Project Joseph Hoane

Esha Gadgil, John Nesnidal, Savannah Franklin, Will Norris, Jared Wong

History

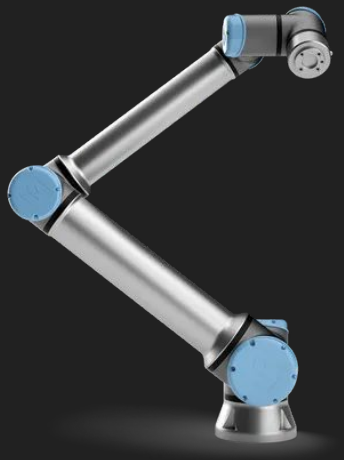
- Joseph Hoane Jr. - computer scientist involved in IBM’s Deep Blue chess engine project
- Lacked one thing: couldn’t interact with the board
- Hoane had to move the pieces on behalf of the machine
- **How do we make a robotic chess player that can actually play chess?**

Project Overview

Develop a fully autonomous UR10e robotic system capable of playing chess against a human opponent or another robot. This project integrates advanced robotics, artificial intelligence, and strategic algorithmic problem-solving to achieve seamless interaction and precise gameplay

Client and Equipment

Client: Dr. Bowen Weng



How will the robot see the chessboard?

- GoPro Camera
- QR Codes or April Tags

How will the robot move the pieces?

- Universal Robotics UR10e Robot Arm

Key Project Components

Stockfish

Open-Source Chess Engine

OpenCV

Computer Vision Library

ROS

Robot Operating System

Gazebo

Robot Simulator

Stockfish

What is Stockfish?

- Free, open-source, chess engine
- Considered the strongest chess engine available to the public

Why we picked Stockfish over other chess engines

- Top rated
- Higher Elo compared to other engines

How are we going to use Stockfish?

- In order to input raw chess move data into Stockfish, we need a way for the robot to visualize and process the board

#	Chess Engine	Elo
1	Stockfish 20231230	3845
2	Berserk 12	3719
3	RubiChess 20231230	3711
4	Dragon 3 by Komodo Chess	3700
5	Clover 6.1	3681
6	Koivisto 9.0	3647
7	Caissa 1.15 (Wine)	3633
8	Alexandria 5.1.1	3613
9	rofChade 3.1	3585
10	Revenge 3.0 (Wine)	3579
11	Seer 2.6.0	3554
12	Black Marlin 8.0	3551
13	Stormphrax 4.0.0	3533
14	Viridithas 11.0.0	3533
15	Altair 6.0.0	3502
Average Elo		3626

OpenCV

What is OpenCV?

- Open Source Computer Vision Library
- Provides tools for image and video processing
- Supports multiple programming languages (e.g., Python, C++, Java)
- Extensive range of functions for image manipulation, object detection, and more

Why Use OpenCV for Chess?

- Analyzes the chessboard grid.
- Detects QR codes on chess pieces.
- Captures and processes live video from GoPro.

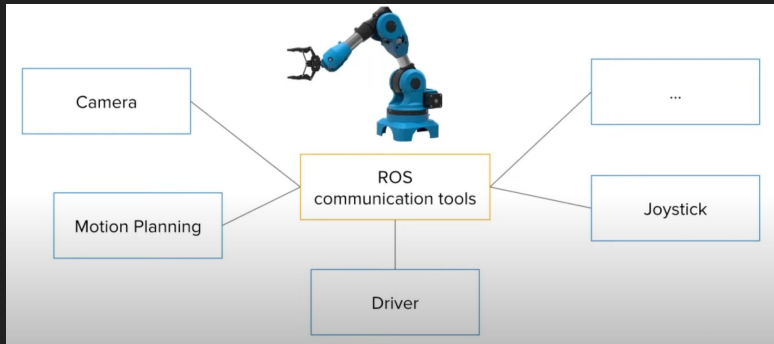
ROS (Robot Operating System)

What is ROS?

- Open-source framework designed to facilitate the development of robotic applications
- Provides essential tools, libraries, and drivers to control hardware

Why ROS?

- Acts as a middleware and framework built for robotics applications
- ROS has drivers for both the robot arm and the gripper
- Includes a simulator that can be used to test the robot arm virtually
- Great for beginners
- Plug & Play Libraries



Development Plan

Phase 1: Software Setup (complete)

Development environment setup

Stockfish integration and tests

ROS integration and tests

Robot simulator setup and tests

Phase 2: Hardware Setup

Integration with physical robot arm

Camera installation and setup

OpenCV integration and tests

Interface between OpenCV and Stockfish

Robot arm calibration

Phase 3: Actually Playing Chess

Interface between Stockfish and robot arm

Robot arm motions

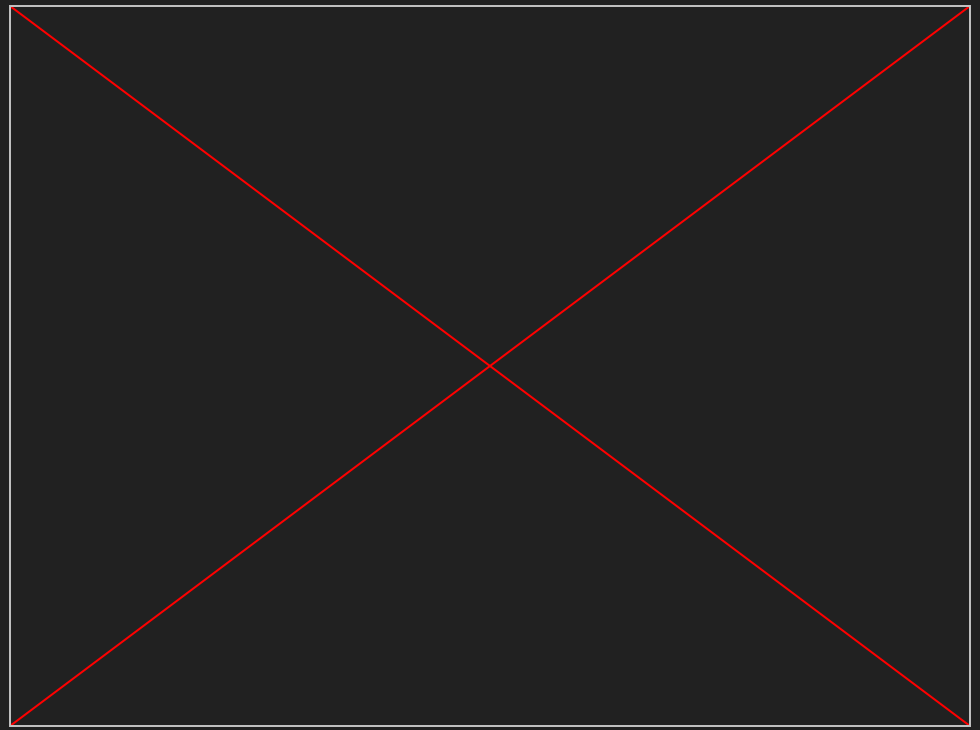
Final debugging

Stockfish Setup and Integration

```
White is represented as UPPERCASE
Black is represented as lowercase
+---+---+---+---+---+---+---+---+
| R | N | B | K | Q | B | N | R |
+---+---+---+---+---+---+---+---+
| P | P | P | P | P | P | P | P |
+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
| p | p | p | p | p | p | p | p |
+---+---+---+---+---+---+---+---+
| r | n | b | k | q | b | n | r |
+---+---+---+---+---+---+---+---+
```

```
Final board after simulation (white pawn promoted):
+---+---+---+---+---+---+---+---+
| r | Q | b | q | k | b | n | r |
+---+---+---+---+---+---+---+---+
| p |   | p | p | p | p | p | p |
+---+---+---+---+---+---+---+---+
|   |   | n |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
|   |   | p |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
| P |   |   | P | P | P | P | P |
+---+---+---+---+---+---+---+---+
| R | N | B | Q | K | B | N | R |
+---+---+---+---+---+---+---+---+
```

Robot Arm Simulation (Gazebo)



Next steps

1. Test Existing Code on the Actual Robot Arm

- Verify the robot arm's movement is consistent with the simulated behavior
- Observe the arm's trajectory to ensure smooth and accurate motions
- Use safety measures like emergency stops during initial testing

2. Calibrate the Robot Arm with the Chessboard

- Physically measure and set the chessboard's position relative to the robot base
- Run calibration scripts to align the end effector with specific board squares
- Fine-tune joint angles based on calibration results
- Test the robot's ability to consistently move to predefined board positions
- Ensure accuracy when picking and placing chess pieces

Questions?