

The background of the slide is a dark blue color with a complex, glowing circuit board pattern. The pattern consists of numerous thin, light blue lines that form a dense network of interconnected paths, resembling a printed circuit board (PCB) layout. These lines are interspersed with small, circular nodes or vias, creating a sense of depth and technological complexity. The overall effect is a futuristic and high-tech aesthetic.

CS104 Robotics Project

Vincent, Mason, and Zak

Roles of Each Team Member

- All Members
 - SDD Logging
 - Gantt Chart
- Vincent
 - Block Code
 - created in Sphero Edu
 - Github Management
- Mason
 - Algorithm Development
- Zak
 - Flowchart Design
 - created in draw.io using algorithm as template



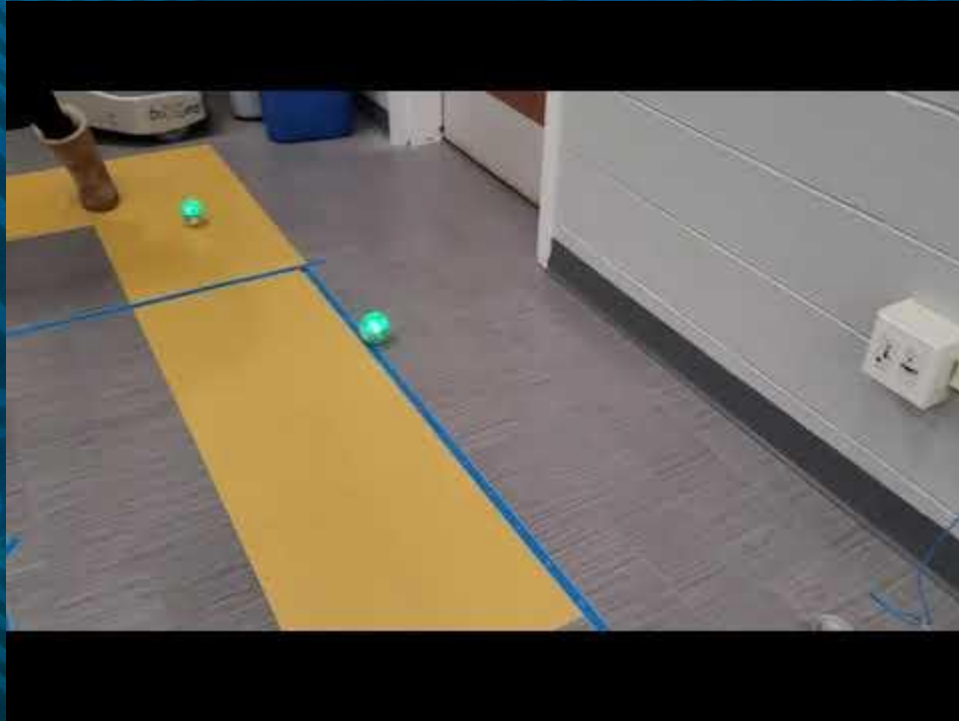
Algorithm Developments

- All steps that are used within the Algorithms are used to properly run the robot sprint.
 - Must be specific to be successful.
- They are the steps that the programmer must use in order for the program to run successfully.
- They are not only used for developing the program
 - ex) placing the robot and orienting it at the proper angle.

Sprint 1 - Endurance

- Most straightforward sprint
- Challenges
 - Getting the robot to stop at the correct position
 - Adjusting aim angle so robot does not go off track
- How we overcame the challenges
 - Testing constantly by adjusting specific values in the code

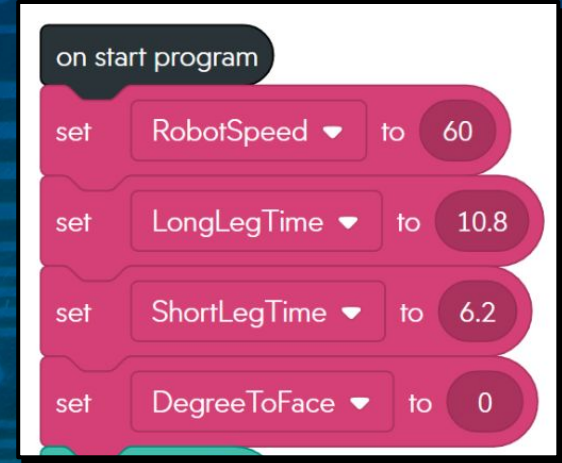
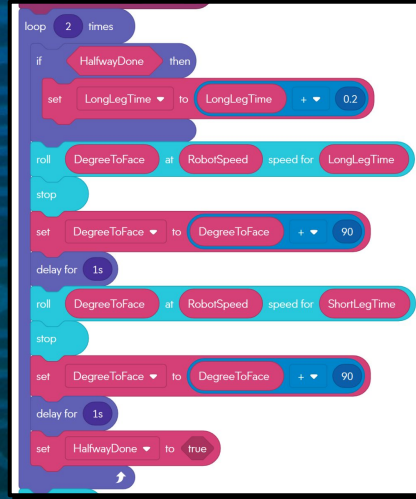
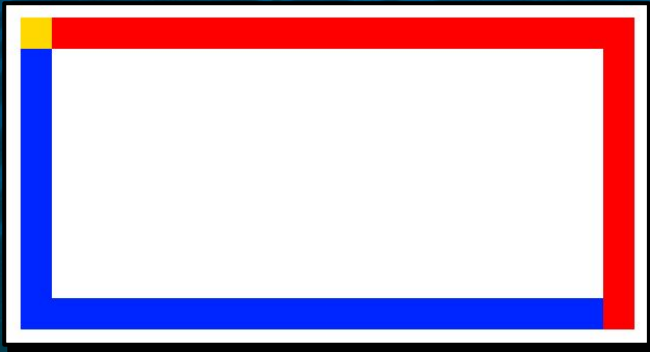
Sprint 1 - Endurance Test Run



<https://youtu.be/Gx9xvKXsH7Y>

Endurance Sprint Programming

- Use a programming approach to allow easy adjustment of variables at the beginning
- LOOPING
 - We know the course is a rectangle



Sprint 2 - Accuracy

- Most intimidating sprint
- How to approach a challenge:
 - Research yields results
 - Sphero website tutorial
 - Geometry ideas
- Was it accurate?
 - Course shape challenges



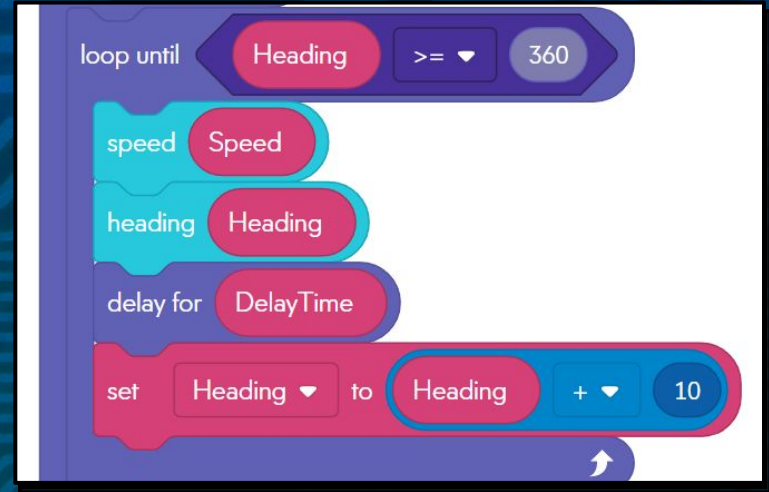
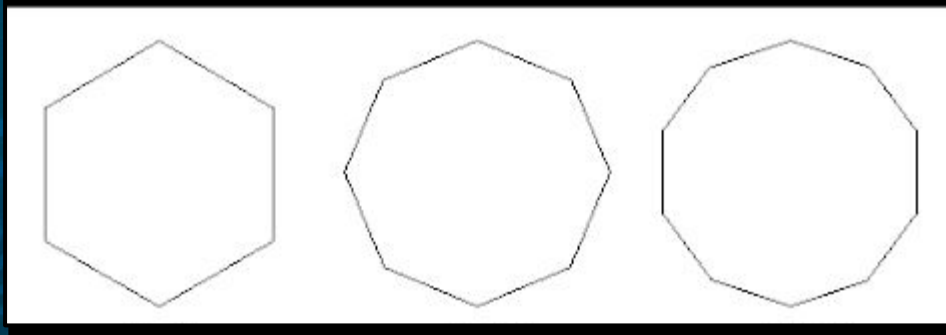
Sprint 2 - Accuracy Test Run



https://youtu.be/QCU5f_AwZuU

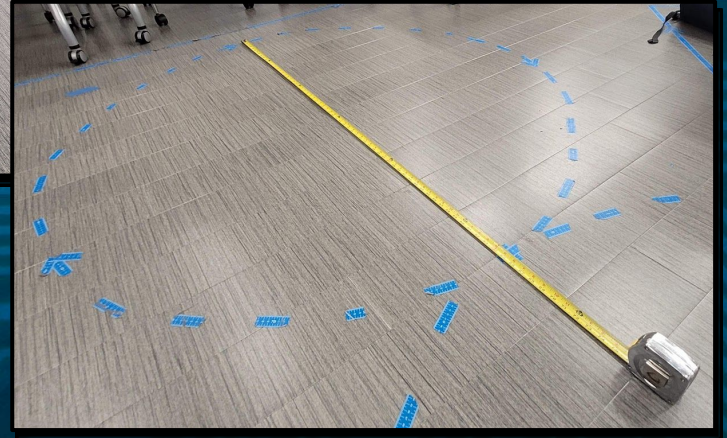
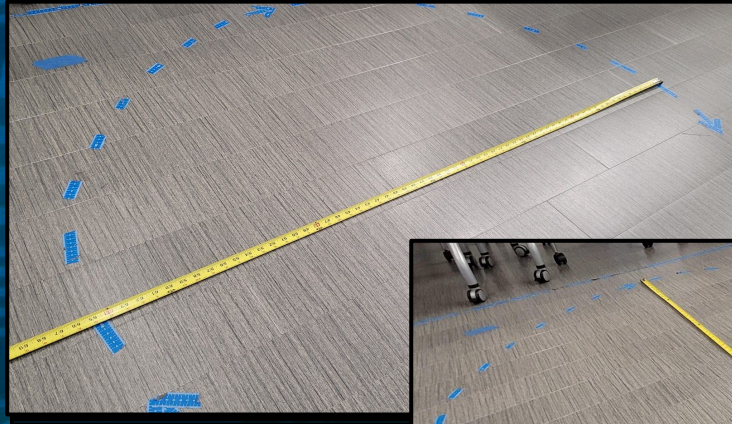
Accuracy Sprint Movement

- Move in a short line, then turn at a small angle repeatedly



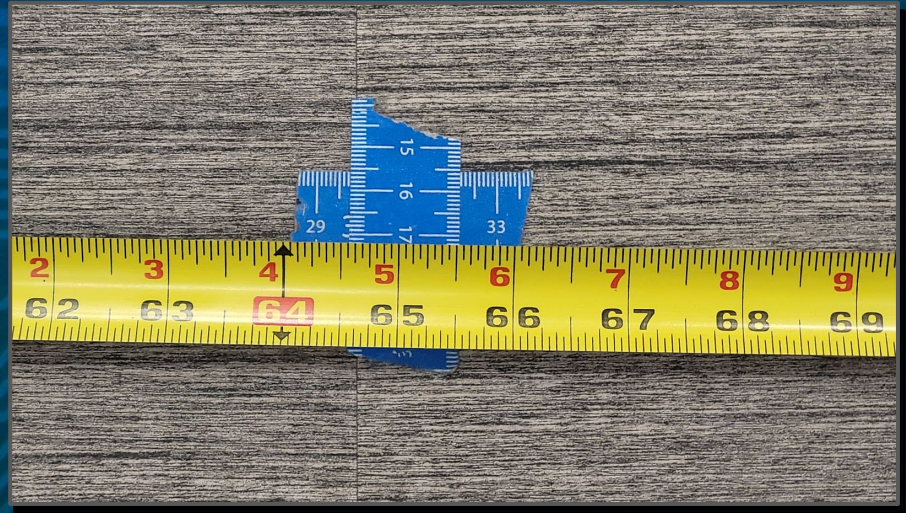
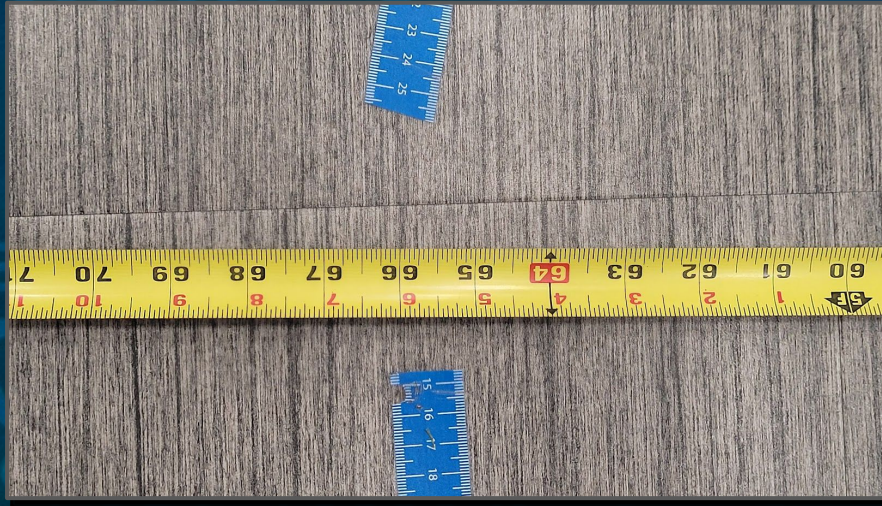
Accuracy Sprint Course Challenges

- The dimensions of the track are not perfect



Accuracy Sprint Course Challenges

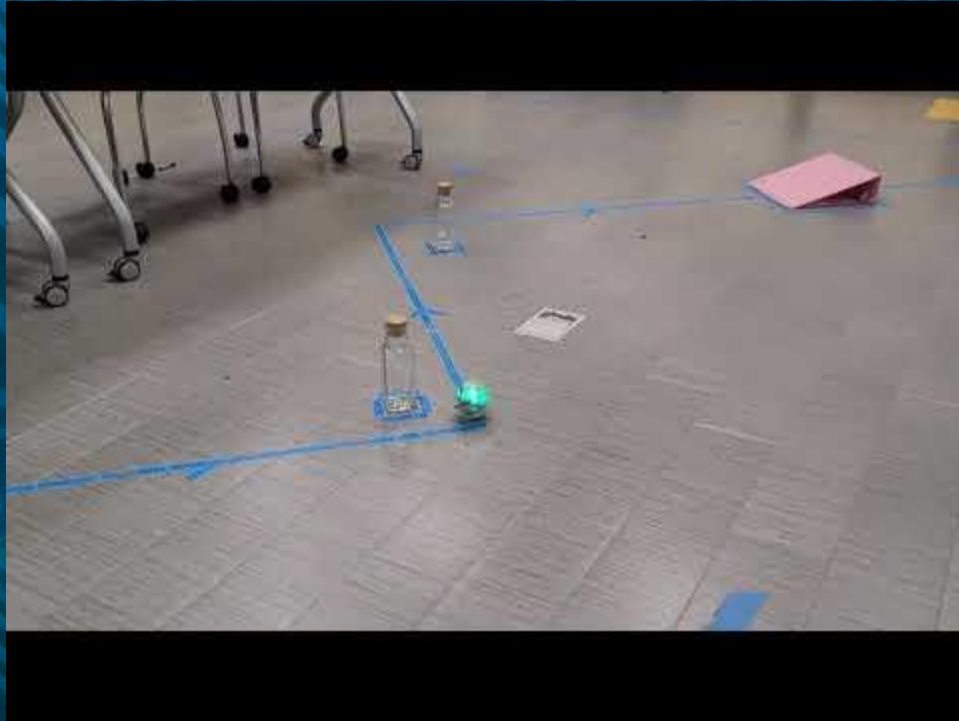
We program for perfect circles, when that is not the case in the real world



Sprint 3 - Agility

- Don't trust physics
 - Ramp landing will usually be different on every test run
- Timing is everything
 - Overshoot or undershoot = object collision
- Cheating on bowling
 - Ensure all pins are knocked over
 - [never said we couldn't]

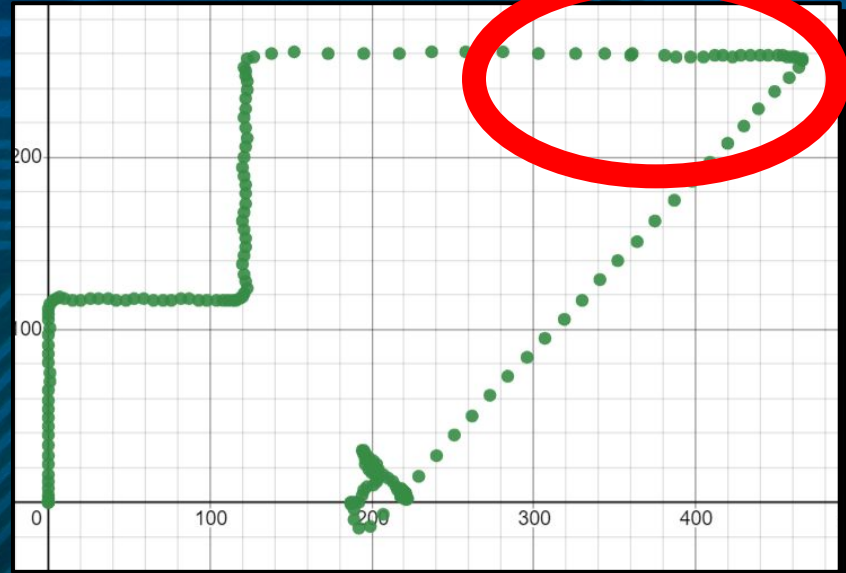
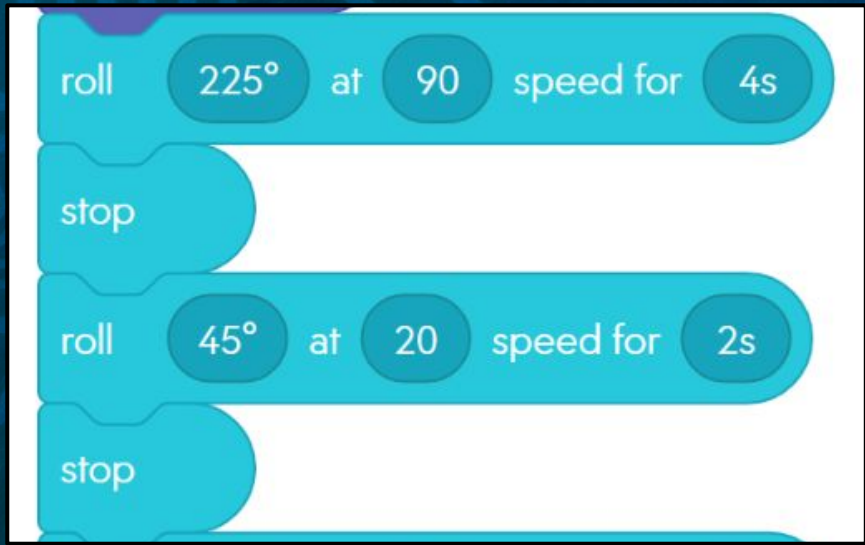
Sprint 3 - Agility Test Run



<https://youtu.be/LT2LWjVF-tk>

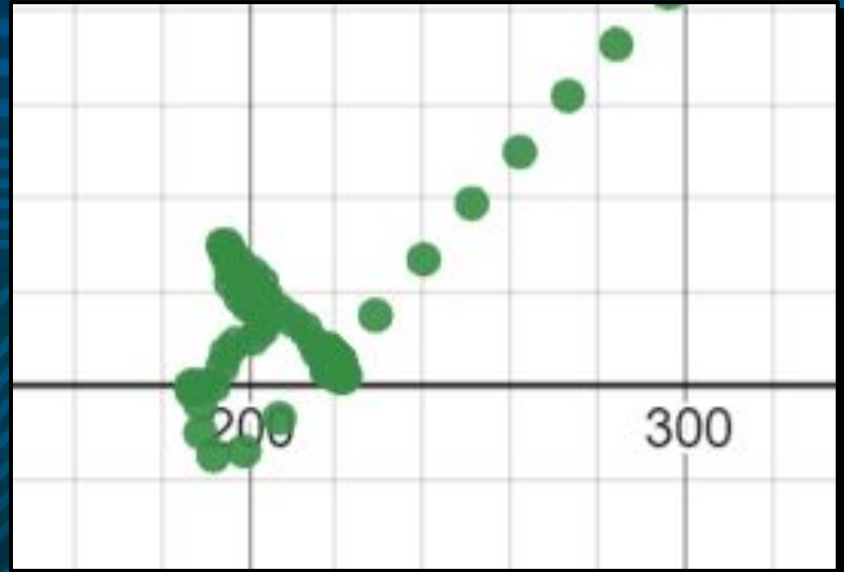
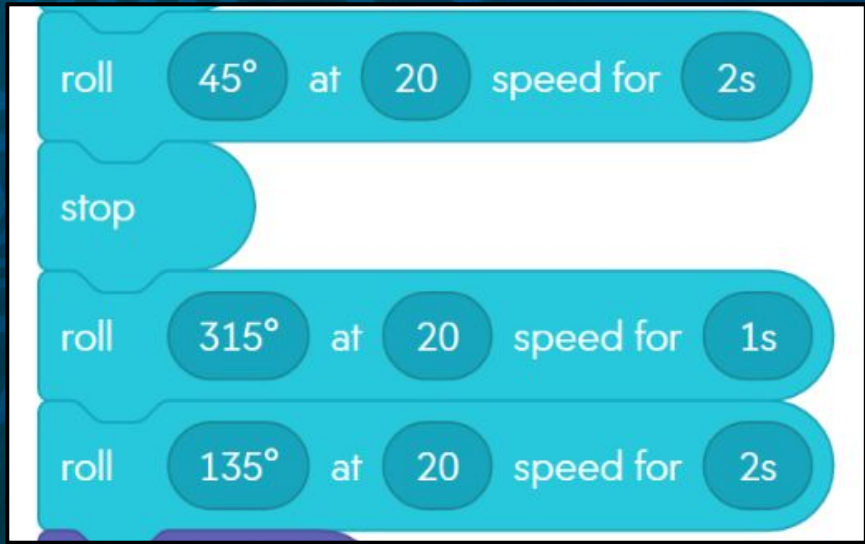
Agility Sprint Challenges

- Ramp Speed & Landing
 - Solution: Let robot fall near ramp back and adjust from there



Agility Sprint Challenges

- Pins
 - Solution: Move robot back and forth to knock over all pins



What We've Learned About Software Engineering

- Every variable must be taken into account and worked around to create an end result that functions properly
- Very time consuming since everything about software engineering is very detail-oriented
- Most of the work is recording information in the System Design Document

What We'd Do Differently

- Ensure the robot is charged before using it to test
- Use a robot on wheels rather than a sphere
- Work around each other's schedules better

The background is a solid blue color with a subtle, intricate pattern of white and light blue lines that resemble a printed circuit board (PCB) or a network diagram. These lines form various geometric shapes, including rectangles, circles, and wavy patterns, creating a sense of depth and complexity.

Questions?