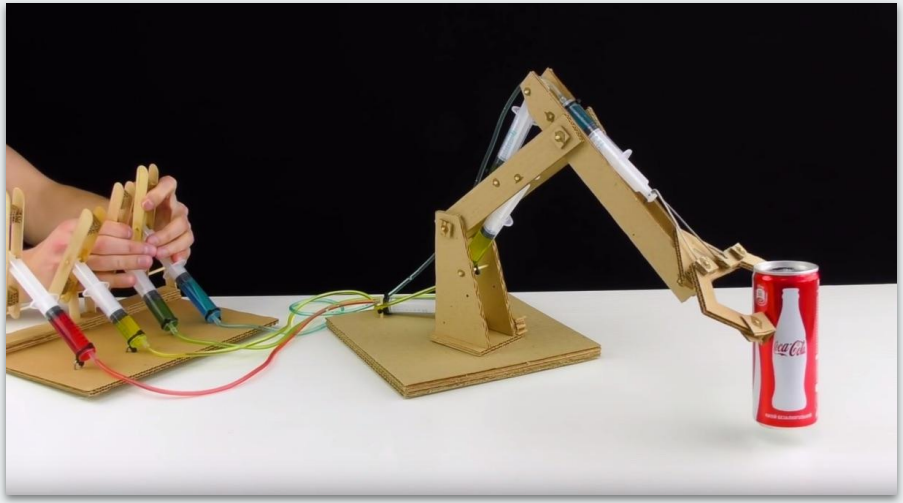
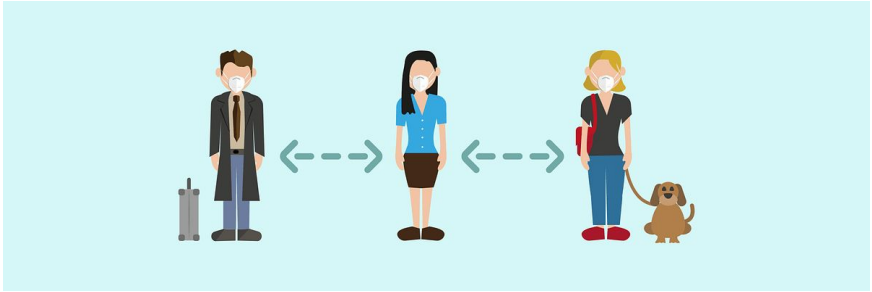
Group E4: Automatic Gentleman

JP Nelson, Juan Pablo Botello, Logan Kojiro



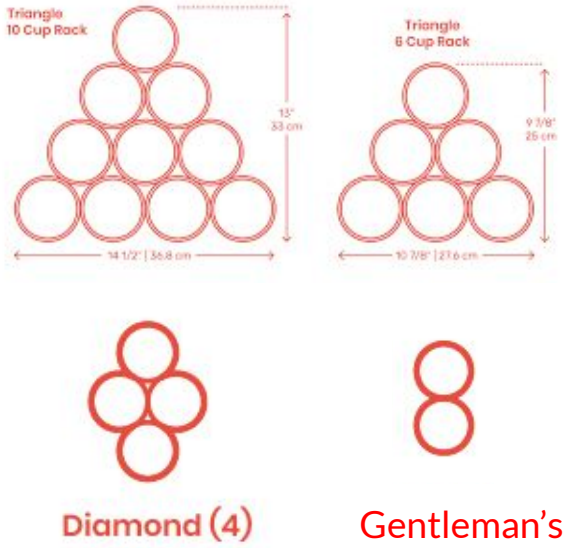
Use Case

- Due to the current situation, it’s impossible to safely play the game of cup pong
- Everyone is out of practice from not playing in a long time
- Missing the social aspect of playing with other people
- Robot doesn’t have to be perfect, only has to be as good as a real person to recreate the experience



General Requirements

- Robotic (automatic) cup pong robot
- Hit cups consistently in various cup formations
- Recognize and display win/lose metrics
- Smooth gameplay experience



Object Recognition Requirements

- Detect ellipses within < 1 s
- Detect only 1 ellipse per cup
- Correctly map depth data on to RGB image
- 90% cup detection on any turn

Motion Planning & Gameplay Requirements

- Track current game state
- Awareness of game rules (stretch goal: optimize shots based on rules)
- Generate accurate direction and launch velocity for shot

Launcher Requirements

- Power and precision
- Rotational freedom
- Low variability in ball exit velocity

Technical Challenges

- **Error Rate In Cup Detection**
  - 90% cup detection with no ellipse duplication on individual cups

- **Consistency in Launcher Velocity**
  - To be within +-1.5”(0.0381m) (standard cup radius), exit velocity needs to be approximately +-1% of the target velocity (calculated at 2 meters)

$$|v_i| = \sqrt{distance * g}$$

$$\frac{\sqrt{(x + 0.0381) \times 9.81} - \sqrt{x \times 9.81}}{\sqrt{x \times 9.81}} \times 100$$

Solution Approach

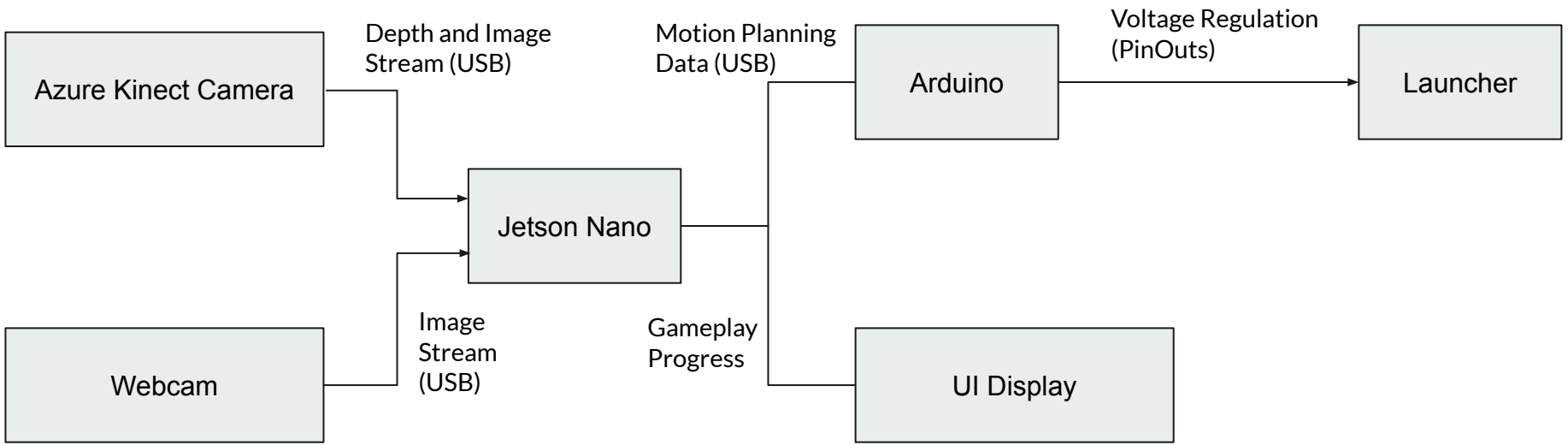
- **Object Recognition**
  - Azure kinect camera + OpenCv to detect circles/ellipses on cups
  - Transform pixels to real-world coordinate system

- **Motion Planning**
  - Standard kinematic equations to calculate direction and velocity

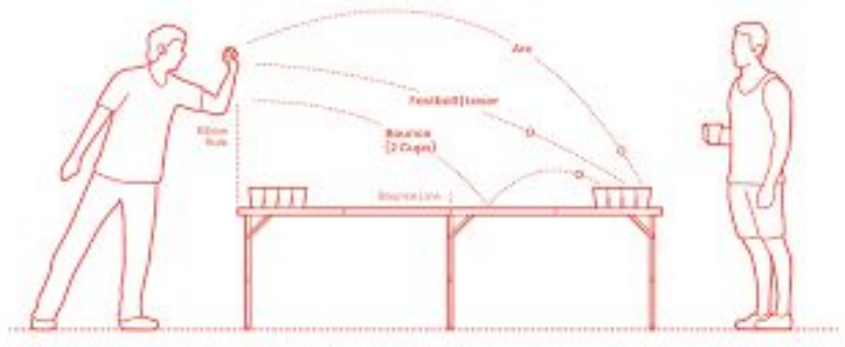
Linking Launcher Velocity to Input Voltage

- Receive velocity and angle over USB to arduino
- Use arduino PWM to set voltage of launcher
- Drive motor to rotate launcher to correct angle
- Measure relationship between voltage and exit velocity with slow motion camera and measured boxes
    - Use slow motion camera to calculate the exit velocity

Architecture



Testing & Verification

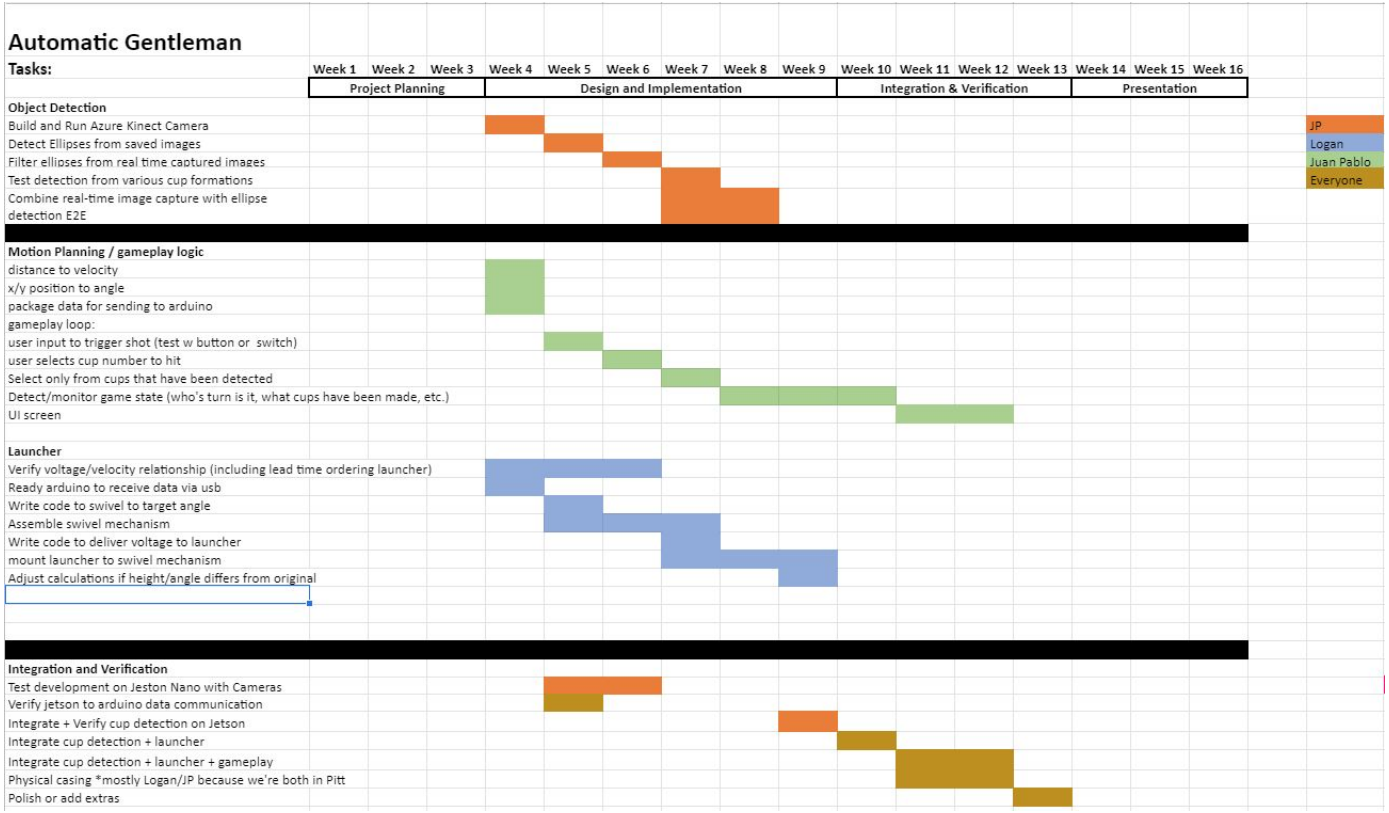


- Detection rate with 4 cup formations
- Velocity consistency of launcher
- Overall gameplay experience (time it takes for game, etc)
- We will measure the consistency/accuracy of real people playing as a target for the robot
    - Have X different people play a full game.
    - We will have them count the number of balls it takes them to make all 10 cups
    - Want our robot to be at least within one standard deviation of the average player, hopefully better

Division of Labor

- Object Recognition - JP
  - Gather image/depth data
  - Process cup location coordinates
- Gameplay Mechanics - Juan Pablo
  - Motion planning using coordinate info
  - Monitor/ control gameplay (lose/win, game UI, etc)
- Projectile Dynamics - Logan
  - Trigger launcher to shoot at desired exit velocity
  - Rotate launcher to aim at desired cup
- Fabrication - Logan & JP (both in Pittsburgh)

Schedule


Automatic Gentleman
Tasks:
Week 1 Week 2 Week 3 Week 4 Week 5 Week 6 Week 7 Week 8 Week 9 Week 10 Week 11 Week 12 Week 13 Week 14 Week 15 Week 16
Project Planning
Design and Implementation
Integration & Verification
Presentation
Object Detection
Build and Run Azure Kinect Camera
Detect Ellipses from saved images
Filter ellipses from real time captured images
Test detection from various cup formations
Combine real-time image capture with ellipse detection E2E
Motion Planning / gameplay logic
distance to velocity
x/y position to angle
package data for sending to arduino
gameplay loop:
user input to trigger shot (test w button or switch)
user selects cup number to hit
Select only from cups that have been detected
Detect/monitor game state (who's turn is it, what cups have been made, etc.)
UI screen
Launcher
Verify voltage/velocity relationship (including lead time ordering launcher)
Ready arduino to receive data via usb
Write code to swivel to target angle
Assemble swivel mechanism
Write code to deliver voltage to launcher
mount launcher to swivel mechanism
Adjust calculations if height/angle differs from original
Integration and Verification
Test development on Jeston Nano with Cameras
Verify jetson to arduino data communication
Integrate + Verify cup detection on Jetson
Integrate cup detection + launcher
Integrate cup detection + launcher + gameplay
Physical casing *mostly Logan/JP because we're both in Pitt
Polish or add extras
JP
Logan
Juan Pablo
Everyone