🌊SPLASH💦

Team A2
Josiah Miggiani, Gordon Xu, Jimmy Zhou

Use Case

Have you ever tried for a cup pong shot, only for the ball to bounce off the rim and skitter into the darkest, most inaccessible recesses of your living room? No longer!

Splash is a computer vision-assisted robot that will aid individuals practicing their cup pong shots

- Tracks thrown ping pong balls and moves the target cup to the ball's projected landing location
    - Reduces time spent chasing stray shots
- Provides helpful performance metrics!

ECE Areas:

- Embedded Systems, Hardware, Software and Sensors, Robotics, CV

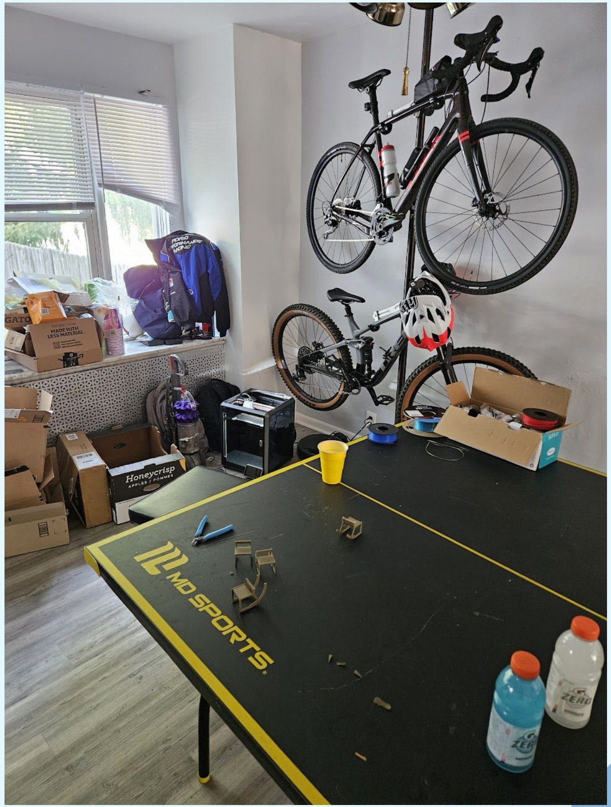
Use Case Requirements

We aim to catch the pong ball ~90% of the time

- Standard throw to be from range of ~2.4 meters (ping pong table length)
- Use red solo cup of size 5 cm radius
- Move system accurately within ~10 cm radius
- To fulfill requirement of making cup pong training effortless

Other requirements to benefit cup pong training

- Retrieval system
- Visual Feedback system



Pictured: A living room that might benefit from Splash :)

Technical Challenges

What are the key technical challenges to meeting the requirements?

1. Quickly detecting and tracking a thrown ball using CV, determining its world coordinates
2. Computing the ball's trajectory and landing location (kinematics projectile motion!)
3. Moving the cup to the landing location within the ball's flight time

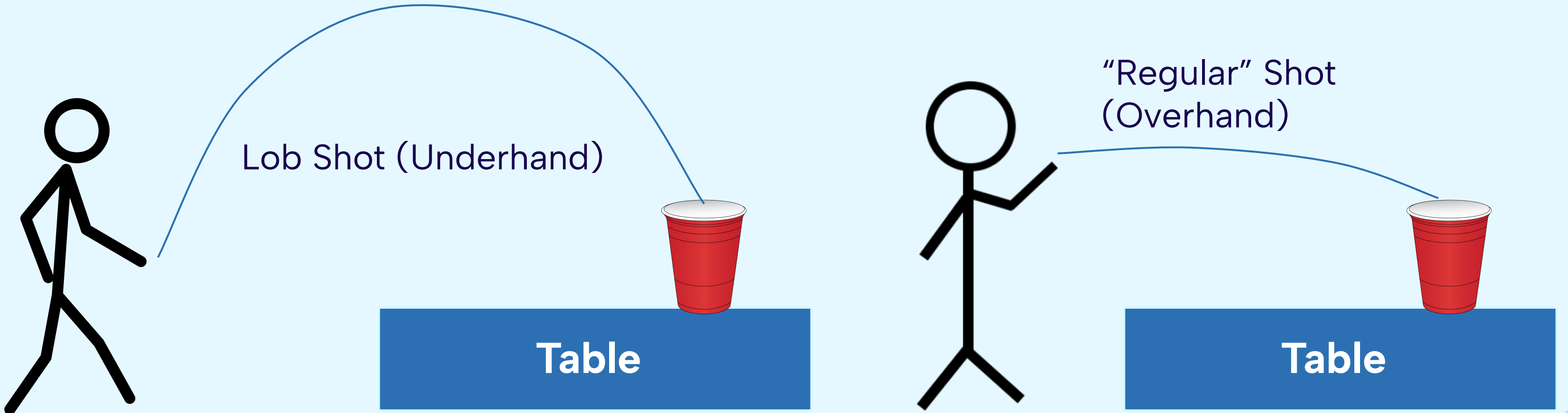
What risk mitigation plans do we have?

To help alleviate the strict timing constraints of this project, we may:

- Require an underhand lob rather than the typical overhand throw for a longer flight time
- Reduce the valid radius around the cup so that the robot doesn't need to move as far
- Require the user to hold the ball still before throwing to give the CV algorithm extra time to lock onto the ball

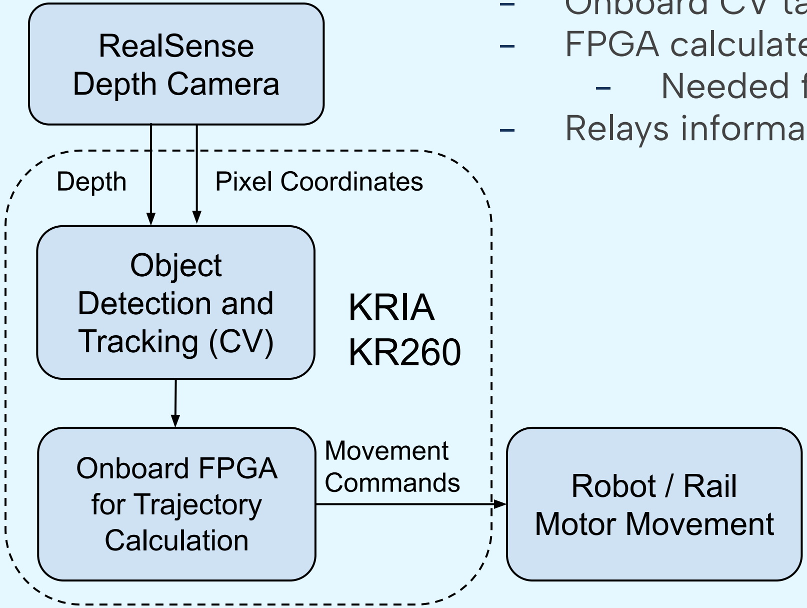
Preliminary Testing

- Concerns about the shot being too fast for system
- Backup plan of making user switch to lob shots if necessary
- Testing from distance of ~2.4m (8ft) Lob shot vs regular shot airtime
  - Regular shots: 1.2s, 1.24s, 0.9s, 1.1s, 1.12s, Average of 1.112s, or 1112ms
  - Lob shots: 1.4s, 1.35s, 1.29s, 1.38s, 1.33s, Average of 1.35s, or 1350ms



Solution Approach (Sensor + HW Integration)

- Intel RealSense D415 or Luxonis OAK-D S2 Depth Camera
  - Low end 30FPS, each frame takes ~33.33ms
  - Enough details for trajectory with 2-5 frames
  - Worst case 5 frames ~= 166.66ms
  - Leaves ~950ms or ~1200ms (lob) to calculate and move receiving system
- AMD KRIA KR260 Board (Onboard CV + FPGA)
  - Onboard CV talks to Camera via USB
  - FPGA calculates trajectory via kinematics
    - Needed for fast calculations during short airtime
  - Relays information to receiving system via USB


RealSense Depth Camera
Depth
Pixel Coordinates
Object Detection and Tracking (CV)
KRIA KR260
Onboard FPGA for Trajectory Calculation
Movement Commands
Robot / Rail Motor Movement

Solution Approach (Sensors SW / CV)

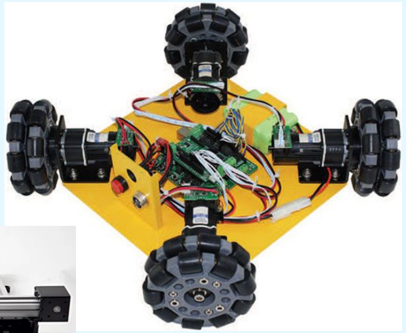
- CV detects ping pong ball from incoming RGB camera feed
- Determine camera-relative coords from depth camera pixels
- Translate ping pong ball coordinates to real-world coordinates
- Becomes a simple projectile motion problem (physics 1!) to determine expected landing location

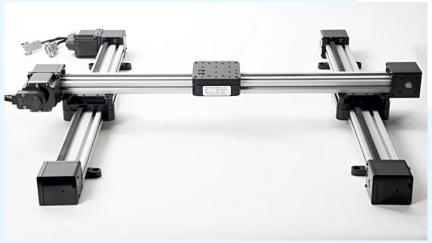
$$x = v_o t + \frac{1}{2}at^2 \qquad v^2 = {v_o}^2 + 2ax$$

$$v = v_o + at \qquad x = \frac{1}{2}(v_o + v)t$$

Solution Approach (Robotics)

- Receives landing spot coordinates from KRIA
- Needs to quickly and accurately move to the coordinate
- Deciding between using a gantry system or a robot





	Cartesian Gantry	Omnidirectional Robot
Pros	More precise coordinate movement Easier implementation for receiving xy coordinate Easier acceleration control	More mobile, is not constrained to an area Easier construction, cheaper
Cons	Physically restrained to an area Harder to physically create Team doesn't have mechanical experience More expensive	Harder to precisely move wheels Slower reaction time for wheel movements

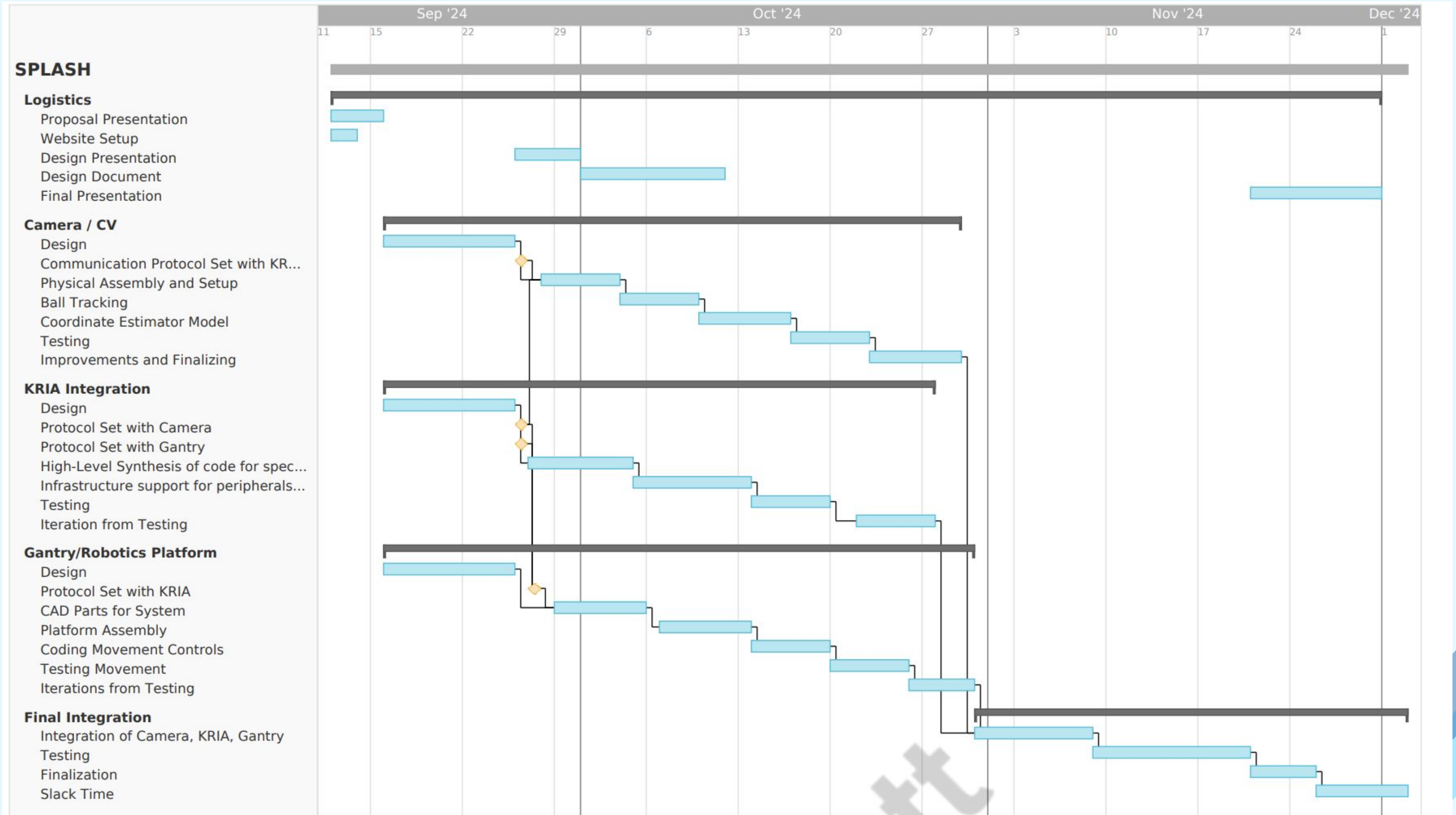
Testing, Verification, and Metrics

1. CV and Coordinate Calculation: **detect and isolate** a ping pong ball with ~98% accuracy and within ~100ms
   a. From known world coordinates, verify that we can **compute the ball's landing location** using data from the depth camera **within 5 cm**
2. Testing Device: device that shoots a ball with consistent precision to a known world position, verify our projected trajectory and landing location is within reasonable bounds.
   a. Alternatively, mark the location the ball lands on the table and compare with the computed landing location
3. Robotics/Gantry System: Verify that our robot can move any location within its radius accurately **within 5 cm** margin of error, and **within 1 second**
4. Overall Integration: test that the robot can **catch the ball shot** from the shooting device consistently (90%), and lastly balls that land within the radius of the cup.

Tasks and Division of Labor

Gordon	Josiah	Jimmy
KRIA/Hardware Integration	Robotics	CV/Cameras
• High-Level Synthesis of code for specific KRIA board • Integrating KRIA with robotics and camera systems	• Design and construction of receiving system • Coding movement controls, prioritizing speed and precision	• Model for ball detection and tracking • Translation from camera coords to real world coords • Physics projectile motion equation

Schedule



SPLASH

Logistics
- Proposal Presentation
- Website Setup
- Design Presentation
- Design Document
- Final Presentation

Camera / CV
- Design
- Communication Protocol Set with KR...
- Physical Assembly and Setup
- Ball Tracking
- Coordinate Estimator Model
- Testing
- Improvements and Finalizing

KRIA Integration
- Design
- Protocol Set with Camera
- Protocol Set with Gantry
- High-Level Synthesis of code for spec...
- Infrastructure support for peripherals...
- Testing
- Iteration from Testing

Gantry/Robotics Platform
- Design
- Protocol Set with KRIA
- CAD Parts for System
- Platform Assembly
- Coding Movement Controls
- Testing Movement
- Iterations from Testing

Final Integration
- Integration of Camera, KRIA, Gantry
- Testing
- Finalization
- Slack Time

Minimum Viable Product

- The cameras + CV identify the exact location of the ball in 3D
- Capable of tracking the trajectory of the ball for at least a few frames
- KRIA system receives coordinate locations, quickly calculates the trajectory, and determines how the robot should move to intercept the ball.
- Receiving system is able to receive instructions and quickly move to the right position

After using our product, the only noise you'll hear is

🌊 **SPLASH** 💦