Sprint 3 – Agility Design Document

December 3, 2019

# 

# Executive Summary

## Project Overview

The robot is activated and programed through the Sphero.edu app. All team members will collaborate to create an algorithm that**is developed and tested to fulfill all functional requirements.**The team’s focus will be on the robot’s agility for sprint three. We will observe the robot’s agility during the obstacle course, in which the robot will circumnavigate around objects without letting it obstruct its run. Once the robot passes through the obstacle course and ends at the original starting point, the team can verify the algorithm works well. Hence, the robot proves its agility. We will demonstrate this to the class and the professor with documentation of the robot’s run to demonstrate the effectiveness of our algorithm and the overall agility of our robot during its sprint.

## Purpose and Scope of this Specification

In scope

This document addresses requirements related to phase 3 of the Software Development Project: Agility, sprint 3. Our audience is class CS-104 01.

Out of Scope

Phase 1 & 2 of the system development project is not included in this document.

# Product/Service Description

The robot should be charged and connected to the Sphero app to verify its capability to execute the algorithm. Additionally, to be sure the robot is not obstructed during its run, the team must develop a clear pathway in which the robot can pass through, during its navigation. Any barriers in the robot’s pathway could pose a possible delay to the robot’s potential in fully achieving the functional requirements, therefore there should be a pathway that allows the robot to travel around them. If physical barriers prove too much of an issue, all team members will work to improve the algorithm to suit accordingly.

## Product Context

The robot is relatively similar to other products in that it is programmed through a block code program of a user-friendly app. In the Sphero.edu app, team members create an algorithm to direct the robot through its sprint. Thus, the robot is dependent on the app and its user. Other outside sources that are important include Bluetooth connection. Without the link of the robot to the app, we cannot work on test runs. The connection of the robot to the app is just as important as an effective algorithm.

## User Characteristics

The general customer profile includes that of a Freshman student in CS-104 01 at Monmouth University with about 3-4 months' worth of experience on the Sphero.edu app and block code.

## Assumptions

Easy access to the Sphero.edu app is expected, this will ensure a smooth process in creating an algorithm for the robot. When the main operating system is working properly, the user should easily be able to adjust if necessary. Spacious areas, or lack thereof, may also force the user to adjust the speed and distances embedded in our algorithm.

## Constraints

There are many constraints team members have noted: trouble connecting the app to the robot, block code program on sphero.edu is not executed properly, lack of access to the app, not being able to create an efficient algorithm, all team members not being able to physically meet, lack of space at the testing site, a potential difference of measurements from software development project outline, and the robot not knocking down the pins.

## Dependencies

The robot must be fully charged before every test run. The user must make sure the robot is connected to the Sphero app to confirm a successful run. The block code algorithm must also be well-organized before a test sprint. The block code, in addition, should be according to the software development project outline, otherwise not effective.

# Requirements

## Functional Requirements

| Req# | Requirement | Comments | Priority | Date Rvwd | SME Reviewed / Approved |
| --- | --- | --- | --- | --- | --- |
| AGIL\_01 | Robot must start in the designated square of the obstacle course | Starting point | 1 | 12/3/20 | Members reviewed + approve |
| AGIL\_02 | Robot must roll 180 degrees at a 25 speed for 2 sec. |  | 1 | 12/3/20 | Member reviewed + approved |
| AGIL\_03 | Robot must roll 90 degrees at a 25 speed for 2 sec. |  | 1 | 12/3/20 | Members reviewed + approve |
| AGIL\_04 | Robot must roll 0 degrees at a 25 speed for 2 sec. |  | 1 | 12/3/20 | Members reviewed + approve |
| AGIL\_05 | Robot must roll 270 degrees at a 25 speed for 2 sec. and Stop |  | 1 | 12/3/20 | Members reviewed + approve |
| AGIL\_06 | Robot must roll 180 degrees at a 49 speed for 3 sec. |  | 1 | 12/3/20 | Members reviewed + approve |
| AGIL\_07 | Robot must roll 225 degrees at 25 speed for 3 sec. |  | 1 | 12/3/20 | Members reviewed + approve |
| AGIL\_08 | Robot must roll 135 degrees at 25 speed for 3 sec. |  | 1 | 12/3/20 | Members reviewed + approve |
| AGIL\_09 | Robot must roll 225 degrees at 25 speed for 3 sec. and stop |  | 1 | 12/3/20 | Members reviewed + approve |
| AGIL\_10 | Robot must roll 90 degrees at 25 speed for 3 sec. and stop |  | 1 | 12/3/20 | Members reviewed + approve |
| AGIL\_11 | Robot must roll 0 degrees at 100 speed for 8 sec. and stop |  | 1 | 12/3/20 | Members reviewed + approve |
| AGIL\_12 | Program ends | Robot should be at original starting point | 2 | 12/3/20 | Members reviewed + approve |

## Security

### Protection

There should be a designated pathway in-order for the robot to pass around the objects. No physical object should keep the robot from executing functional requirements. Robot should be closely observed during its run to affirm the robot does not collide or get stopped by any object. User should also keep algorithm under their account in a private folder to make certain there is no threat to the robot. This makes sure outside sources cannot have access to the robot’s program.

### Authorization and Authentication

Team members will use Github.com in-order to access all separate components of the project individually. Here, all information will be easily accessible and will allow each team member to commit their work. One team member will create a repository to allow certain people access to the project’s documentation.

## Portability

In this project downloading Shero.edu app is required. Block code language is also the most recommended for an orderly algorithm. In addition, a spacious environment is suggested as the sprint is based on completing an obstacle course. Robot must also be cared for and not be damaged in any way, as this might affect its performance.

# Requirements Confirmation/Stakeholder sign-off

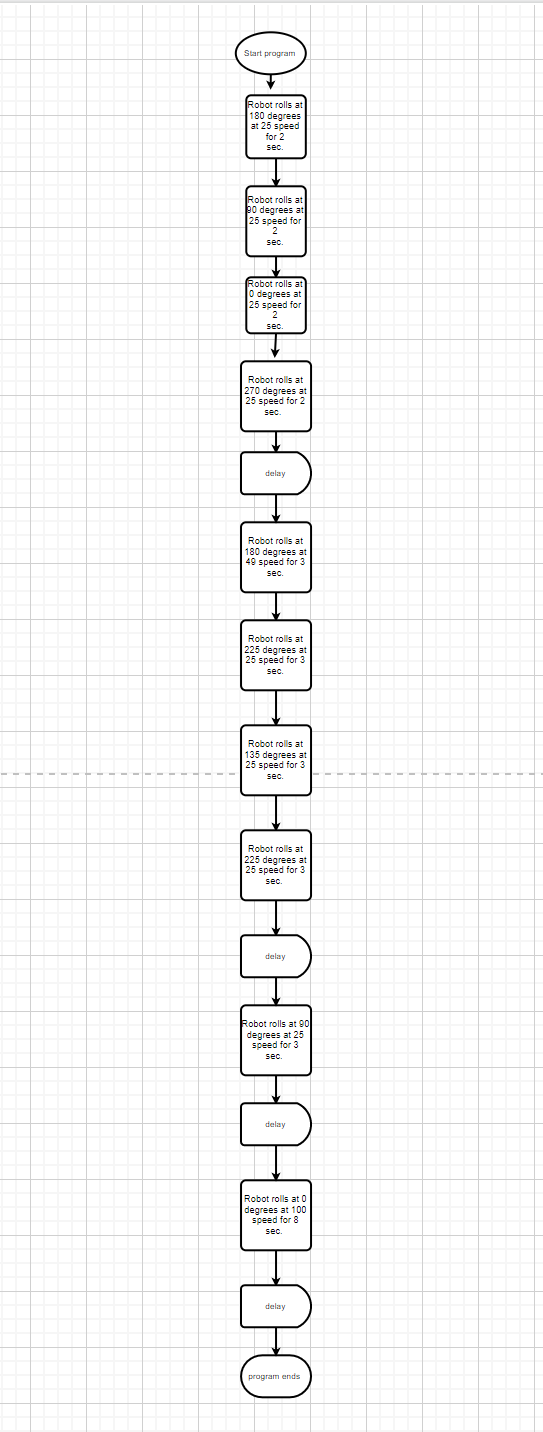
|  |  |  |
| --- | --- | --- |
| Meeting Date | Attendees (name and role) | Comments |
| 12/03/20 | Griffin - create + test algorithms | confirmed all functional requirements |
| 12/03/20 | Jenni – complete SDD + flowchart | Confirmed all functional requirements |
| 12/03/20 | Angaya – Flowchart | Confirmed all functional requirements |

# System Design

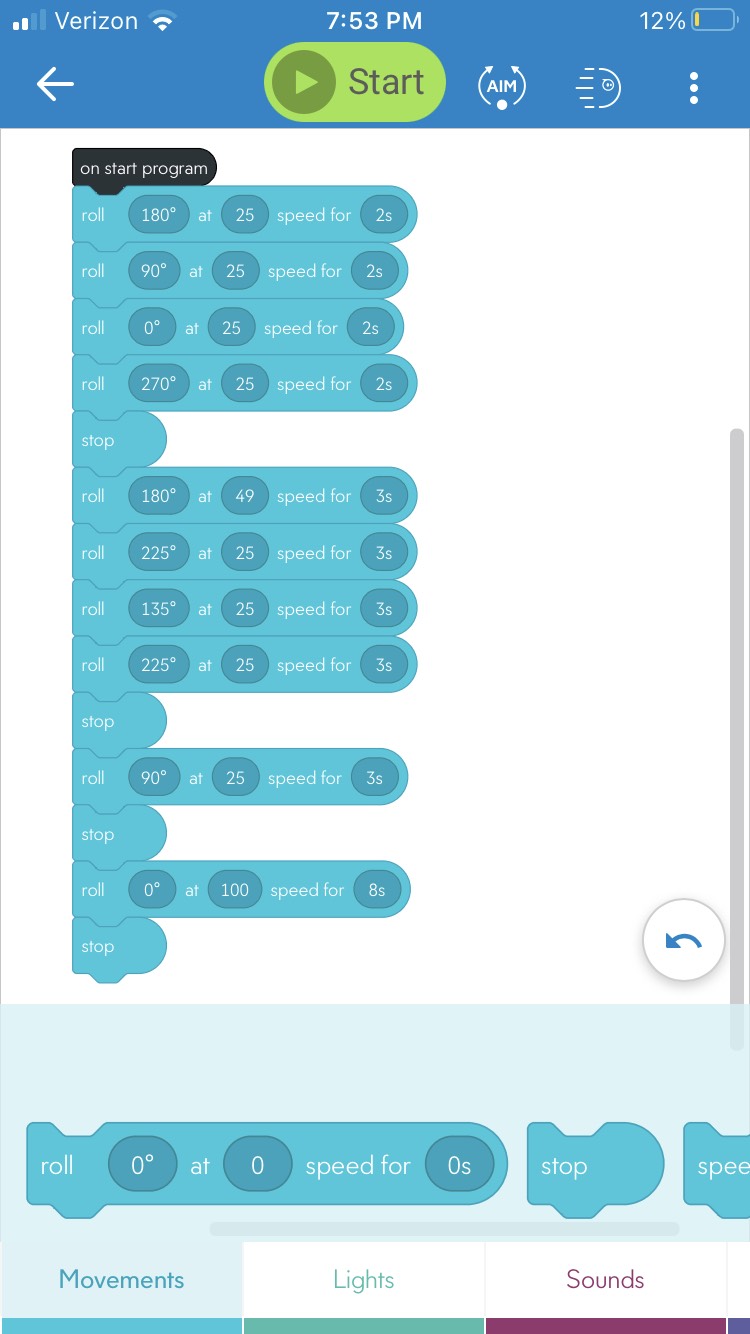
## Algorithm

|  |
| --- |
| 1. On start program |
| 1. Robot rolls at 180 degrees at 25 speed for 2 sec. |
| 1. Robot rolls at 90 degrees at 25 speed for 2 sec. |
| 1. Robot rolls at 0 degrees at 25 speed for 2 sec. |
| 1. Robot rolls at 270 degrees at 25 speed for 2 sec. and stop |
| 1. Robot rolls at 180 degrees at 49 speed for 3 sec. |
| 1. Robot rolls at 225 degrees at 25 speed for 3 sec. |
| 1. Robot rolls at 135 degrees at 25 speed for 3 sec. |
| 1. Robot rolls at 225 degrees at 25 speed for 3 sec. and stop |
| 1. Robot rolls at 90 degrees at 25 speed for 3 sec. and stop |
| 1. Robot rolls at 0 degrees at 100 speed for 8 sec. stop |
| 1. Program ends |

## System Flow

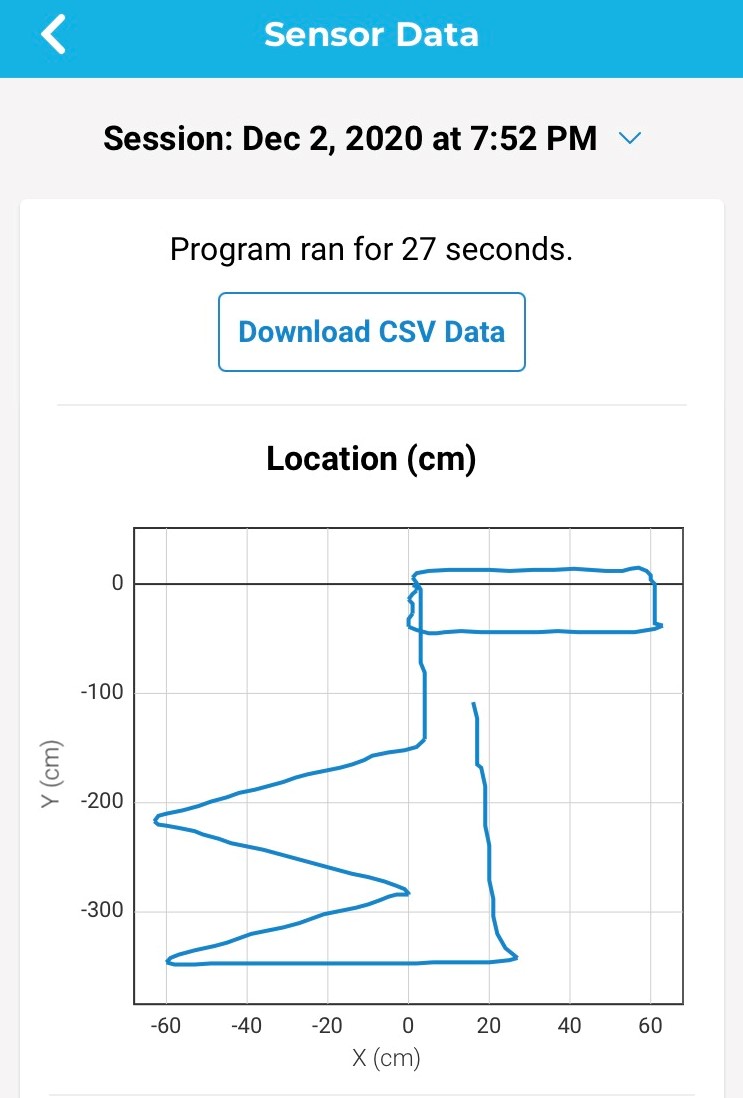


**Block Code below:**



**Flowchart here:**

**Sensor Data Below:**



## Software

The team used block code program on the Sphero.edu app. Through the app, we are able to control the robot’s travel of speed, distance, and time.

## Hardware

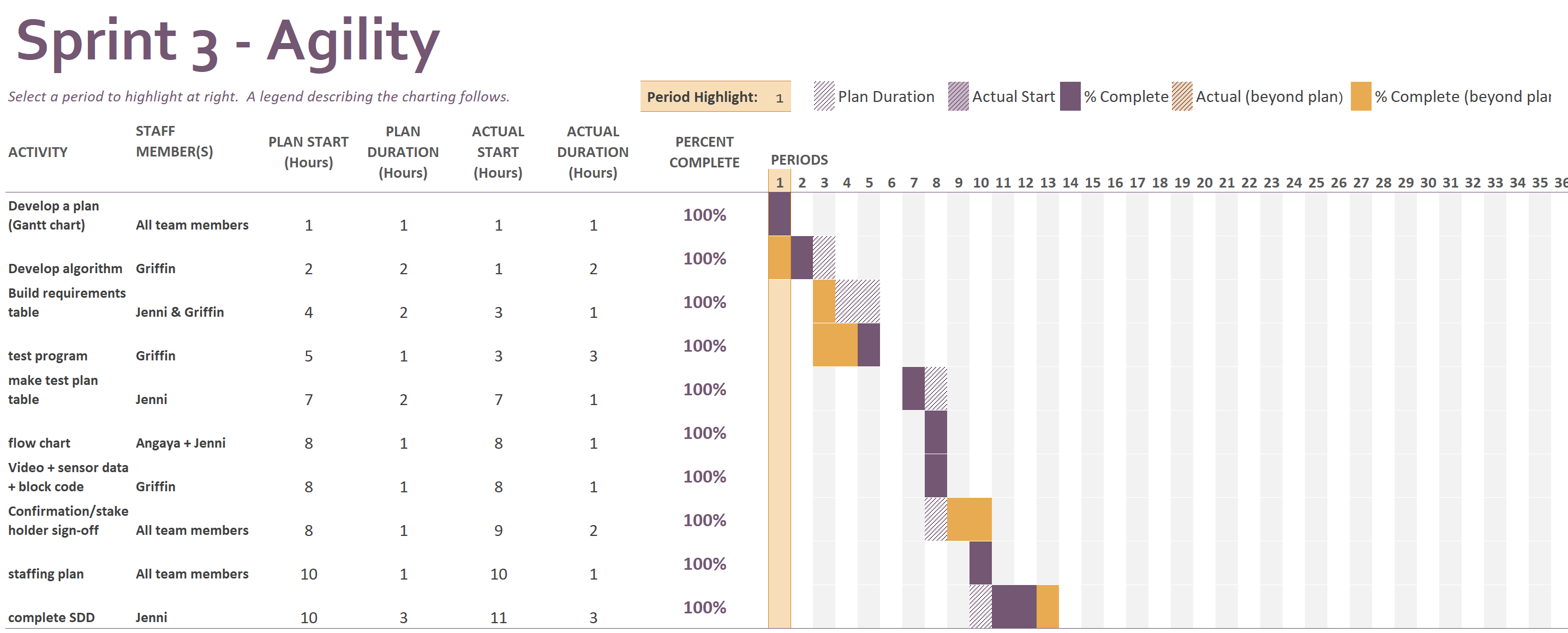
The team uses a robot that connects to the Sphero.edu app through Bluetooth. The user also uses his/her phone to create/adjust the program easily, always controlling the robot’s performance.

## Test Plan

| **Reason for Test Case** | **Test Date** | **Expected Output** | **Observed Output** | **Staff Name** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| To observe if robot starts at the designated square as its starting point | 12/3/20 | Robot will start in designated square | Robot starts in the designated square | Griffin | Pass |
| To observe if robot travels the appropriate speed and distance around the first object | 12/3/20 | Robot will travel the appropriate speed and distance around the first object | Robot traveled the appropriate speed and distance around the first object | Griffin | Pass |
| To observe if robot travels the appropriate speed and distance around the second object | 12/3/20 | Robot will travel the appropriate speed and distance around the second object | Robot traveled the appropriate speed and distance around the second object | Griffin | Pass |
| To observe if robot travels the appropriate speed and distance around the third object | 12/3/20 | Robot will travel the appropriate speed and distance around the third object | Robot traveled the appropriate speed and distance around the third object | Griffin | Pass |
| To observe if robot travels the appropriate speed and distance, knocking down three pins | 12/3/20 | Robot will travel the appropriate speed and distance, knocking down three pins | Robot traveled the appropriate speed and distance, knocking down three pins | Griffin | Pass |
| To observe if the program ends when it is supposed to | 12/3/20 | Program will end | Program ended | Griffin | Pass |

## Task List/Gantt Chart

**Gantt chart Below:**



## Staffing Plan

| Name | Role | Responsibility | Reports To |
| --- | --- | --- | --- |
| Jenni | System Design Document + Flowchart | Fill out documentation in the SDD + flowchart | GitHub repository called ‘Agility’ |
| Griffin | Create + Test Algorithms | Create algorithms + run test sprints | GitHub repository called ‘Agility’ |
| Angaya | Flowchart | Create a flowchart of the algorithm | GitHub repository called ‘Agility’ |