Team e4: Automatic Gentleman

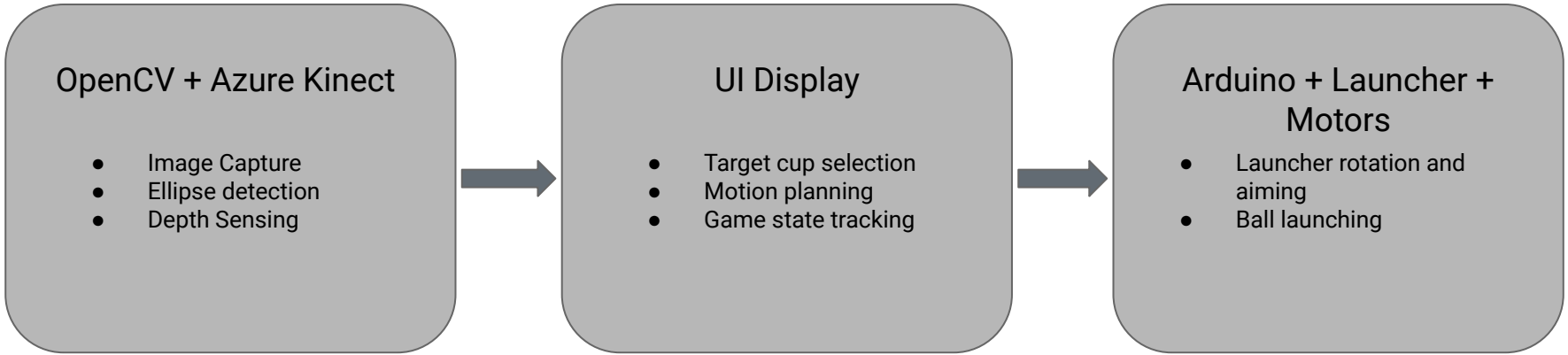
JP Nelson, Logan Kojiro, Juan Pablo Botello

Design Presentation

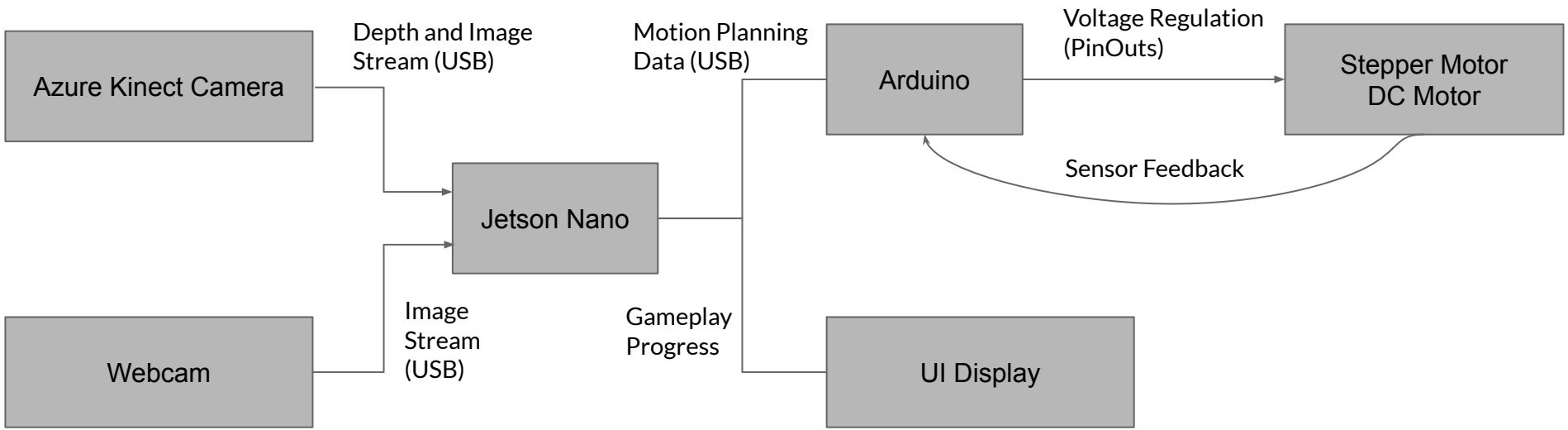
Application Area: Cup Pong Robot

- Solo practice
  - Can’t always find someone to play with
  - With current times, difficult to play a game safely
- Facilitate Virtual Competition (stretch)
  - Real-time game with opponent who is somewhere else

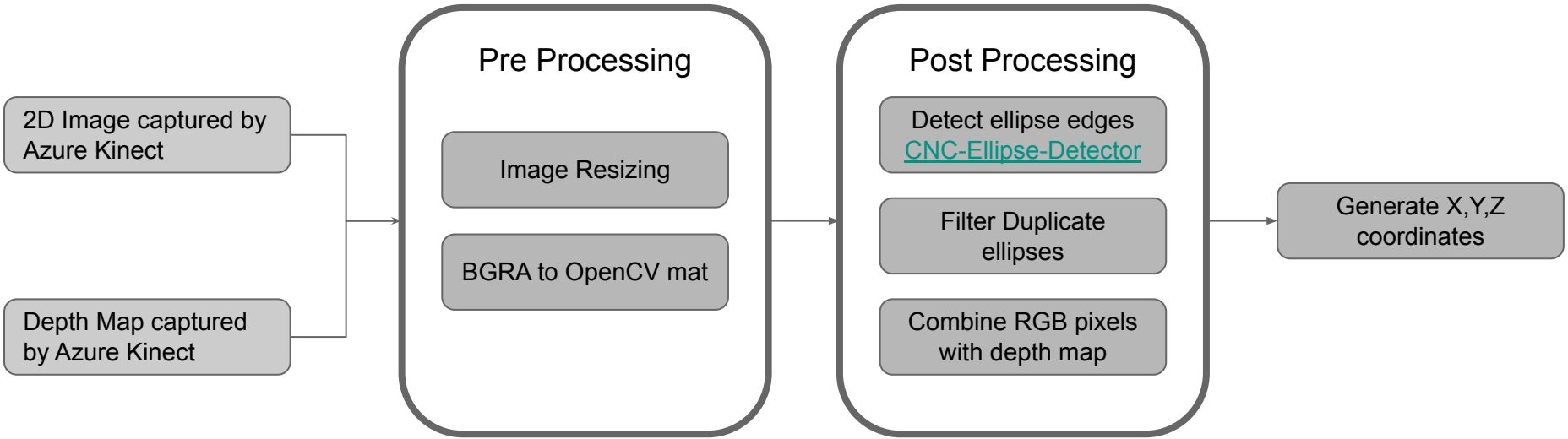
Solution Approach



System Specification



Implementation Plan: Image Processing



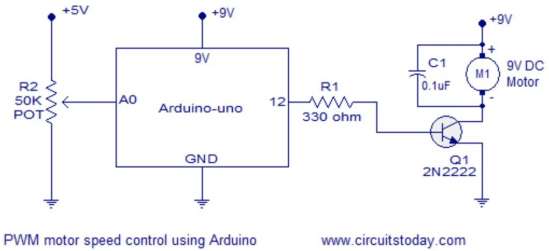
Implementation Plan: Launcher

Original Plan:

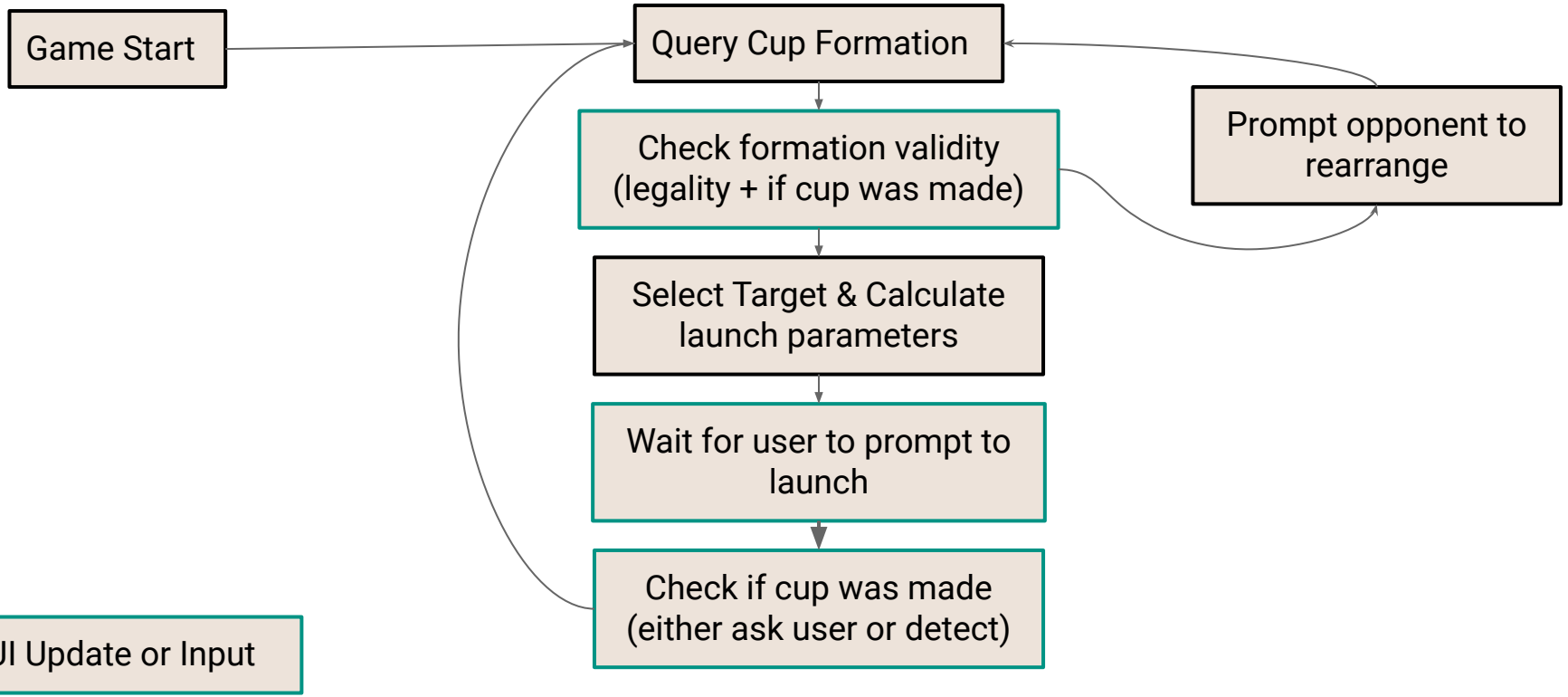
- Repurpose premade launcher

New Plan:

- DC motors can be controlled directly with PWM
- Test to extrapolate voltage/velocity relationship with fixed radius
- Similar Design to Stanford Ball robot launcher



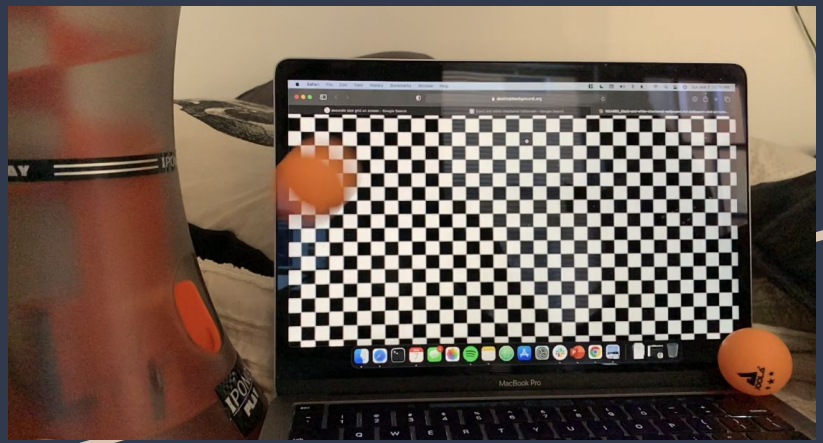
Implementation Plan: Gameplay/Planning



Metrics & Validation

Component	Requirement	Metric	Pass
System	Recreate experience of a game	Number of cups needed to complete a game	Robot accuracy is within one std of average human
Cup Detection	Accurate ellipse edge detection	Number of cups detected	Detect 1 ellipse per cup, Detect ellipses < 1s, 90% cup detection any given turn
Launcher	Consistency exit velocity	Velocity of the ball exiting launcher	With many trials, want within 1% consistency of exit velocity
Rotator	Hit correct angle	Measured angle	Within 1 degree angle accuracy

Testing: Launcher



Initial Exit Velocity Testing

- Grid square = ⅓”
- Framerate: 240 fps (1 frame = 4.167ms)
- Offers rough estimate of launcher velocity

Initial Findings

- Spring Mechanism undesirable
- Much more inconsistent than we would like
- As is currently no way to control power

Moving Forward

- Larger squares
- More light for less blurr

Rotator Testing

- Measure found angle against target angle

Testing: Image Processing

Cup Detection

- Completed ellipse detection in ~800 ms
- Average 2 - 3 ellipses per cup with minimal filtering

Risk Factors & Mitigation

Problem	Solution
Losing track of game progress (i.e remaining cups)	Allow manual input of game state through UI
Accuracy values are way below standard due to external factors	Shot adjustment based on shot percentage in addition to base coordinate calculation
Automatic Gentleman is too good	Manually lower performance rating through UI option

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