Sprint 3 - Agility Design Document

November 30 2023

This document contains instructions and examples which are for the benefit of the person writing the document. All text in RED should be removed and replaced with information pertinent to your project.

Text in the finalized document must be **BLACK**.

This is the System Design Document (SDD) and will include sections detailing system flow, algorithms, staffing plan, software/hardware, and Test Plan

You must complete all sections of this document.

Where required by the Sprint Checklist you must embed images of some artifacts in this SDD

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

Table of Contents

1. Executive Summary

1.1. Project Overview

Describe this project or product and its intended audience, or provide a link or reference to the project charter.

The project aims to demonstrate the capabilities of a robot navigating a predefined track, serving as a scaled showcase of its mechanics, which foreshadows the larger-scale implementations planned for future launches.

1.2. Purpose and Scope of this Specification

This specification aims to lay out the precise conditions and limitations for the creation and application of a small robot that is programmed to follow a predefined path. The project team, engineers, developers, stakeholders, and anybody else involved in the design, development, and testing of the robot are all expected to use this document as a reference and set of guidelines. the specifications for the little robot's design and operation in order for it to follow a predefined path. Included but not restricted to are the following: design parameters for the robot's mobility and maneuverability; safety and fail-safe systems for emergency scenarios and obstacle avoidance; and definition of sensor needs for track detection and navigation. The following items are out of the scope of this particular specification but may be considered in subsequent phases or other documents:advanced features beyond track following (such as object recognition or advanced decision-making capabilities),modifications to the track itself or changes to the environment that the robot will traverse and integration with other systems or robots not directly related to the predetermined track.

2. Product/Service Description

The efficacy of the robot's traversal along the track is contingent upon the impediments it encounters, such as obstructions like a chair or an individual's foot. Upon collision with any of these obstacles, the robot will deflect and endeavor to navigate around them, thereby deviating from its initial trajectory. To ensure optimal performance and prevent impediments to the robot's motion, it is imperative to maintain a clear track throughout its course of operation.

2.1. Product Context

While the small robot primarily operates independently, its capabilities might involve interfacing with related systems or components. For instance, the robot could interact with a control station or central server responsible for initiating its course or providing updates. It might communicate with external sensors placed along the track for data acquisition or safety monitoring. Additionally, it could have interfaces for recharging or receiving software updates, enhancing its functionality and performance.

2.2. User Characteristics

The compact prototype exhibits remarkable potential for deployment in educational institutions globally, serving as a fundamental tool for comprehending the intricate nuances of engineering and robotics. Its user-friendly interface facilitates accessibility, rendering it suitable for individuals aged 15 and above. While prior background knowledge is advantageous for optimal utilization and intended functionality, the product's design accommodates users with varying levels of expertise in the field.

Other users may be hobbyists of engineering or anyone with the interest in robotics. The experience levels can vary ranges from minimal technical proficiency to advanced expertise, depending on the individual's background and interest in robotics.

2.3. Assumptions

Assuming that the necessary sensors, motors, and components for the robot's navigation are available and functional. If any of these components are not available or malfunction, the requirements might need adjustment to accommodate alternative equipment or reconfiguration. Also assuming that users or operators have a certain level of expertise in handling and maintaining the robot. If the users' expertise varies or is limited, the requirements might need adjustments to ensure the robot's usability and maintenance align with the users' skills.

2.4. Constraints

The design may be limited by the requirement for the new robot system to function in parallel with an existing, maybe older system. The old and new systems' compatibility, communication protocols, and interoperability will need to be taken into account.The necessity to include strong access control, efficient management procedures, and top-notch security measures will place restrictions on the system's architecture. The architecture and functionality of the system to guarantee data protection and restricted access to authorized workers will be impacted by these limitations.The design will be limited by the need for reliable audit functions, such as keeping an audit trail or log files. This calls for setting aside funds for logging and making sure the system conforms with audit trail requirements.

Dependencies

2.5. Dependencies

Certain modules or components within the system might be dependent on others. For instance, Module X (such as the navigation system) needs to be completed before Module Y (maybe obstacle avoidance or decision-making module) can be built. These dependencies influence the development order and integration of different functionalities within the robot.If the robot's operation is reliant on specific power sources or energy types, the requirements must accommodate these dependencies. This might involve ensuring the design is energy-efficient or has the capability to work with the available power sources.

3. Requirements

3.1. Functional Requirements

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
Agility_01	The robot starts in a square		3	11/30/23	Approved
Agility_02	The robot avoids the first object	Needed to get full credit	1	11/30/23	Approved
Agility_03	The robot avoids the second object	Needed to get full credit	1	11/30/23	Approved
Agility_04	The robot avoids the third object	Needed to get full credit	1	11/30/23	Approved
Agility_05	Robot will successfully go over the ramp	Needed to get full credit	1	11/30/23	Approved
Agility_06	Robot will knock over as many pins as possible	Needed to get full credit	1	11/30/23	Approved
Agility_07	The robot will get as many points as possible by knocking over pins and avoiding objects		2	11/30/23	Approved
Agility_10	The robot's speed is tbd	Multiple speeds will need to be used depending on if it is avoiding obstacel or trying to knock down all the pins	2	11/30/23	Approved

3.2. Security

3.2.1. Protection

Putting in place access restrictions, like authentication procedures, to guarantee that the robot's control systems can only be accessed by authorized people or systems. Encryption is used in communication protocols to protect the information sent and received by the robot as it tracks, making sure that it cannot be intercepted or manipulated by unauthorized parties.In order to secure the data sent and received by the robot while it is tracking its path, as well as to prevent unwanted parties from intercepting or altering the data, communication protocols also use encryption.

Authorization and Authentication

3.2.2. Authorization and Authentication

Enforcing complex password requirements, regular password changes, and preventing the use of easily guessable passwords enhances the strength of authentication. Employing multiple layers of authentication, such as a combination of passwords, biometrics (fingerprint, facial recognition), security tokens, or one-time codes, significantly enhances security by requiring more than one form of verification.

3.3. Portability

Wireless Control: The Sphero is controlled wirelessly through a smartphone or tablet app. This means you don't need any physical wires or connections, making it easy to use and carry around. The app communicates with the Sphero via Bluetooth, further enhancing its portability.

Rechargeable Battery: Sphero is powered by a rechargeable battery, which eliminates the need for disposable batteries or cumbersome power cords. This means you can charge it in advance and take it with you wherever you go without worrying about finding power sources.

Durable and Rugged Design: Sphero is built to be rugged and able to withstand various terrains and conditions. Its durability allows users to take it outdoors and use it in a variety of environments without the fear of damaging the device.

4. Requirements Confirmation/Stakeholder sign-off

Meeting Date	Attendees (name and role)	Comments
11/30/23	John Coscia(co-founder)	All confirmed
11/30/23	John Coscia(co-founder)	All confirmed

5. System Design

5.1. Algorithm

1: Connection to Sphero

- Establish Bluetooth Connection: Connect Sphero to the app via Bluetooth

2: Control Sphero Movements

- Implement Movement Control:
- Write code to send movement commands to the Sphero based on the predefined obstacle course. Ensure the robot successfully avoids obstacles and completes the course correctly

Step 3: Avoiding Objects

- Use the robot's sensors (gyroscope, accelerometer, or external sensors) to detect objects.
- If an object is detected, implement avoidance maneuvers (e.g., turn left or right).
- Continue moving forward once the path is clear.

Step 4: Going Over the Ramp

- Adjust the Sphero's speed and orientation to successfully navigate the ramp.

Step 5: Toppling Pins

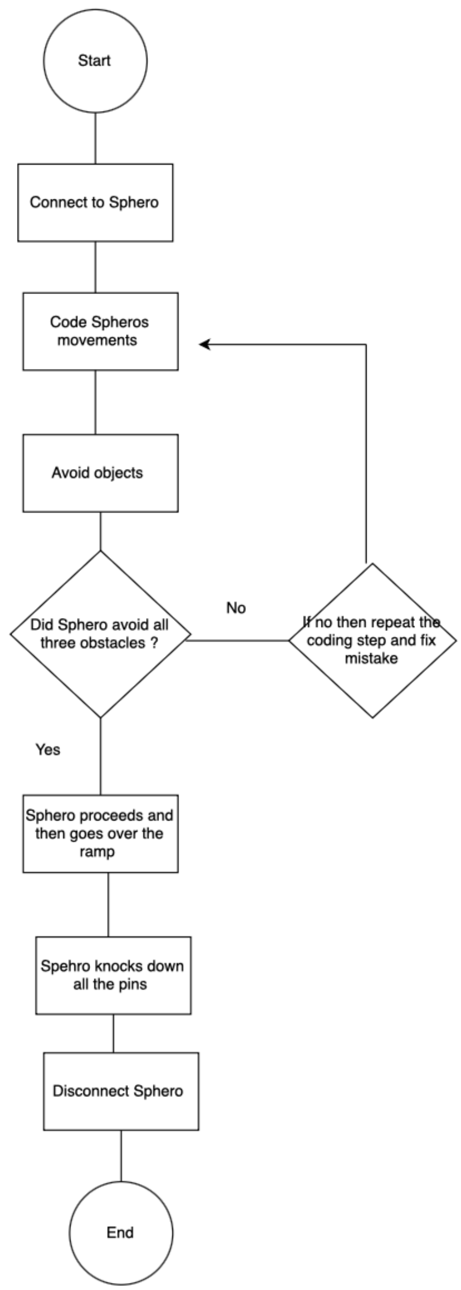
- Implement a mechanism to knock over the pins (e.g., increase speed and use a specific movement pattern).

Step 6: End of Course

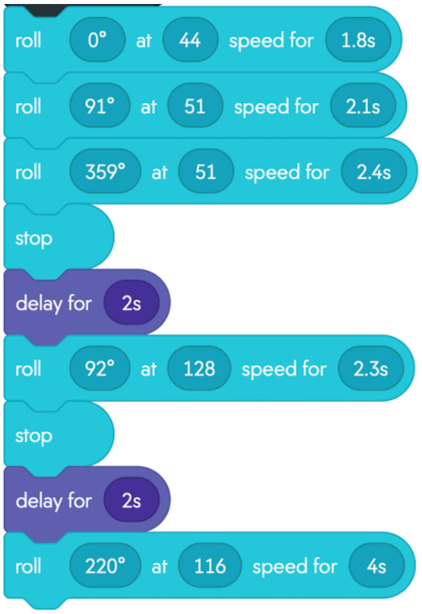
- Once the Sphero completes the entire course, stop its movements.

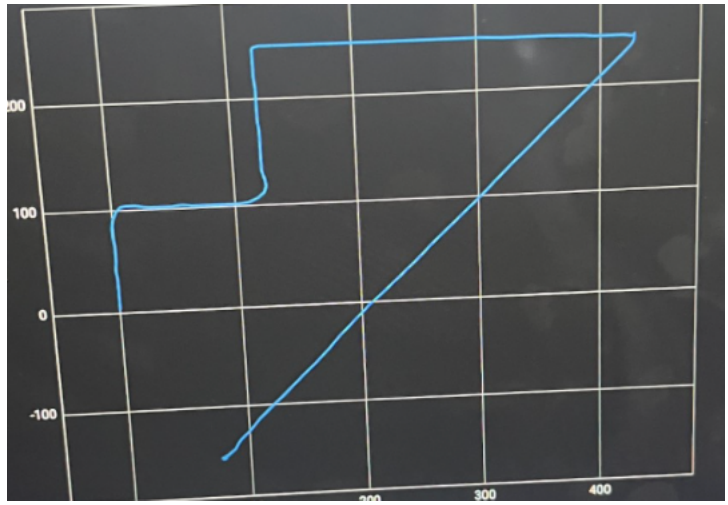
-Disconnect from Bluetooth

5.2. System Flow



5.3. Software





5.4. Hardware

Macbook Air and Sphero Sprk+

5.5. Test Plan

Reason for Test Case	Test Date	Expected Output	Observed Output	Staff Name	Pass/Fail
Agility	11/30/23	Sphero will avoid object in path	Hit the object and went off course	John	Fail
Agility	11/30/23	Sphero will avoid object in path	Went by object one with ease but hit into object two	Linda	Fail
Agility	11/30/23	Sphero will avoid object in path	Went by the first two objects then did not detect third one and hit it	Linda	Fail
Agility	11/30/23	Sphero will avoid object in path	Went by all three objects	John	Pass
Speed	11/30/23	Sphero will make it over the ramp	Shpero was too slow and could not make it up the ramp	Linda	Fail
Speed	11/30/23	Sphero will make it over the ramp	Speed was just right and made it over the ramp	Linda	Pass
Accuracy	11/30/23	Sphero will knock down all the pins	Sphero knocked down three pins	John	Fail
Accuracy	11/30/23	Sphero will knock down all the pins	Sphero knocked down two more pins then last test	John	Fail
Accuracy	11/30/23	Sphero will knock down all the pins	Sphero knocked down all but one pin	John	Fail
Accuracy	11/30/23	Sphero will knock down all the pins	Shpero knocked down all the pins	John	Pass

5.6. Task List/Gantt Chart

Sprint 1 - Endurance

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

Period Highlight: 1 | Plan Duration | Actual Start | % Complete | Actual (beyond plan) | % Complete (beyond plan)

ACTIVITY	STAFF MEMBER(S)	PLAN START (Hours)	PLAN DURATION (Hours)	ACTUAL START (Hours)	ACTUAL DURATION (Hours)	PERCENT COMPLETE	PERIODS 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57
Develop a plan (Gantt chart)	John, Linda	1	2	1	2	100%	
Build requirements table	John	3	2	3	1	100%	
Requirements signoff table	John	6	0.5	4	0.5	100%	
Algorithm	John	6.5	1	4.5	1	100%	
Flowchart	Linda	7.5	1	5.5	1	100%	
Blockcode	John	8.5	1	6.5	0.5	100%	
Sensor data diagram	John	9.5	0.5	7	0.5	100%	
Test table	John	10	1	7.5	0.5	100%	
Staffing plan	Linda	11	0.5	8	0.5	100%	
System design document	Linda	11.5	3	8.5	2	100%	
Robot video	Linda	14.5	0.5	10.5	1	100%	
Github repository	Linda	15	0.5	11.5	1	100%	
Activity 13						0%	
Activity 14						0%	
Activity 15						0%	
Activity 16						0%	
Activity 17						0%	
Activity 18						0%	
Activity 19						0%	
Activity 20						0%	
Activity 21						0%	
Activity 22						0%	
Activity 23						0%	
Activity 24						0%	
Activity 25						0%	
Activity 26						0%	

5.7 Staffing Plan

Name	Role	Responsibility	Reports To
John Coscia	Co-founder	Requirements, requirements signoff, algorithm, blockcode, sensor data diagram, test table,	Linda
Linda Pimpinella	Co-founder	Video, github, flowchart, SDD, staffing plan	John