*

- 3 students
- Limited coding experience

### *2.3. Assumptions*

Although experience is limited, users are aware of how to work block code program

### *2.4. Constraints*

- Only able to complete what the block code allows
- Cannot take up more space than computer allows

### *2.5. Dependencies*

List dependencies that affect the requirements. Examples:

- This new product will require a daily download of data from X,
- Module X needs to be completed before this module can be built.

## 3. Requirements

- Describe all system requirements in enough detail for designers to design a system satisfying the requirements and testers to verify that the system satisfies requirements.
- Organize these requirements in a way that works best for your project. See **Error! Reference source not found.Error! Reference source not found.**, **Error! Reference source not found.** for different ways to organize these requirements.
- Describe every input into the system, every output from the system, and every function performed by the system in response to an input or in support of an output. (Specify what functions are to be performed on what data to produce what results at what location for whom.)
- Each requirement should be numbered (or uniquely identifiable) and prioritized.
  See the sample requirements in Functional Requirements, and **Error! Reference source not found.**, as well as these example priority definitions:

  **Priority Definitions**

  The following definitions are intended as a guideline to prioritize requirements.

  - Priority 1 – The requirement is a “must have” as outlined by policy/law
  - Priority 2 – The requirement is needed for improved processing, and the fulfillment of the requirement will create immediate benefits
  - Priority 3 – The requirement is a “nice to have” which may include new functionality

  It may be helpful to phrase the requirement in terms of its priority, e.g., "The value of the employee status sent to DIS **must be** either A or I" or "It **would be nice** if the application warned the user that the expiration date was 3 business days away". Another approach would be to group requirements by priority category.

- A good requirement is:
  - Correct
  - Unambiguous (all statements have exactly one interpretation)
  - Complete (where TBDs are absolutely necessary, document why the information is unknown, who is responsible for resolution, and the deadline)
  - Consistent
  - Ranked for importance and/or stability
  - Verifiable (avoid soft descriptions like “works well”, “is user friendly”; use concrete terms and specify measurable quantities)
  - Modifiable (evolve the Requirements Specification only via a formal change process, preserving a complete audit trail of changes)
  - Does not specify any particular design
  - Traceable (cross-reference with source documents and spawned documents).

### *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 | Start with green light | | | | |
| ENDUR_02 | Say “ready set go” | | | | |
| ENDUR_03 | Travel from blue tape to middle of yellow floor tile | | | | |
| ENDUR_04 | Repeat above until back at starting location | | | | |
| ENDUR_05 | Stop with red light | | | | |
| ENDUR_06 | Say “Im done and I need water” | | | | |
| | | | | | |
| | | | | | |
| ENDUR_XX | | | | | |

## *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 via iPhone or Macbook password/facial recognition

### 3.2.2. Authorization and Authentication

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

## *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,

- Host dependent code
- Can be easily recreated
- Use of a particular operating system; MacOS

Requirements Confirmation/Stakeholder sign-off

| Meeting Date | Attendees (name and role) | Comments |
|---|---|---|
| 11/05/21 | Bajon, Bianca - project leader | Only Bianca present |
| 11/09/21 | Noah, Bajon, Bianca - project leader | Only Bianca present |

# 5. System Design

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

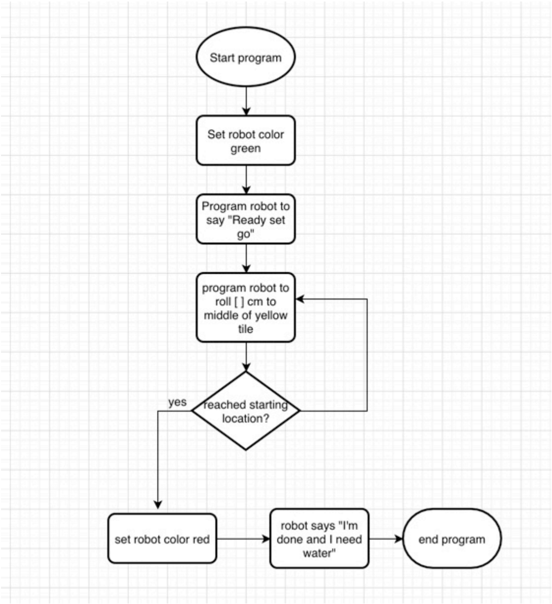
## *5.1. Algorithm*

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

- Set robot color to green
- Program robot to say “Ready set go”
- Program robot to move [    ] cm to the middle of yellow tile 1
- Repeat step 3 with each tile until robot has returned to starting position
- Set robot color to red
- Program robot to say “I’m done and I need water”

## *5.2. System Flow*

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


Start program
Set robot color green
Program robot to say "Ready set go"
program robot to roll [ ] cm to middle of yellow tile
reached starting location?
yes
set robot color red
robot says "I'm done and I need water"
end program

## *5.3. Software*

Used block code in Sphero

## *5.4. Hardware*

- Macbook Air
- Sprk Robot

## *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 |
|---|---|---|---|---|---|
| Develop block code | 11/05 | Run course/develop block code | Could not stay in straight line | Bianca | Fail |
| Develop block code | 11/09 | Start and finish block code | Overshot the line | Bianca | Fail |
| Develop block code | 11/10 | Start and finish block code |  | Bianca & Noah | Pass |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## *5.6. Task List/Gantt Chart*

Embed your gantt chart here

## *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 |
|---|---|---|---|
| Bianca | Leader | Develop block code/fill out SDD |  |
| Bajon | Collaborator | Help develop block code |  |
| Noah | Collaborator | Help develop block code |  |