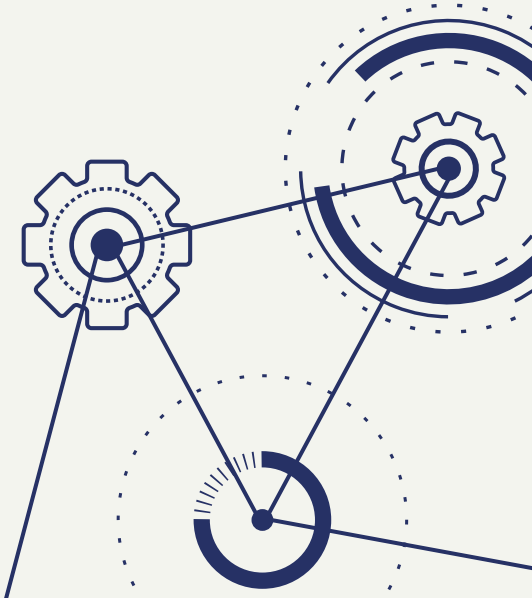
ME134: Carrom Bot

Neil, Julian, Pat, and Krishna

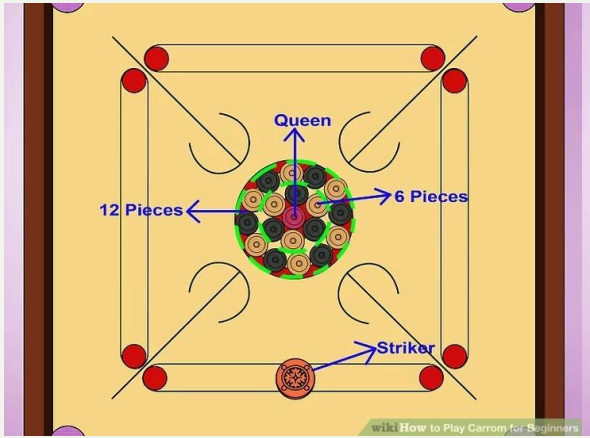


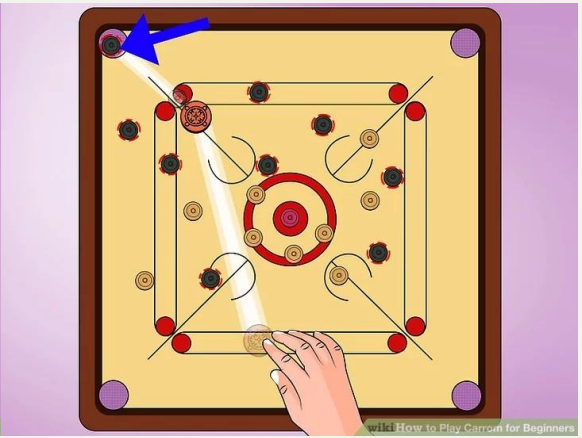
Basic Task: Carrom

Popular Indian game played on square board that is comparable to billiards and shuffleboard

Task:

- Pick up the striker
- Place the striker on its respective axis
- Shoot an optimal shot
- Simultaneously detect the state of the pucks, the score, and the who is currently playing (Robot vs Player)



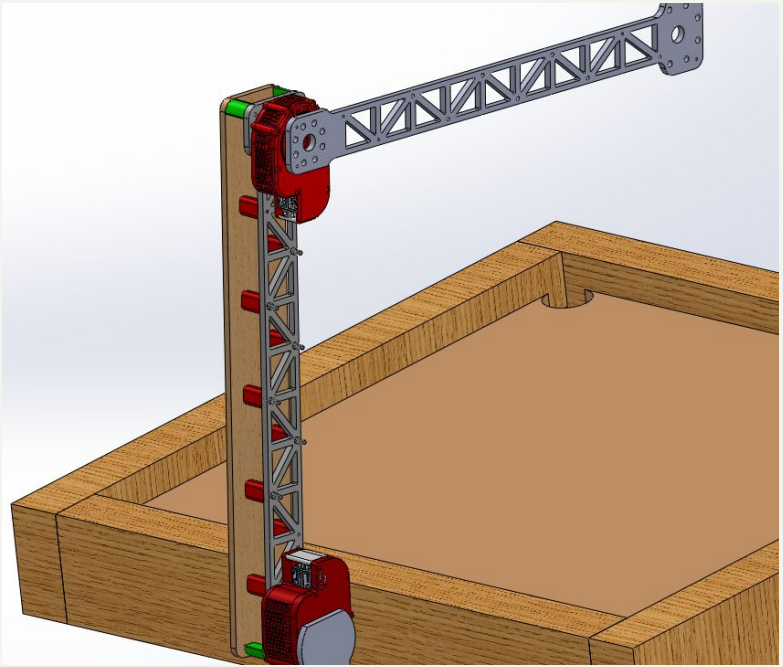


4 Types of Pieces:

- 10-point
- 20-point
- Queen (50-point)
- Striker

Design - Arm Assembly

- 5-DOF (3-DOF + wrist and rotation) arm with 2 additional degrees from gripper and solenoid striker
- Aluminum + wood sandwich to minimize base link vibrations
- Approximate workspace includes 2ft x 2ft Carrom board within a table top surface constrained by Aruco markers
- Typically, the arm operates bend towards the board with the wrist motor carrying the end effector at angle zero



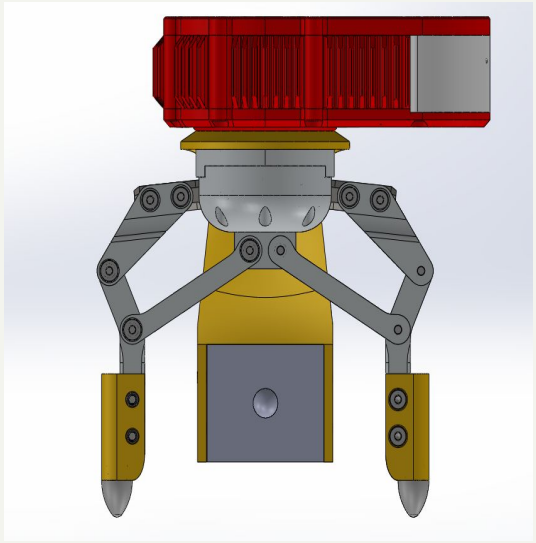
Design - End-Effector

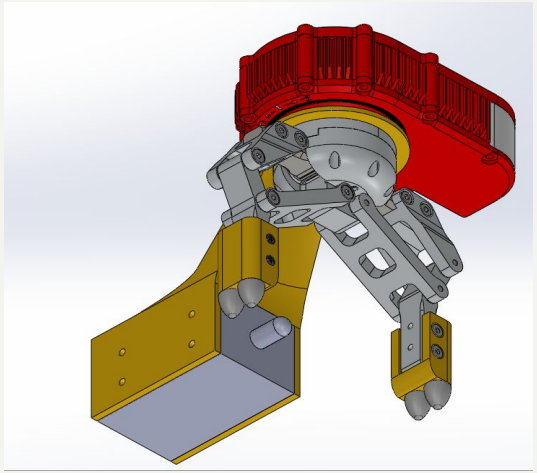
Gripper

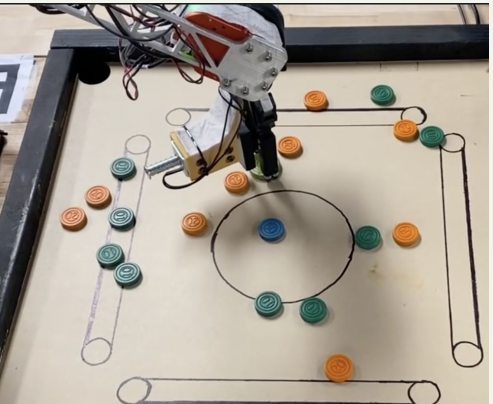
- Rubber tips to grab pucks
- Elongated to avoid collisions with solenoid when gripping

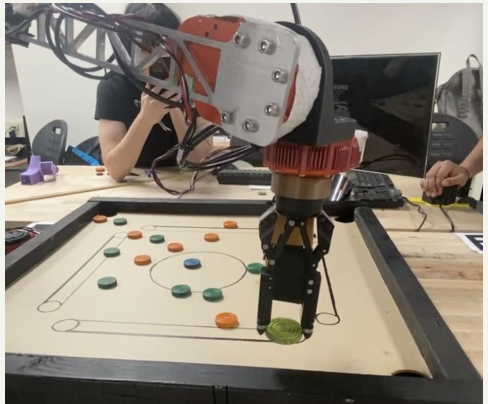
Solenoid with rectangular attachment

- Operated via custom driver circuit
- Elevated to avoid board collisions









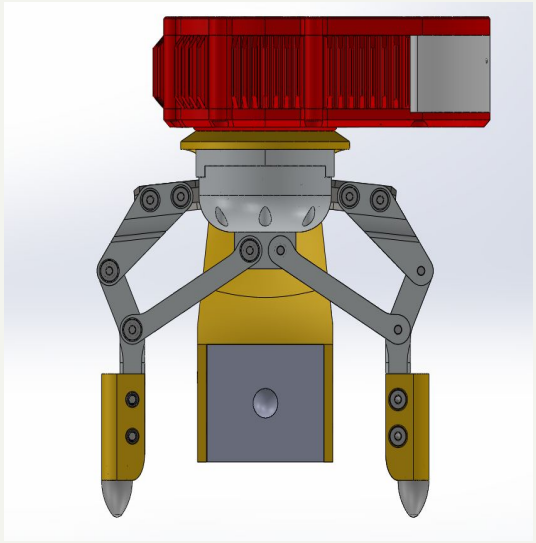
Design - End-Effector

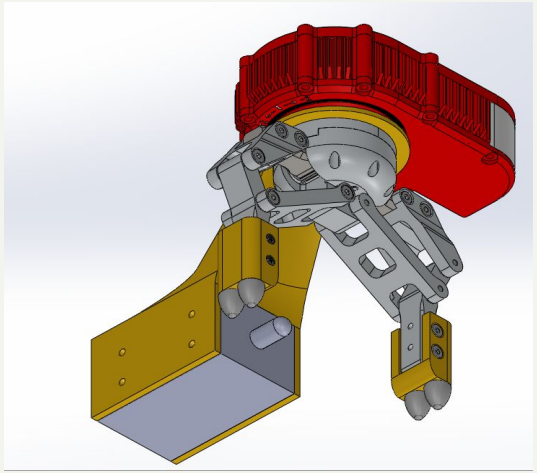
Gripper

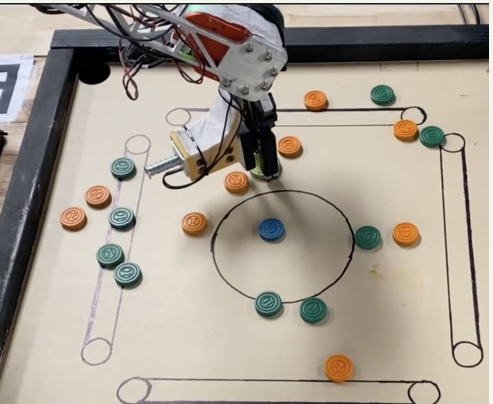
- Rubber tips to grab pucks
- Elongated to avoid collisions with solenoid when gripping

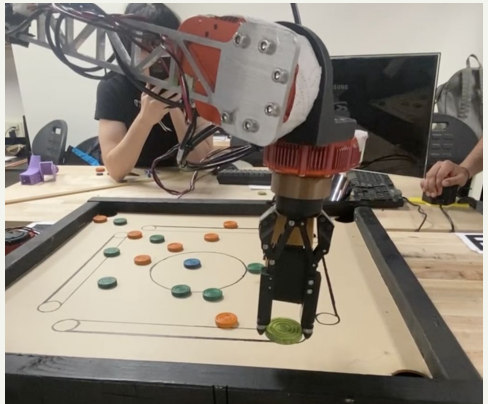
Solenoid with rectangular attachment

- Operated via custom driver circuit
- Elevated to avoid board collisions









Detection

Puck Detectors

- 4 types of pucks (10, 20, Queen/50, Striker) were tuned by HSV Values, filtered based on expected area
- Flipped puck (underside not painted) had similar HSV as border of board,

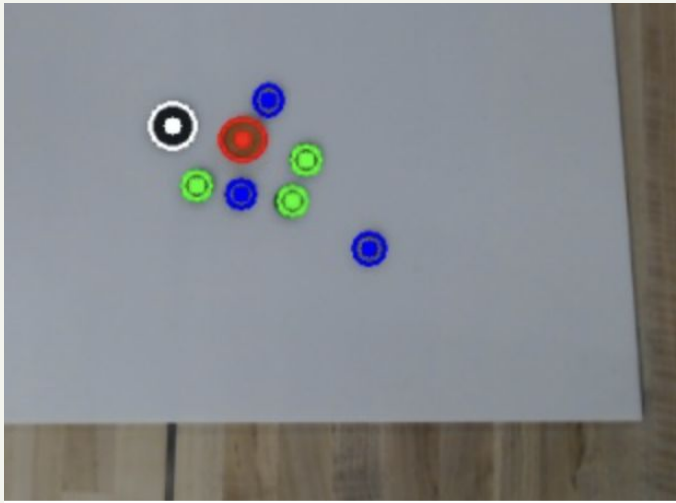
Board Detector

- HSV values to detect board border
- Process the border corners and center

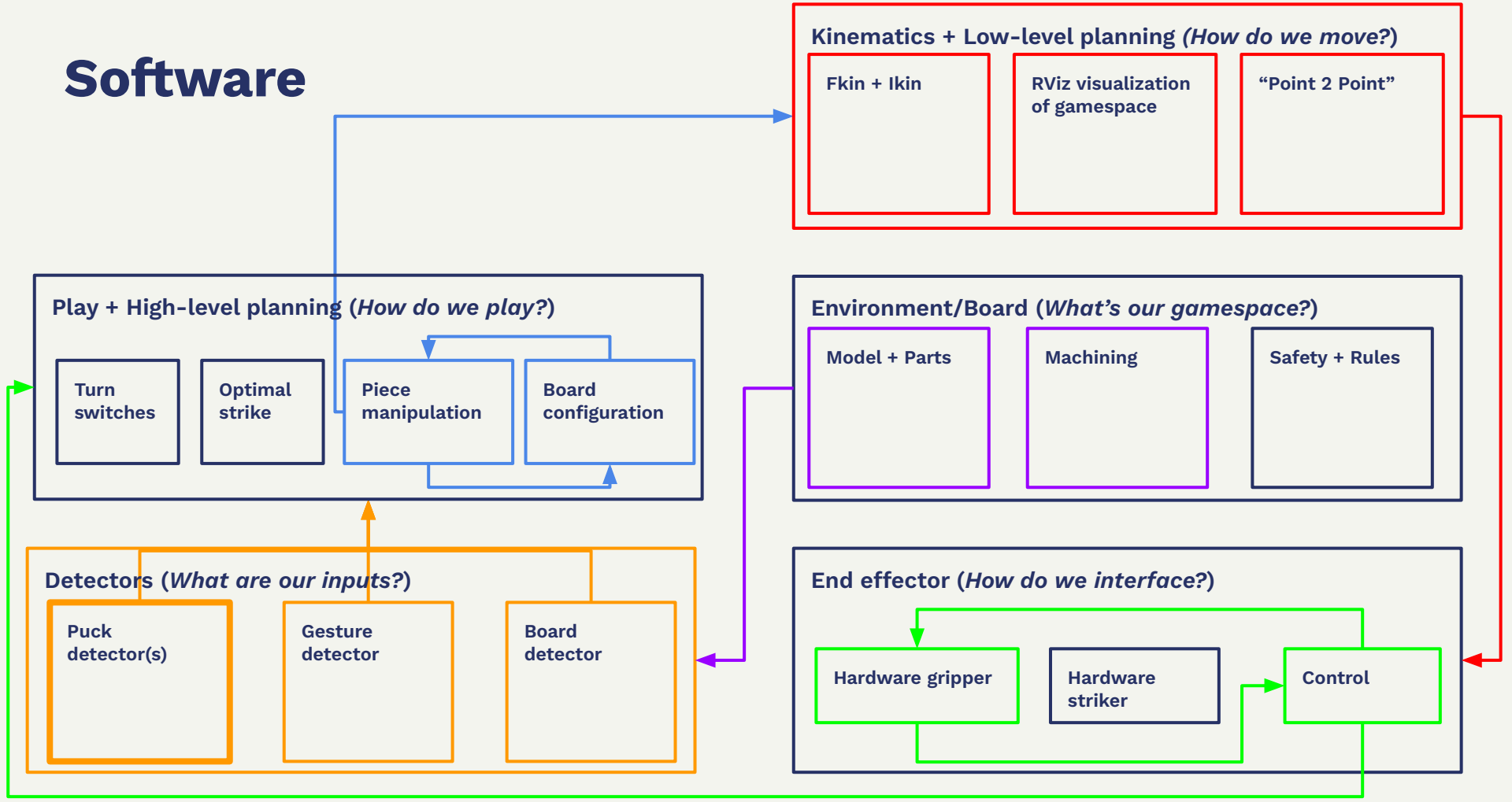
Gesture Detector

- Model to detect thumbs up or thumbs up gesture to detect end of player's turn

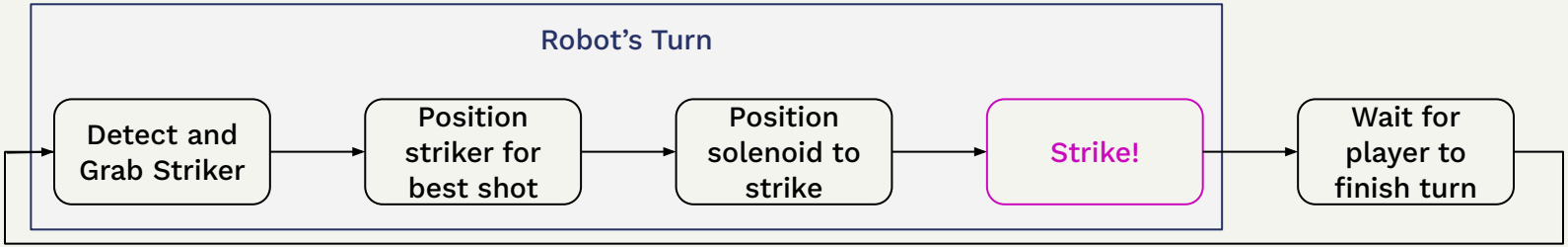
All use **Aruco** markers and error maps to convert pixel coordinates to world coordinates



Software

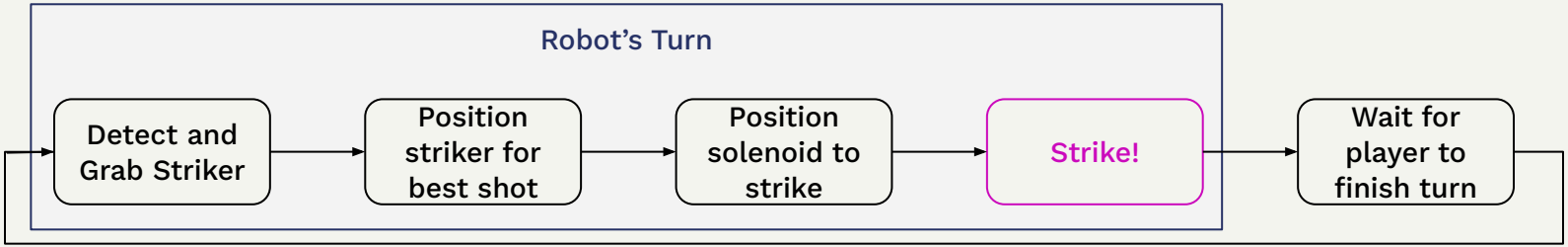

Kinematics + Low-level planning (How do we move?)
Fkin + Ikin
RViz visualization of gamespace
“Point 2 Point”
Play + High-level planning (How do we play?)
Turn switches
Optimal strike
Piece manipulation
Board configuration
Environment/Board (What’s our gamespace?)
Model + Parts
Machining
Safety + Rules
Detectors (What are our inputs?)
Puck detector(s)
Gesture detector
Board detector
End effector (How do we interface?)
Hardware gripper
Hardware striker
Control

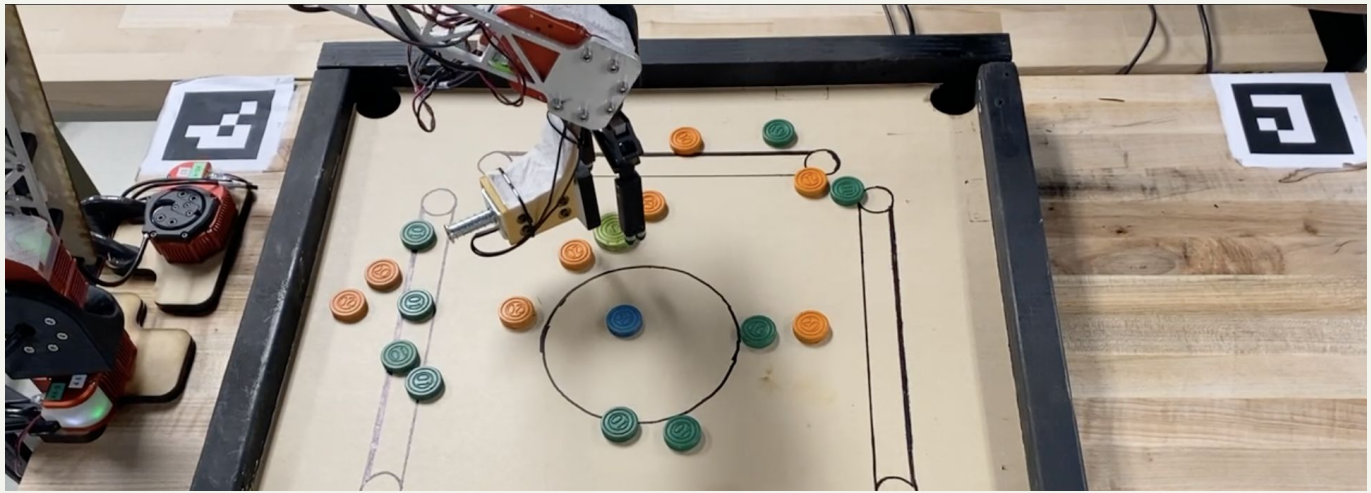
Normal Behaviors and Failure Recovery



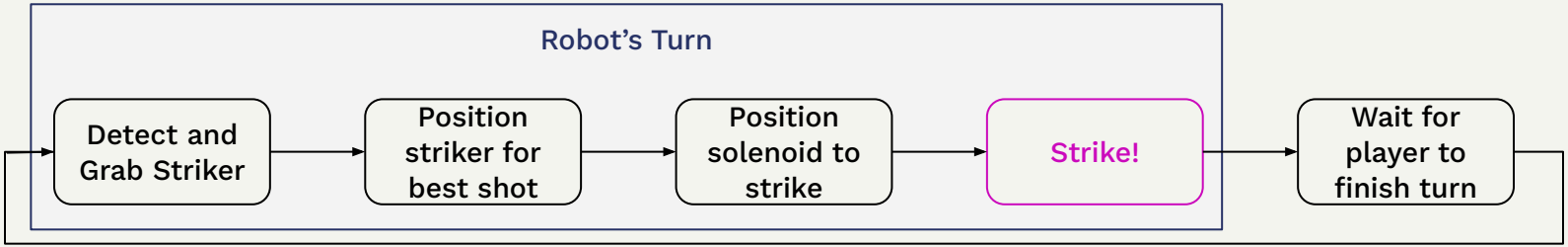
At each transition point, assumptions are made about the robot's state and success at the previous task → check these!

Normal Behaviors and Failure Recovery





Normal Behaviors and Failure Recovery



Shortcomings and Lessons Learned

Arm Design

- Initial design was longer and thinner with no offset, resulting in too much wobbling and weakness. First link was updated to be thicker with an offset to account for weak axis bending and torsion.
- A lot of power is needed to make an effective shot. A future version could package this power source more compactly

Software

- Low-level movement code evolved over time → could be refactored to clean-up and potentially simplify

Video