Sprint 3 - Agility Design Document
Due Date: December 4, 2023

Table of Contents

1. Executive Summary

1.1: Project Overview

The objective of this project is to program a robot to navigate a predefined figure-eight course successfully. The robot must stay within the provided path, start and finish within designated squares, and upon completion, speak the phrase "I am the winner" while flashing multicolored lights for 5 seconds.

1.2: Purpose and Scope of this Specification

This specification document outlines the functional and user experience requirements for the Sphero Edu platform. It serves as a guide for the development team to ensure that the software meets the educational objectives and technical standards set forth by the project stakeholders.

Intended Audience:

- Product Developers and Engineers
- User Experience Designers
- Quality Experience Designers
- Quality Assurance Teams
- Educational Content Creators
- Project Managers and Stakeholders

In Scope:

This document covers the following aspects of the Sphero Edu platform.

- User interface design and navigation
- Programming environment features
- Integration with Sphero Robots (BOLT, SPRK+, etc.)
- Access to educational resources and curriculum
- User authentication and account management
- Basic troubleshooting and support features

Out of Scope:

The following items are not within the scope of this document:

- Hardware specifications and manufacturing guidelines for Sphero Robots
- Firmware development for Sphero Robots
- Network infrastructure and hosting considerations
- Marketing and sales strategies for Sphero Edu
- Legal and compliance documentation

2. Product/Service Description

These general factors provide the foundation for the requirements outlined in subsequent sections. They shape the overall design and functionality of Sphero Edu to meet the educational goals and user needs effectively.

1. Educational Objectives: The primary goal of Sphero Edu is to provide an engaging and effective platform for teaching programming and robotics concepts to students of varying age groups and levels of expertise. The product aims to facilitate hands-on learning experiences that align with educational standards and curriculum requirements.

2. **User Proficiency Levels:** Sphero Edu carters to a diverse user base, including students, faculty, staff, and enthusiasts. These users may have varying experience and technical expertise in programming and robotics. The platform must be designed to accommodate beginners and more advanced users, offering a range of learning materials and challenges.

3. **Accessibility and Inclusivity:** Sphero Edu must be accessible to a broad audience, including individuals with diverse abilities and learning styles. The platform should consider accessibility features, ensuring all users can engage with the product effectively.

4. **Integration with Sphero Hardware:** Sphero Edu is designed to work in conjunction with Sphero robots, such as BOLT and SPRK+. The product's requirements must align with the capabilities and functionalities of these robots, ensuring seamless interaction and program execution.

5. **Learning Outcomes and Assessments:** The platform should support the assessment of student progress and learning outcomes. This process will include the ability to track and evaluate individual and group performance and provide educators with insights into engagement and comprehension.

6. **Platform Stability and Reliability:** To foster a positive learning experience, Sphero must be stable and reliable. This includes minimizing system crashes, ensuring smooth program execution, and providing robust error-handling mechanisms.

7. **Scalability and Growth:** The platform should be designed with scalability in mind to accommodate potential growth in user base and content offerings. This ensures that Sphero Edu remains a viable education tool in the long term.

8. **Compliance and Security:** Sphero Edu should adhere to relevant data protection and privacy regulations. This includes safeguarding user data, providing secure authentication methods, and ensuring compliance with educational privacy standards.

9. **Technological Ecosystem:** Consideration should be given to the broader technological ecosystem within which Sphero Edu operates. This includes compatibility with operating systems, browsers, and network environments commonly used in educational settings.

2.1: Product Context

Sherp Edu is a product within the broader ecosystem of Sphero robotics and education offerings. It is designed to complement and enhance the functionality of Sphero's range of programmable robotics, including models like Sphero BOLT, SPRK+, and others.
Below is a breakdown of how Sphero Edu fits into the larger systems:

1. **Robot Hardware:** Sphero Edu is designed to work with Sphero's various programmable robots. These robots serve as the physical entities that students interact with, and they can execute the programs created in the Sphero Edu platform.

2. **Sphero Edu Platform:** This software component provides the programming interface. It allows users to create, edit, and execute code for the Sphero robots. The platform also includes a range of educational resources, activities, and challenges to support learning.

3. **Educational Curriculum:** Sphero Edu often comes with educational materials and a curriculum designed to teach programming, robotics, and STEM concepts. This may include lesson plans, tutorials, and sample projects.

4. **Mobile Devices or Computers:** The Sphero Edu platform typically runs on mobile devices and computers. These devices are used to access the programming environment and to interface with the robots.

5. **Internet Connectivity:** While only sometimes necessary for basic programming, an internet connection might be required to access additional resources and updates or to collaborate with other users.

6. **External Interfaces:** Depending on the specific use case, Sphero Edu may have interfaces with other educational tools, platforms, or hardware components. For example, it might integrate with learning management systems or other educational software.

2.2: User Characteristics

Understanding the diverse range of users is crucial for maximizing the potential of Sphero Edu. From students exploring the wonders of programming to seasoned faculty members seeking innovative teaching tools, each user brings unique characteristics and expertise to the table. Let's delve into the distinct profiles of students, faculty, staff, and other enthusiasts who will engage with this educational platform. By recognizing their backgrounds, experience levels, and technical aptitude, we can tailor the Sphero Edu experience to meet the needs and aspirations of every user.

1. Student:

- **Role:** Student
- **Experience:** Limited to moderate experience in programming and robotics, likely as part of their curriculum or extracurricular activities.
- **Technical Expertise:** Basic to moderate proficiency in using devices or computers. It may require some guidance in setting up and using the platform.
- **Other Characteristics:** Curios, eager to learn, may have varying interests in STEM subjects.

2. Faculty Member:

- **Role:** Educator/Faculty Member
- **Experience:** Experienced in teaching, with a background in STEM education. They may have used educational technology or robotics platforms in the past.
- **Technical Expertise:** Proficient in using educational software and technology. Can quickly adapt to new platforms and integrate them into teaching methods.
- **Other Characteristics:** Goal-oriented, interested in enhancing teaching methods, may be responsible for creating lesson plans and educational content.

3. Staff Member:

- **Role:** Administrative/Staff Member
- **Experience:** Limited experience in programming and robotics unless specifically related to their job function or interests.
- **Technical Expertise:** Basic proficiency in using office software and standard digit tools. It may require some support in setting up and troubleshooting.
- **Other Characteristics:** Task-oriented, may use the product for specific educational events or programs, interested in supporting educational initiatives.

4. Other Enthusiasts:

- **Role:** Enthusiasts or Hobbyists
- **Experience:** Varied, from novice to advanced, depending on personal interests and hobbies. May have a keen interest in robotics or programming.
- **Technical Expertise:** This can range from basic to advanced, depending on the individual's background and involvement in related activities.
- **Other Characteristics:** Highly motivated by personal interest, may use the product for individual projects, experimentation, or recreational purposes.

2.3: Assumptions

These assumptions are critical in ensuring that the Sphero Edu platform functions as intended and that users can effectively utilize it for educational purposes. If any of these assumptions do not hold, adjustments to the requirements or implementation approach may be necessary.

1. **Device Compatibility:** It is assumed that users can access compatible devices such as tablets, smartphones, or computers with supported operating systems.

2. **Internet Access:** If the Sphero Edu platform relies on online resources, users are assumed to have reliable internet access.

3. **Sphero Robot Availability:** Users are expected to have access to the Sphero robots, including models like BOLT, SPRK+, or others. The platform's functionality is contingent on these physical robots.

4. **Basic User Proficiency:** Users are assumed to have a basic understanding of operating mobile devices or computers, including tasks like app installation and primary software navigation.

5. **Educator's Familiarity with Sphero Edu:** For educators using Sphero Edu in a teaching context, it is assumed that they are familiar with the platforms, their features, and how to integrate them into their lesions.

6. **Access to Curriculum Materials:** If Sphero Edu includes educational curriculum materials, users are assumed to have access to these resources.

7. **Robustness of the Internet Connection:** If the platform requires continuous online access, users are assumed to have a stable and reliable Internet connection.

8. **Availability of Software Updates:** It is assumed that users can update their Sphero Edu platform and any associated apps to the latest versions for optimal performance.

9. **Compliance with Age Restrictions:** If there are age restrictions or guidelines for using the product, users and educators are assumed to adhere to these guidelines.

10. **Availability of Support and Documentation:** Users may assume that there is access to adequate support channels (such as user forums, documentation, or customer service) to address any issues or queries.

2.4: Constraints

These constraints guide the design process of Sphero Edu, ensuring that the platform meets technical and educational requirements while operating within defined limitations and standards.

1. **Parallel Operations with Legacy Systems:** Sphero Edu may need to operate in parallel with existing legacy systems in specific educational environments. This requires compatibility and integration capabilities with older hardware or software configurations

2. **Audit Functions:** The platform must incorporate robust audit functions, including an audit trail and log files, to track user interactions, system activities, and changes to programs or educational resources. This is essential for accountability, troubleshooting, and compliance with academic standards.

3. **Access, Management, and Security:** Access control mechanisms must be implemented to regulate user permissions and ensure that only authorized individuals can modify content or access certain features. Robust authentication methods, data encryption, and secure communication protocols are imperative to protect user data privacy.

4. **Criticality of the Application:** Since Sphero Edu is an educational tool, reliability and uptime are critical. The platform should be designed to minimize downtime and ensure uninterrupted access for students and educators.

5. **System Resources Constraints:** The platform must be optimized to operate within defined resource constraints, such as limits on disk space, memory, and processing power. Efficient resource management guarantees smooth performance across various devices and environments.

6. **Compatibility with Educational Standards and Frameworks:** Sphero Edu must adhere to educational standards in terms of content and technical specifications. This may include compliance with specific programming languages, frameworks, or academic curriculum guidelines mandated by educational authorities.

7. **Regulatory and Compliance Requirements:** The platform must comply with relevant data protection and privacy regulations, including but not limited to GDPR, COPPA, and other local or international laws. This includes providing features for data access requests, deletion, and consent management.

8. **Local and Internationalization:** The platform may need to support multiple languages and cultural contexts to cater to a diverse user base across different regions. This includes considerations for date formats, language translation, and localized content.

9. **Scalability and Performance Optimization:** Sphero Edu should be designed to scale efficiently as the user base grows. This requires a scalable architecture that handles increased traffic, user accounts, and educational content without compromising performance.

2.5: Dependencies

These dependencies are crucial considerations that will shape the requirements for the Sphero Edu platform, ensuring that it meets both technical and educational objectives effectively.

1. **Sphero Robot Compatibility:** The development of Sphero Edu is dependent on ensuring compatibility with specific models of Sphero robots, such as BOLT and SPRK+. The platform's functionality and features will be closely tied to the capabilities of these robots.

2. **Programming Language and Frameworks:** The selection of programming languages and frameworks may impact development. For example, if the platform is built using a specific framework, the requirements will need to align with the capabilities and limitations of that framework.

3. **Integration with Educational Standards:** The platform's requirements will be influenced by the need to align with educational standards and curriculum frameworks. This may include the incorporation of specific programming concepts or instructional methodologies.

4. **Data Access and Integration:** If the platform requires access to external data sources or APIs, there may be dependencies on data availability, retrieval frequency, and integration methods. For example, regular data updates will be necessary if the platform fetches educational resources from an external database.

5. **Operating System Compatibility:** Dependencies on specific operating systems (e.g., iOS, Android, and Windows) may affect the platform requirements. For instance, if the product aims to run on iOS and Android devices, the conditions must account for each platform's unique features and constraints.

6. **Security and Compliance Requirements:** Compliance with data protection and privacy regulations will be a critical dependency. The platform's requirements must align with the necessary security measures and data handling practices to ensure legal and ethical compliance.

7. **Educational Content Development:** The availability and development of educational content, such as lesson plans, tutorials, and challenges, will influence that platform's requirements. For example, if specific content types or formats are planned, the platform must support their integration and display.

8. **User Authentication and Account Management:** The requirements for user authentication and account management will be dependent on the chosen authentication methods, such as email/password, social login, or single sign-on (SSO).

9. **Hardware and Network Infrastructure:** The platform's requirements may be affected by the hardware capabilities of the devices it will run on and network constraints (e.g., Wi-Fi or cellular connectivity). For example, offline features may not be prioritized if the platform requires a stable internet connection.

10. **Regulatory Approvals and Certifications:** If the product requires specific certifications or approvals (e.g., educational standards compliance, safety certifications), the requirements must align with the criteria set forth by the relevant authorities.

3. Requirements

Each requirement is clearly defined and prioritized and includes a brief description. This structure allows for easy reference, verification, and modification while ensuring all requirements are correct, complete, and verifiable.

Priority 1 - Must have

1.1. The platform must support compatibility with Sphero robots, including BOLT and SPRK+ models and future releases.

1.2. Users must be able to create, edit, and save programs within the platform.

1.3. The platform must provide a secure user authentication system, allowing users to create accounts and log in securely.

Priority 2 - Immediate Benefits

2.1. The platform should offer an intuitive user interface, ensuring easy navigation for users of all proficiency levels.

2.2. Users must be able to access a library of educational resources, including tutorials, lesson plans, and challenges.

2.3. The platform should support real-time feedback and execution of programs on connected Sphero robots.

Priority 3 - Nice to Have

3.1. The platform could include a social component, allowing users to share programs—challenges, and achievements with a community of educators and learners.

3.2. It would be beneficial if the platform provided progress tracking and reporting features, allowing educators to monitor student performance.

3.3. The platform might include a simulation mode, enabling users to test programs without requiring a physical Sphero robot.

Input:

- User login credentials (username and password)
- Program code and instructions from users
- Educational content selection by users (e.g., lesson plans, challenges)

Output:

- Program execution on connected Sphero robots
- User progress reports and achievement notifications
- Error messages or notifications for incorrect program syntax.

Functions:

- User authentication and account creation
- Program creation, editing, and saving
- Access to educational resources and content library
- Real-time program execution on Sphero robots
- Program tracking and reporting

Locations:
- Mobile devices (iOS, Android)
- Desktop/laptop computers (Windows, macOS)
- Sphero robots (BOLT, SPRK+)

Users:
- Students
- Educators/Faculty
- Staff
- Enthusiasts/Hobbyists

3.1: Functional Requirements

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
1	Meet as group	Easily done	high	10/28	Colin G
2	Go past first bottle without hitting it and stop	A little trial and error	low	10/28	Colin G
3	Change direction and go around the second bottle and stop	A little trial and error	low	10/28	Colin G
4	Change direction and go around 3rd bottle and stop	A little trial and error	low	10/28	Colin G
5	Change direction and go over the binder. End at the corner and stop	The hardest part, definitely took a few tries	high	10/28	Colin G
6	Change direction and end in the final square	Easy as long as the other things went smoothly	low	10/28	Colin G

3.2 Security

3.2.1: Protection

By incorporating these factors into the design and implementation of the Sphero Edu platform, the system will be well-equipped to protect against both malicious and accidental security threats.

Encryption: Implement encryption protocols to secure data transmission between the platform and external servers, ensuring that sensitive information remains confidential and protected from interception.

Activity Logging and Auditing: Maintain comprehensive logs of user activities, including login attempts, program modifications, and other critical operations. This enables administrators to track and investigate any suspicious or unauthorized activities.

Data Integrity Checks: Implement data integrity checks (e.g., checksums, digital signatures) to verify that data has not been tampered with during transit or storage, thereby ensuring the authenticity and integrity of information.

Firewalls and Intrusion Detection/Prevention Systems: Employ firewalls to filter incoming and outgoing traffic and utilize intrusion detection/prevention systems to monitor and respond to suspicious network activities in real time.

Security Training and Awareness: Provide training and awareness programs for users, educating them on best practices for maintaining system security and avoiding common security pitfalls.

3.2.2: Authorization and Authorization:

By incorporating these authentication and authorization factors into the design and implementation of the Sphero Edu platform, the system will establish a strong foundation for secure user interactions and access control.

Authentication Factors:

Username and Password: Users must create usernames and strong passwords. Password policies should include requirements for length, complexity, and regular expiration.

Multi-Factor Authentication: Implement MFA to add a layer of security. This may include using SMS codes, authentication apps, or hardware tokens as a second form of authentication.

Biometric Authentication: Optionally, support biometric authentication methods (e.g., fingerprint, facial recognition) for users who have compatible devices, providing a convenient and secure login option.

Account Lockout Policy: Implement a policy that locks user accounts after a specified number of failed login attempts to mitigate brute-force attacks.

Authorization Factors:

Role-Based Access Control: Utilize RBAC to assign specific roles (e.g., student, teacher, administrator) to users. Each position will have defined permissions, limiting their access to certain features and data.

Resource-Based Access Control: Implement access control at the resource level, ensuring that users can only interact with or modify resources (e.g., programs, educational content) for which they have appropriate permissions.

Permission Granularity: Fine-tune permissions to ensure that users can only access the specific functionalities and data relevant to their roles and responsibilities within the platform.

Owner-Based Authorization: Define ownership rules for resources (e.g., programs, projects) and grant specific permissions to the owners, allowing them to manage and control their content.

3.3: Portability

Considering these attributes, the Sphero Edu platform can be designed and implemented with portability, ensuring it can quickly adapt to different host environments without sacrificing functionality or performance.

Percentage of Components with Host-Dependent Code: Identify the percentage of code components that depend on specific host machines or operating systems. Minimizing host-dependent code will enhance portability.

Percentage of Code that is Host Dependent: Quantify the portion of the codebase that relies on host-specific features or libraries. Strive to keep this percentage as low as possible to facilitate porting.

Use of a Proven Portable Language: Select a programming language known for its portability across different operating systems and hardware platforms. This ensures that the codebase can be easily adapted to various environments.

Use of a Particular Compiler or Language Subset: Ensure that the chosen compiler or language subset is compatible with a wide range of operating systems and platforms. Avoid relying on compiler-specific features or extensions.

Use of a Particular Operating System: Minimize dependencies on specific operating system functionalities. Provide fallback mechanisms or alternative implementations for non-supported environments if certain features are required.

Environment Independence: Design the system to be environment-independent, ensuring consistent behavior regardless of the operating system, network configurations, or development/production environments. This includes compatibility with different web browsers and mobile platforms.

4. Requirements Confirmation/Stakeholder sign-off

Meeting Date	Attendees (name and role)	Comments
10/28/23	Gavin K and Collin G	Completed three videos along with the corresponding block code and sensor data for each.

5. System Design

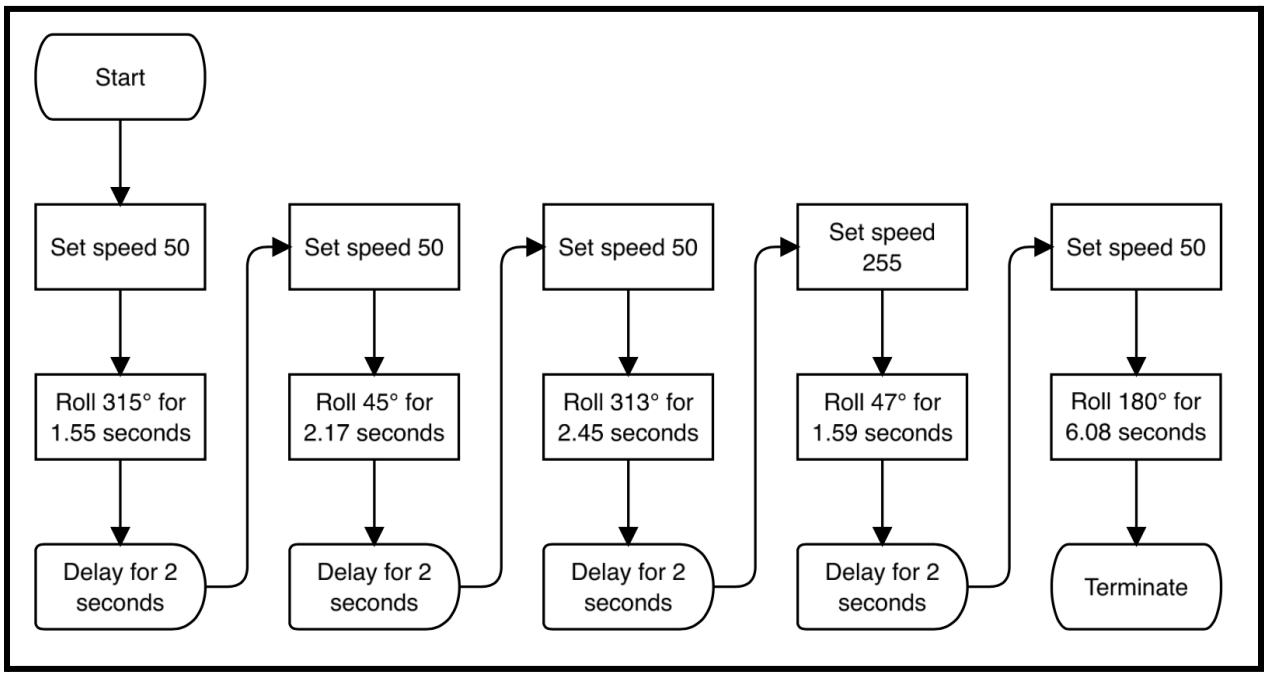
This section provides a detailed insight into the technical aspects of the Sphero Edu system. It covers architectural design, hardware and software requirements, staffing roles, coding standards, and testing strategies. The section emphasizes the importance of maintaining coding conventions, employing effective testing methodologies, and conducting thorough code reviews for quality assurance. Additionally, it highlights the significance of comprehensive technical documentation and knowledge transfer plans to facilitate seamless development and maintenance of the platform. This section serves as a complete guide for the technical foundation of the Sphero Edu system.

5.1: Algorithm

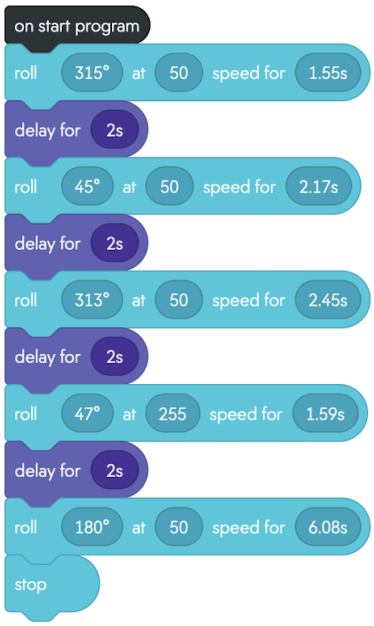
Step I: Charge the SPRK Robot until it's able to be used for test runs
Step II: Align the SPRK Robot up with the tape of the desired outline
Step III: Adjust alignment so that SPRK will go straight when the code starts
Step IV: Have the SPRK Robot roll 315 degrees at 50 speed for 1.55 seconds
Step V: Delay for 2 seconds
Step VI: roll 45 degrees at 50 speed for 2.17 seconds
Step VII: delay for 2 seconds
Step VIII: roll 313 degrees at 50 speed for 245 seconds
Step IX: delay for 2 seconds
Step X: roll 47 degrees at 255 speed for 1.59 seconds
Step XI: Delay for 2 seconds
Step XII: roll 180 degrees at 50 speed for 6.08 seconds
Step XIII: End the program

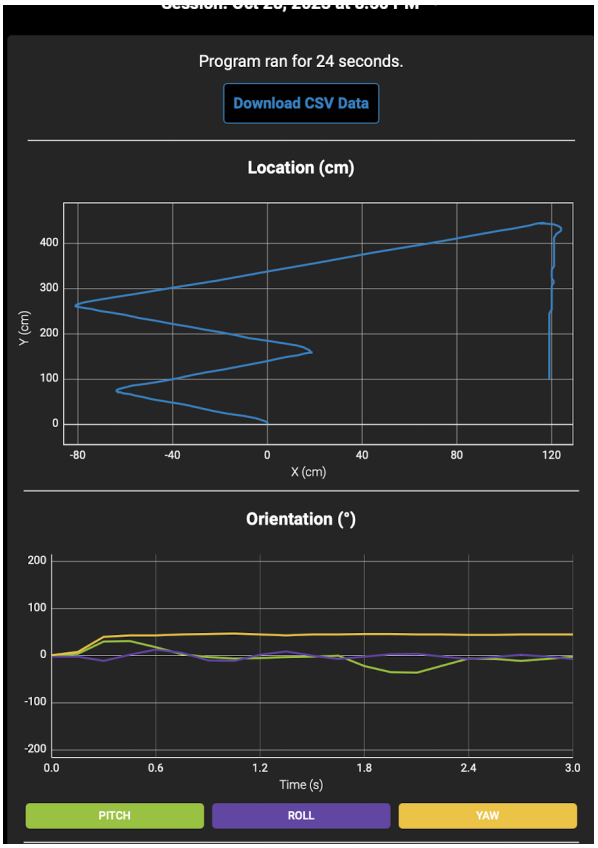
5.2: System Flow

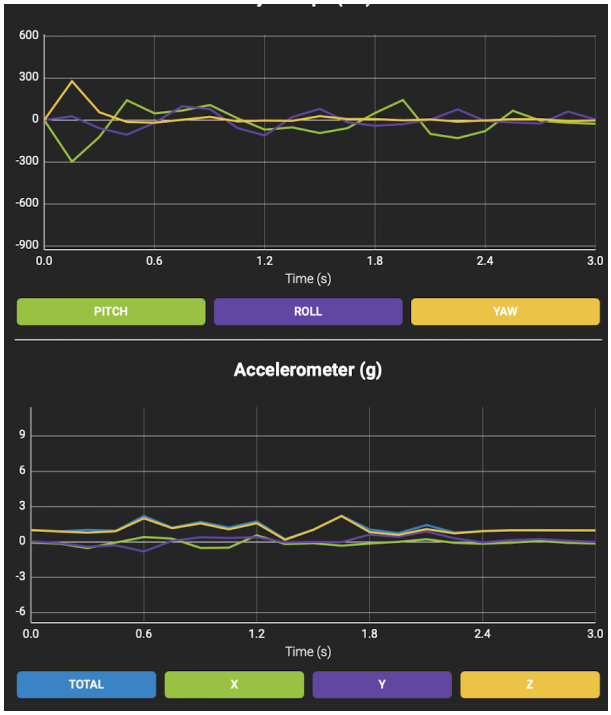
The depicted flowchart illustrates a sequence that comprises programmed movements involving rolling actions characterized by defined angles, speeds, and durations. These movements are interspersed with pauses, represented as delays, occurring between each specific action in the sequence.

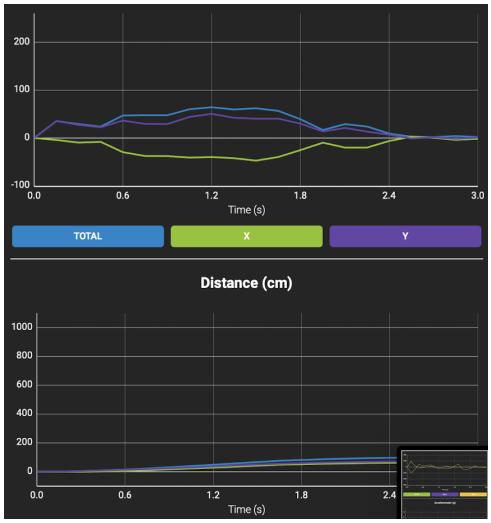


5.3: Software


on start program
roll 315° at 50 speed for 1.55s
delay for 2s
roll 45° at 50 speed for 2.17s
delay for 2s
roll 313° at 50 speed for 2.45s
delay for 2s
roll 47° at 255 speed for 1.59s
delay for 2s
roll 180° at 50 speed for 6.08s
stop


Program ran for 24 seconds.
Download CSV Data
Location (cm)
Y (cm)
X (cm)
Orientation (°)
Time (s)
PITCH
ROLL
YAW


Time (s)
PITCH
ROLL
YAW
Accelerometer (g)
Time (s)
TOTAL
X
Y
Z


Time (s)
TOTAL
X
Y
Distance (cm)
Time (s)

5.4: Hardware

The hardware devices played a crucial role in the application's development, testing, and demonstration phases. The SPRK+ robot was instrumental in testing and interacting with the application's robotic functionalities. The MacBook Pro served as the primary development environment, providing the necessary computing power and tools for creating the application. Additionally, the iPhone was explicitly utilized for video recording, allowing for comprehensive documentation and analysis of the application's performance. Each device contributed uniquely to the application's functionality and usability.

5.5: Test Plan

The following outlines the comprehensive test plan for Sprint 3 - Agility. It encloses a detailed listing of all unit tests conducted, providing essential information, including the testing explanation, testing date, assigned staff member, and the resulting pass/fail status.

Reason for Test Case	Test Date	Expected Output	Observed Output	Staff Name	Pass/Fail
Basis	10/28	13% completion avoid the first glass	Hit the glass	Gavin Seb Collin	Fail
First Glass Pass	10/28	13% completion avoid the first glass	Went around the first glass and successfully was able to calculate the distance from the second glass	Gavin Seb Collin	Pass
Two Glass Pass	10/28	Get past the two glasses	Went around the two glasses successfully	Gavin Seb Collin	Pass
Binder Ramp	10/28	Jump the binder successfully	Had the binder facing the wrong way, and the robot ran into Binder	Gavin Collin Seb	Fail
Binder Ramp 2	10/28	Jump the binder successfully	Went to the left of the binder and didn’t end up at the endpoint	Gavin Collin Seb	Fail
Binder Ramp 3	10/28	Jump the binder successfully	Went too far over the binder but went straight over	Gavin Seb Collin	Fail

Binder Ramp 4	10/28	Jump the binder successfully	It was able to successfully Jump the binder and end up in the proper spot	Gavin Seb Collin	Pass
Turn Right and Straight	10/28	Turn right and go straight	Turned right and went the distance	Gavin Seb Collin	Pass

5.6: Task List/Gantt Chart

5.7: Staffing Plan

Name	Role	Responsibility	Reports To
Sebastien Reed	Communicator	Communicate with the professor and group	Collin Gavin
Collin Gavin	Manager	Making sure the overall project runs on time	Professor
Gavin Kinsella	Assembly person	Makes sure all parts of the project fit together correctly	Collin Gavin