

Project Joseph Hoane: A Robotic Chess Player

By: SD02

Team members: Autrin Hakimi, Cal Hokanson-Fuchs, Junhyung Shim,
Nhat Anh Bui, Thanh Mai

Client: Dr. Bowen Wang

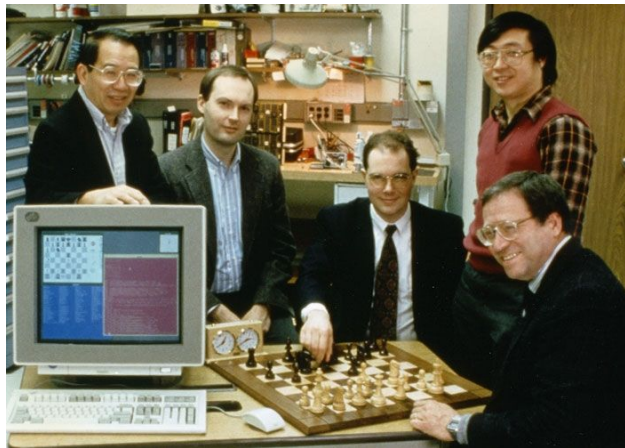
Computer Science Department

Iowa State University

02/25/2025

Problem Statement

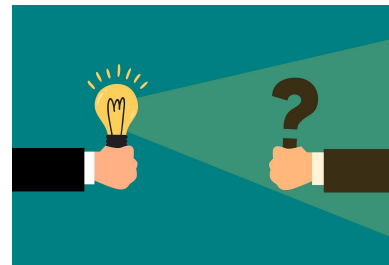
- Joseph Hoane is the man behind Deep Blue, a revolutionary AI that, in 1996, was the first ever to beat the reigning world chess champion.
- AI has surpassed humans in chess, but robotic intelligence lacks physical embodiment.
- Our goal is to enable a robotic manipulator to play chess in the real world.



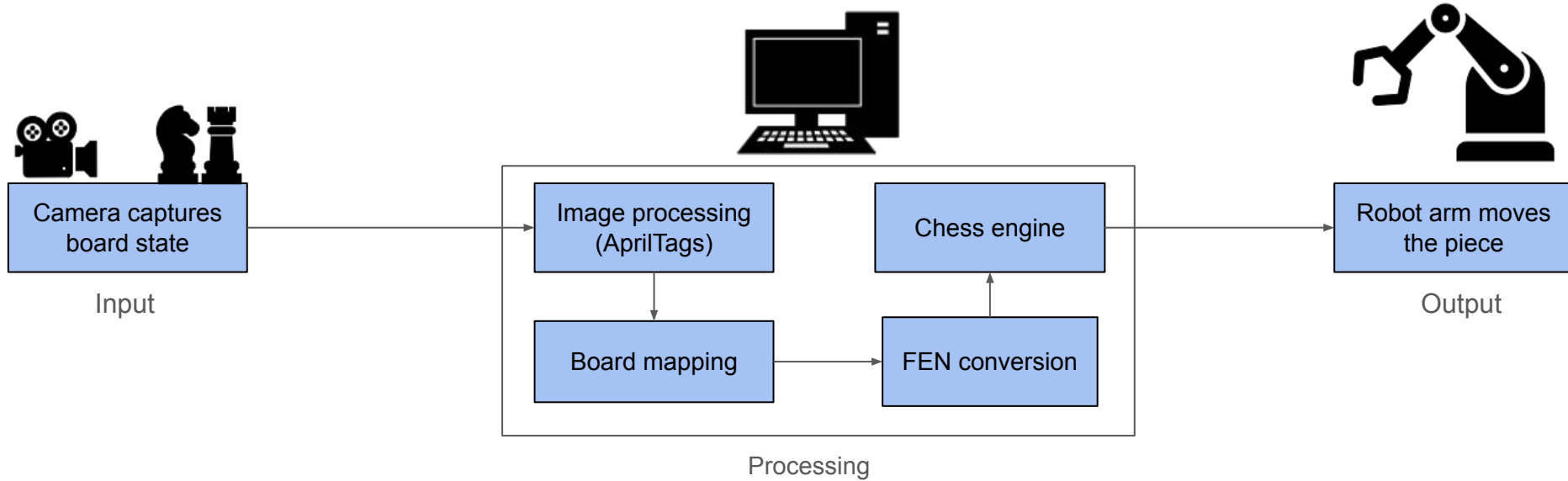
Solution Overview

What our system does in a high-level manner:

- Uses a **robotic arm** to physically move chess pieces.
- Uses a **camera and AprilTags** for board and piece recognition.
- Uses a **chess engine** for move calculation.
- Bridges perception and action using **ROS (Robot Operating System)**.



System Architecture



Hardware and Software Used

Hardware:

- UR10e robotic arm + gripper
- GoPro camera
- 3D-printed chess pieces



Software:

- Ubuntu 20.04
- ROS (Robot Operating System)
- OpenCV (for image processing)
- Stockfish (chess engine)
- Python



Chess Engine Integration

- The camera captures the board.
- The board state is converted to a **FEN string**.
- Stockfish receives the FEN and calculates the best move.
- The best move is converted into a physical move for the robotic arm.



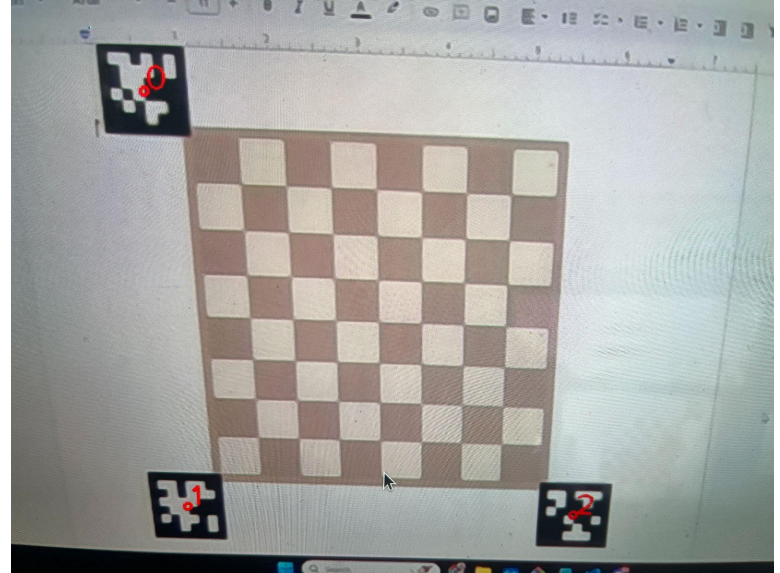
Input: position fen 6Q1/8/r4r1k/P4P2/5K2/7p/8/8 w - - 0 60
go depth 5



Output: bestmove g8h8

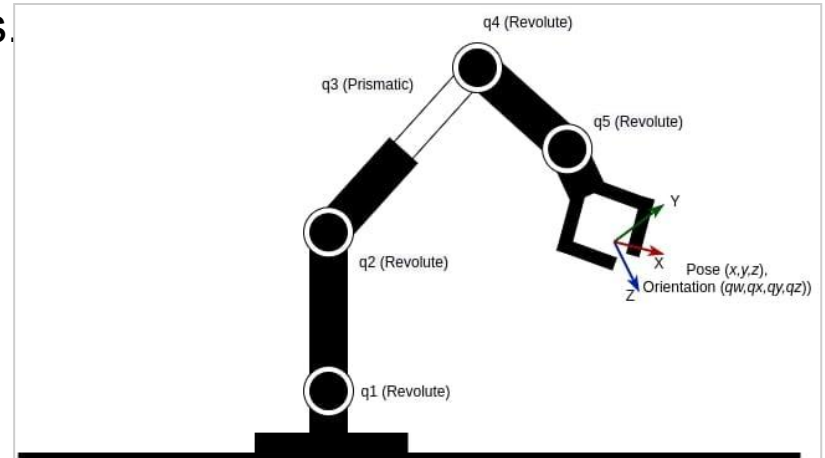
Board Detection Using AprilTags

- Three AprilTags on the chessboard allow precise **board mapping**.
- Camera detects and aligns the board coordinates.

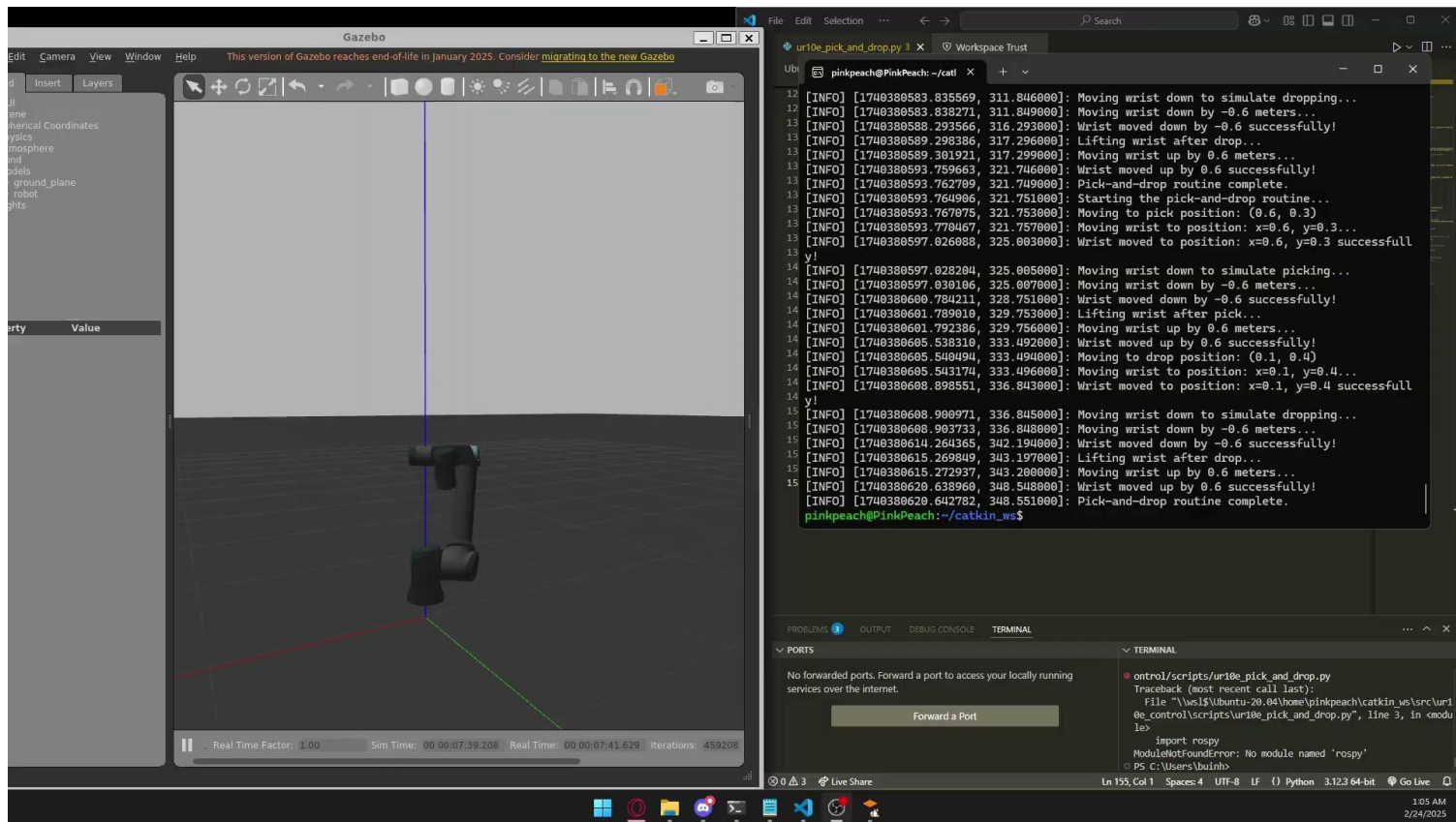


Robotic Arm Control

- Uses MoveIt in **ROS** for motion planning.
- Takes the **best move from Stockfish** and translates it into joint movements.
- Uses **inverse kinematics** to calculate joint position given coordinates.
- Handles precise pick-and-place actions.



Simulated Pick-and-Place Motion in ROS



Current Progress

- ✓ Installed **Ubuntu 20.04** and **ROS**
- ✓ Integrated **chess engine (Stockfish)**
- ✓ Implemented **AprilTag-based board recognition**
- ✓ Converted board state into **FEN string**
- ✓ Stockfish can now **calculate legal moves**
- ✓ Basic robotic arm movements implemented

Challenges Faced

- **Board Alignment Problems** → Using **AprilTags** for precise mapping.
- **Camera Calibration Issues** → Adjusting contrast and gamma correction.



Next Steps & Timeline

What remains to be done?

- Fine-tune **robotic arm movement** for accurate pick-and-place.
- Improve **chess piece detection** using better **image processing**.
- Add a **user interface** for interaction.
- Perform extensive **testing** for real games.

