Sprint 3- Agility Design Document

December 2, 2023

Table of Contents

1. Executive Summary

1.1

1.2 Project Overview

This project is showing our ability to problem solve, quantify, organize, document, code, test, and present a software system. For this project we will be working with our group overcoming an endurance based challenge relating to software engineering. We will be using a robot to show that we accomplished the task at hand. The intended audience for this project is the Professor and the rest of the students taking CS-101.

1.3 Purpose and Scope of this Specification

In Scope

This document addresses requirements related to Sprint 3 of the Robotics Project.

- modification of code to travel around obstacle course and not hit any objects to meet requirements
- modification of performance optimization for this sprint
- rule based classification

Out of Scope

The following items in Sprint 1 and Sprint 2 of the Robotics Project are out of scope.

- modification of code to travel around a square course
- modification of code to travel in a figure eight

2. Product/Service Description

2.1 Product Context

This product relates to other products in the fact that they are controllable robots. The difference between his product and others is that instead the robot can only perform in certain baths that are programmed into the robot. It is an independent product. It does not interact with a variety of related systems.

2.2 User Characteristics

- Student and staff will use this product
- Need coding experience
- Must have access to a computer or phone
- Technical expertise

2.3 Assumptions

- Assumed user has access to sphereo +, if not available must change requirement
- Assume user has access to Sphero Edu, if not can't complete the project
- Assumed that user has access to test room, can't complete tests if not available
- If user does not know how to access app, then requirements would have to change

2.4 Constraints

- Old software being used to run the code
- Different software being used to run the code
- Robot is damaged
- Device that has code doesn't have bluetooth
- Language is not Sphero Edu block code

2.5 Dependencies

- Code must be done on Mac or iPhone
- Robot must be fully charged
- Code must be up to date
- Robot synced with bluetooth to the coding device.

3. Requirements

3.1 Functional Requirements

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
AGIL_01	Robot makes it completely around the obstacle course	Required to get full credit	1	12/4	Approved
AGIL_02	Robot starts in a square	Required to get full credit	1	12/4	Approved
AGIL_03	Robot avoids all obstacles on obstacle course	Required to get full credit	1	12/4	Approved
AGIL_04	Robot successfully makes it over ramp on course	Get extra points if completed	2	12/4	Approved
AGIL_05	Robot knocks over bowling pins	Get extra points if completed	2	12/4	Approved

3.2 Security

3.2.1 Protection

- encryption
- password login
- private server
- constant server checks
- logging user activity

3.2.2 Authorization and Authentication

Only members of the group making the code and the professor will be able to see the code. This will make sure that the code isn't changed by a random person. It will be accessed through a link provided by one person.

3.3 Portability

- The robot is host dependent
- The device that is writing code must have Sphero Edu
- The environment must be flat and smooth that the robot is used in
- Their needs to be a good network connection

4. Requirements Confirmation/Stakeholder sign-off

Meeting Date	**Attendees (name and role)**	**Comments**
12/4	Conor(Recorder), Ziv(Problem Solver), Lucas(Coder)	Confirmed all of the Requirements

5. System Design

5.1 Algorithm

1.place the robot in the provided square

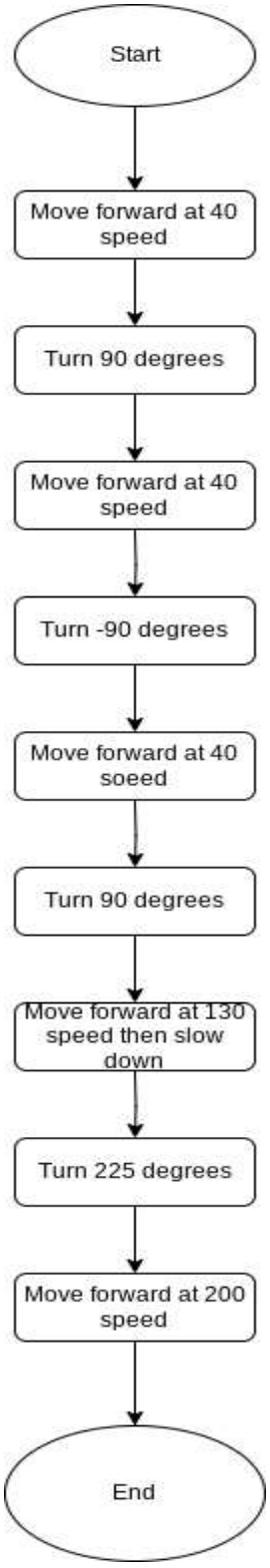
2.set the robot to run the obstacle course.

3. Set the robot to avoid the obstacles on the course

4. set the robot to go over the ramp

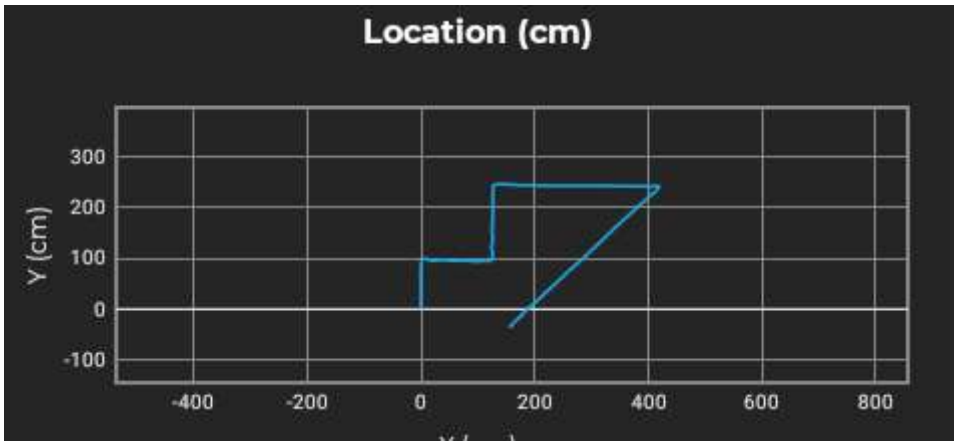
5.set the robot to knock over bowling pins.

5.2 System Flow



5.3 Software

The software program that was used was Spehro Edu. The language that was used was block code. Block code is a visual and easy approach to programming that uses puzzle-like pieces that link together to form code. It is very easy and user accessible for beginners.



5.4 Hardware

Hardware that was used for this Sprint was a computer. We used the computer to download the programing application Sphero Edu. The computer was used to develop the application. We used the

Spark Plus robot to test our application. To demonstrate our application we used a phone to record a video that will be uploaded online.

5.5 Test Plan

Reason for Test Case	Test Date	Expected Output	Observed Output	Staff Name	Pass/Fail
Test to see if the robot makes it past the first obstacle and stops at the corner.	12/4	Robot makes it past the first obstacle and stops at the corner without hitting an obstacle.	Robot made it past the first obstacle and stopped at the corner and didn't hit the obstacle.	Conor	Pass
Test to see if the robot makes it from the first corner to the second corner.	12/4	Robot makes it from the first corner to the second corner without hitting an obstacle.	Robot was a little short of the second corner and hit an obstacle.	Conor	Fail
Test to see if the robot makes it from the first corner to the second corner.	12/4	Robot makes it from the first corner to the second corner without hitting an obstacle.	Robot was a little long on the second corner and didn't hit the obstacle.	Conor	Fail
Test to see if the robot makes it from the first corner to the second corner.	12/4	Robot makes it from the first corner to the second corner without hitting an obstacle.	Robot made it from the first corner to the second corner didn't hit the obstacle	Conor	Pass
Test to see if the robot makes it from the second corner to the third corner	12/4	Robot makes it from the second corner to the third corner without hitting an obstacle.	Robot made it to the third corner from the second corner without hitting an obstacle.	Conor	Pass
Test to see if the robot makes it over the ramp.	12/4	Robot makes it over the ramp.	Robot was too slow to make it over the ramp.	Conor	Fail
Test to see if the robot makes it over the ramp.	12/4	Robot makes it over the ramp.	Robot made it over the ramp.	Conor	Pass
Test to see if the robot stops at the corner after going over the ramp.	12/4	Robot stops at the corner after going over the ramp.	Robot stopped a little after the corner	Conor	Fail
Test to see if the robot stops at the corner after going over the ramp.	12/4	Robot stops at the corner after going over the ramp.	Robot stopped at the corner.	Conor	Pass
Test to see if the robot rolls down the line of tape and knocks over all of the markers.	12/4	Robot rolls down the line of tape and knocks over all of the markers.	Robot rolled down the line of tape and knocked over all of the markers.	Conor	Pass

Reason for Test Case	Test Date	Expected Output	Observed Output	Staff Name	Pass/Fail
Test to see if the Robot makes it to the first corner, then second corner, then third corner without hitting an obstacle, makes it over the ramp, stops at the corner, rolls down the tape line and knocks over all of the markers.	12/4	Robot makes it to the first corner, then second corner, then third corner without hitting an obstacle, makes it over the ramp, stops at the corner, rolls down the tape line and knocks over all of the markers.	Robot completed everything but didn't knock over all markers.	Conor	Pass
Test to see if the Robot makes it to the first corner, then second corner, then third corner without hitting an obstacle, makes it over the ramp, stops at the corner, rolls down the tape line and knocks over all of the markers.	12/4	Robot makes it to the first corner, then second corner, then third corner without hitting an obstacle, makes it over the ramp, stops at the corner, rolls down the tape line and knocks over all of the markers.	Robot completed everything.	Conor	Pass

5.6 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)	**All team members**	1	1	1	1	**100%**
Build requirements table	**Conor**	2	2	2	1	**100%**
Algorithm	**Ziv**	3	1	2	1	**100%**
FlowChart	**Lucas**	4	2	3	2	**100%**

Block code	**Lucas**	4	2	3	3	**100%**
Sensor data diagram	**Lucas**	6	1	6	1	**100%**
Test table	**Conor**	6	1	6	1	**100%**
Staffing Plan	**All team member**	1	1	1	1	**100%**
System Design Document	**Conor**	5	4	5	3	**100%**
Robot video	**Ziv**	6	1	6	1	**100%**
Github Repository	**Lucas**	8	1	7	3	**100%**
Review SSD	**Ziv**	9	1	9	1	**100%**
Submit Sprint	**Lucas**	10	1	10	1	**100%**

5.7 Staffing Plan

Name	Role	Responsibility	Reports To
Ziv	Problem Solver	write the algorithm and film the robot, checks over SSD	Conor
Conor	Recorder	Record info and fill out the SSD	Lucas
Lucas	Coder	Write the flowchart and the code	Ziv+Conor