CprE/EE/SE 492 IRP Review

Team: sdmay21-05

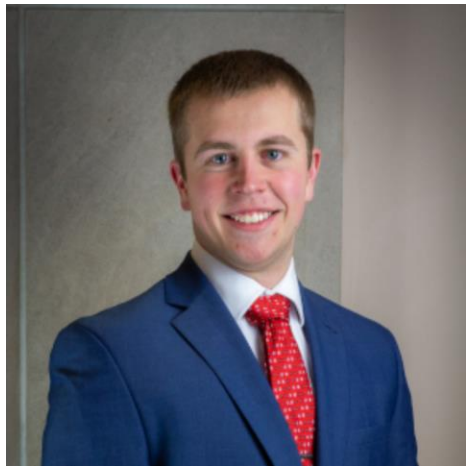
Client: Dr. Joseph Zambreno

https://sdmay21-05.sd.ece.iastate.edu

4.30.2021

AGENDA

- INTRODUCTIONS
- PROJECT VISION
- USE CASES
- CONCEPTUAL SKETCH
- REQUIREMENTS AND STANDARDS
- CONCEPTUAL DESIGN DIAGRAM
- RISKS AND MITIGATIONS
- FUNCTIONAL DECOMPOSITION
- USER INTERFACES
- COMPUTER VISION
- TESTING
- DEMO
- LESSONS, KNOWLEDGE & SKILLS
- EXISTING CHALLENGES AND NEXT STEPS



MEET THE TEAM

Dillon Peters

Team Lead

Jamie Peterson

Mobile Lead

Parker Bibus

Computer Vision Lead

Aidan Sherburne

Report Manager

Jake Aunan

AR Lead

Brett Santema

Test Lead

Dr. Zambreno

Advisor and Product Owner

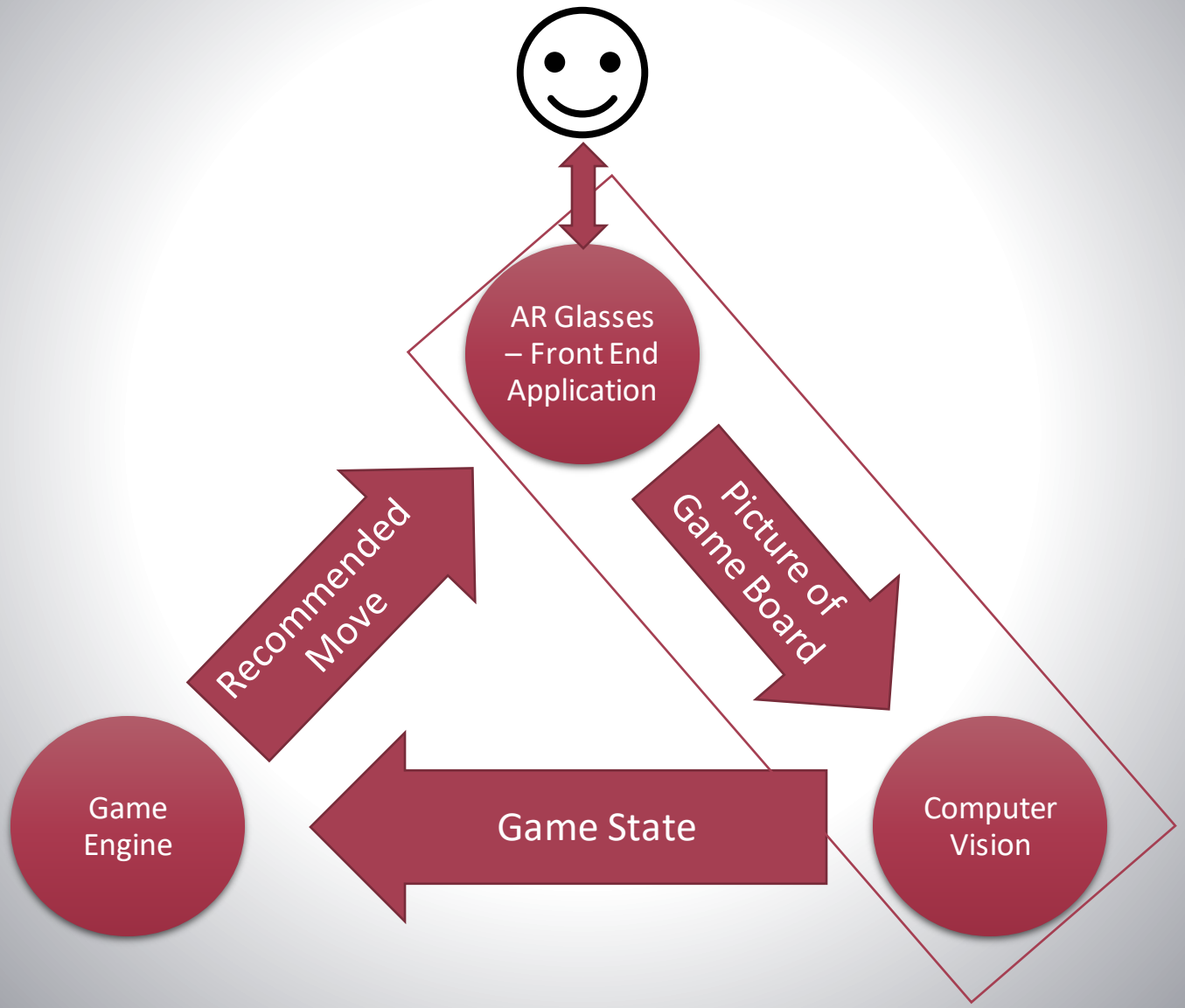
PROJECT VISION

Develop a creative and engaging way to improve strategy board game enthusiasts' skills at strategy board games through Augmented Reality and Computer Vision while increasing the excitement around these technologies for all people.

USE CASES

1. Play with the assistance of the glasses and chess engine to learn and develop one's skill
2. Play against the chess engine by using the glasses and chess engine to recommend moves for your “opponent”

CONCEPTUAL SKETCH



REQUIREMENTS AND CONSTRAINTS

Functional

- Detect Board State using AR Glasses
- Detect and Process Board State using computer vision
- Determine best move given current state using top game engines
- Display Recommended Move on AR Glasses

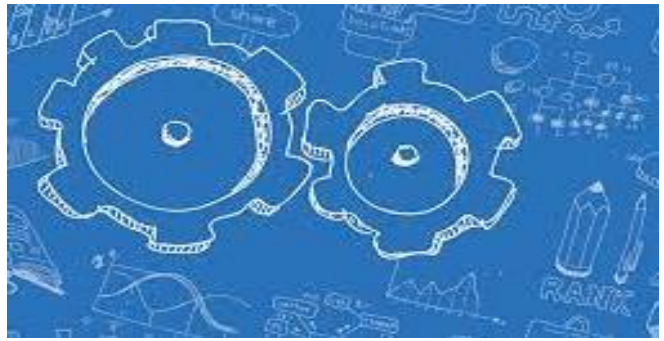
Environmental

- Game should be played in a well-lit room
- For best results area around chess board should be high contrast
- AR device should be kept in a clean, dry, and dust free environment
- Game should be played in a low dust environment to keep lens clear

Economic

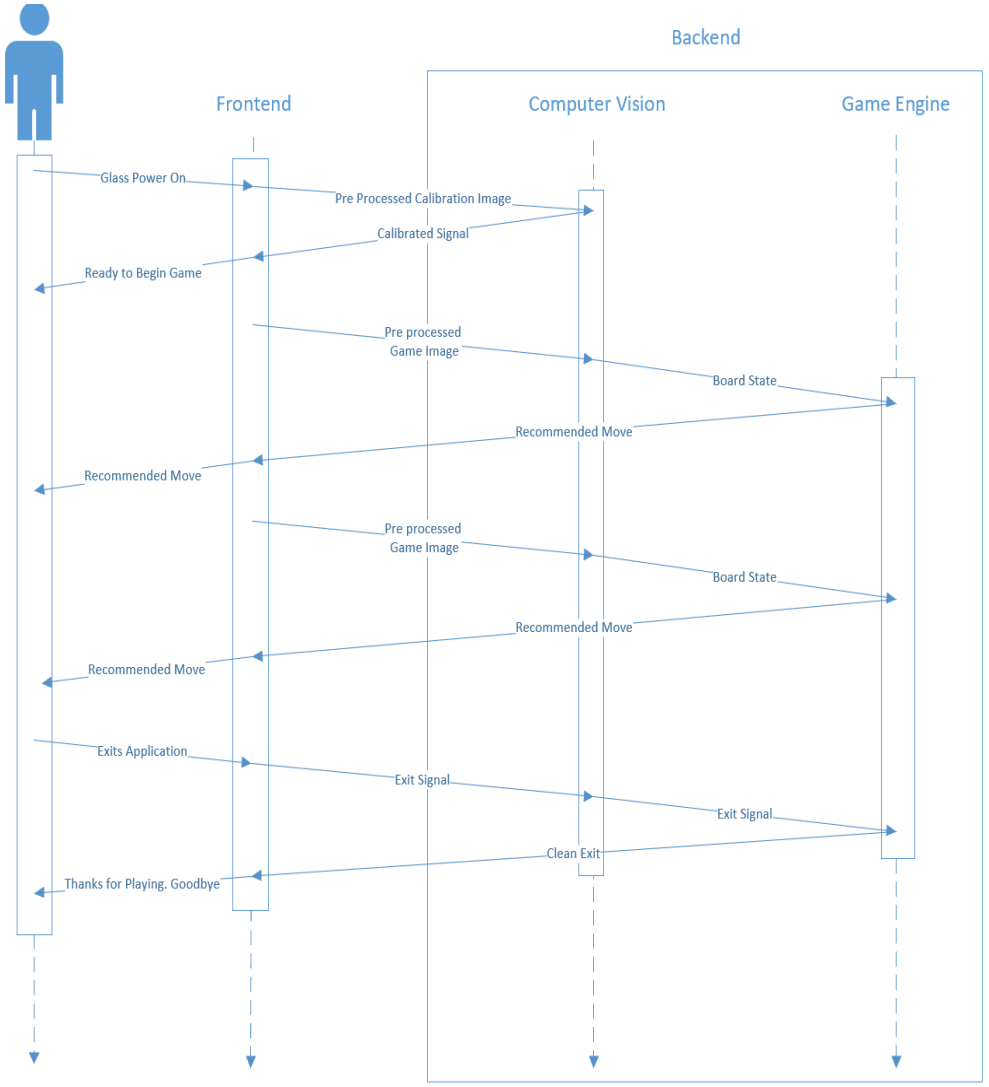
- Project should not exceed ~$1000 (Need Advisor Approval for over $300)
- Proof of concept should be completed no later than end of Spring Semester 2021
- Build when you can, buy when necessary

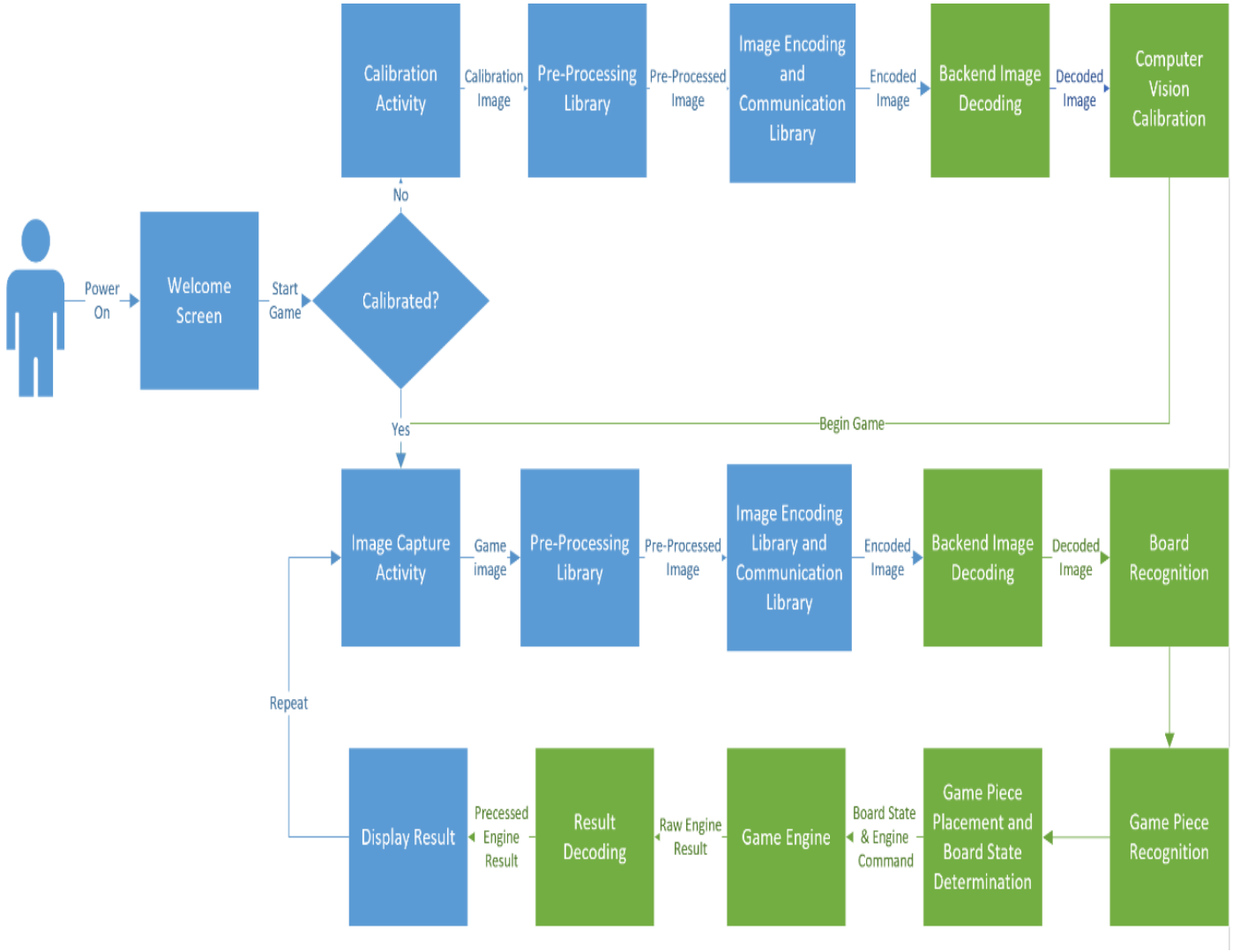
STANDARDS



- IEEE 730-2014: SOFTWARE QUALITY ASSURANCE PROCESSES
- IEEE 26515–2018: DEVELOPING INFORMATION FOR USERS IN AN AGILE ENVIRONMENT
- IEEE 12207–1996: SOFTWARE LIFE CYCLE PROCESSES
- IEEE 1028–2008: STANDARD FOR SOFTWARE REVIEWS AND AUDITS
- IEEE 16085–2006: SYSTEMS AND SOFTWARE ENGINEERING - LIFE CYCLE PROCESSES - RISK MANAGEMENT
- ISO 9241-210-2019: ERGONOMICS OF HUMAN-SYSTEM INTERACTION

CONCEPTUAL DESIGN DIAGRAM





RISKS AND MITIGATION

Risk

1. Board and piece detection quality/confidence may be too low to be viable for a release candidate
2. Project may be unable to pick up from an arbitrary game state

Mitigation

1. Temporarily restrict board design and assume images are taken in a near-optimal environment
2. Temporarily restrict chess game starting point

DESIGN – FUNCTIONAL DECOMPOSITION

AR Glasses

1. Capture Images
2. Preprocess Images
3. Communication
4. Display Recommended Move

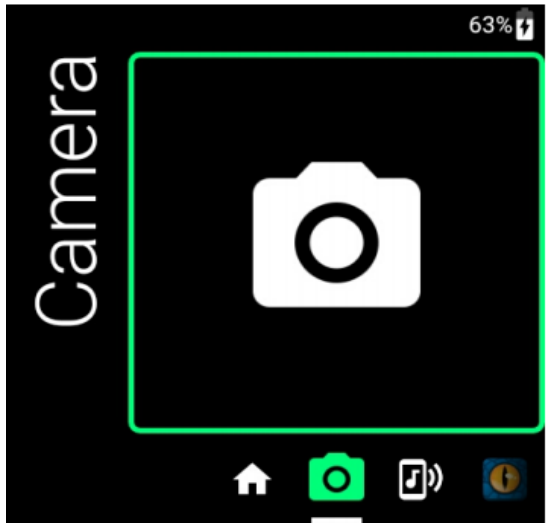
Computer Vision

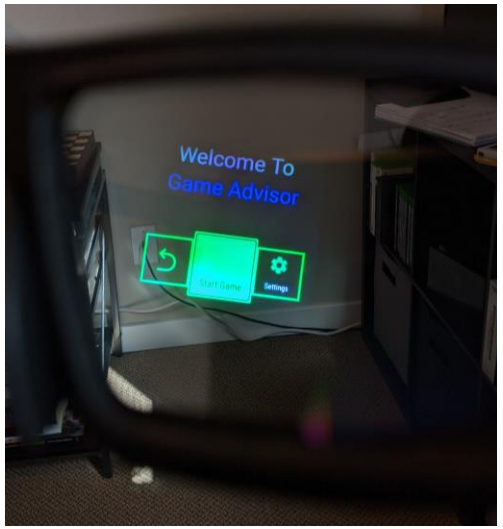
1. Detect Board
2. Identify Chess Pieces
3. Determine Board State
4. Calibration
5. Communication

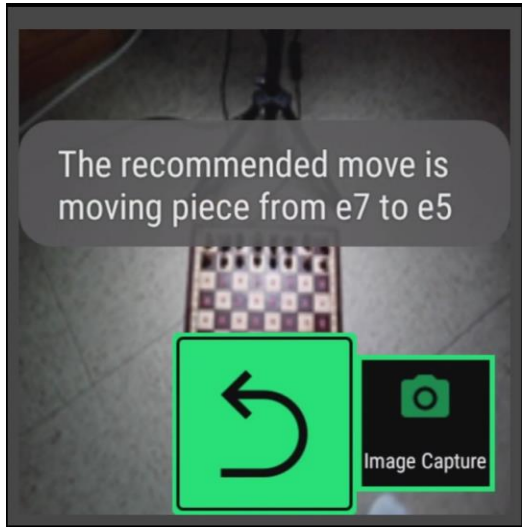
Game Engine

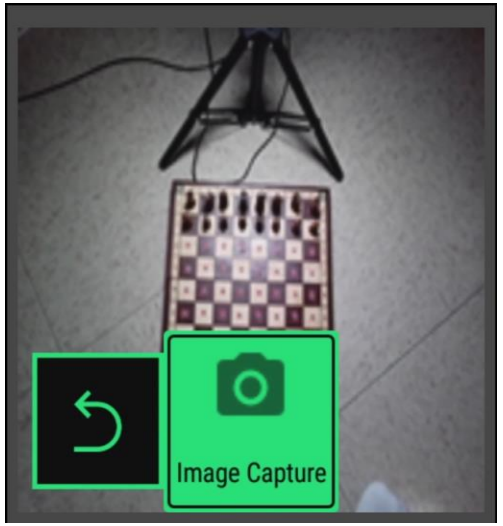
1. Calculate Best Move
2. Customization for time and difficulty
3. Communication
4. Result Decoding

USER INTERFACES



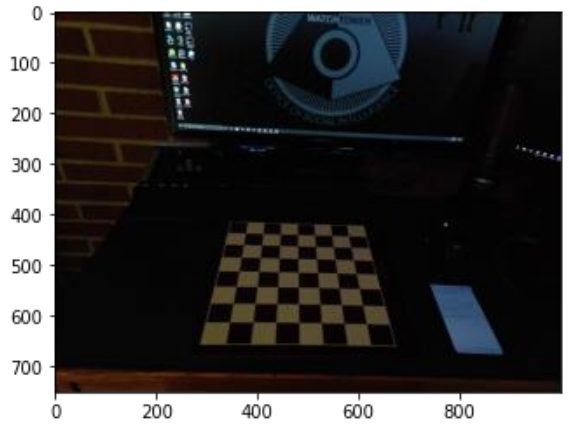


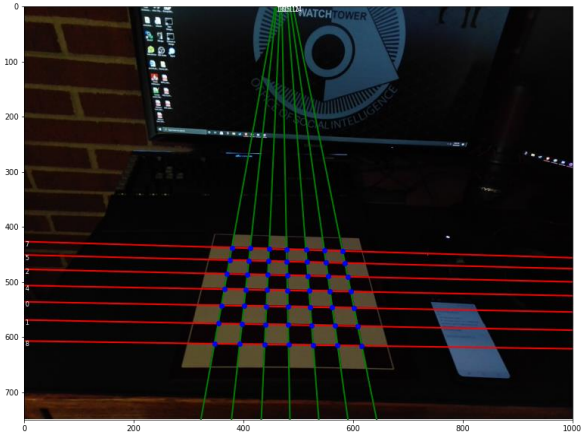




COMPUTER VISION ALGORITHM-FIRST PASS

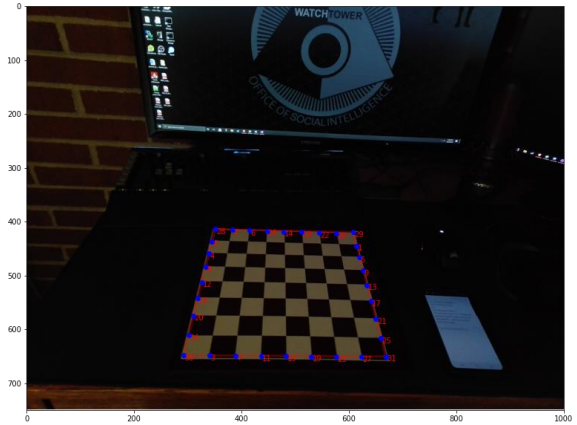
Start with image of the board.

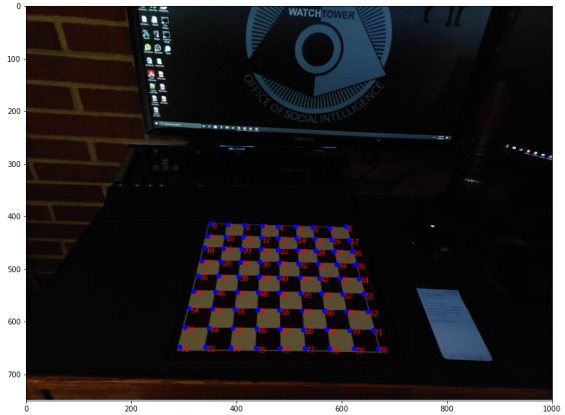




1. Detect lines with Hough Line Detection.
2. Group lines based on angle and confidence.
3. Find the intersections of the lines.

Extrapolate outer points using interior points.





Finish with all the chess square corners which we can segment for further processing.

DEMO VIDEO

UNIT TESTING

- Computer Vision Pipeline Using Set of >1000 Test Images
  - Corner Checks
  - Individual Square Checks
  - Accuracy Checks
  - Valid Move Checks
  - Individual Function Checks (Ex: get_board_diffs)
  - Move Edge Cases (Ex: En Passant)
- Android Application
  - Stockfish Library
  - Communication

```
PS C:\Users\txcyu\Desktop\CV\sdmay21-05\ComputerVision> python test_CV.py -v
test_getboarddiffs1 (__main__.TestCV) ... ok
test_getboarddiffs2 (__main__.TestCV) ... ok
test_getboarddiffs3 (__main__.TestCV) ... ok
test_getboarddiffs4 (__main__.TestCV) ... ok
test_getboarddiffs5 (__main__.TestCV) ... ok
test_getlastmove1 (__main__.TestCV) ... ok
test_getlastmove2 (__main__.TestCV) ... ok
test_getlastmove3 (__main__.TestCV) ... ok
test_getlastmove4 (__main__.TestCV) ... ok
test_getlastmove5 (__main__.TestCV) ... ok

----------------------------------------------------------------------
Ran 10 tests in 0.003s

OK
```

```
Found 4 corner circles!
Found 4 corner circles!
Found 4 corner circles!
image: test_imgs/corners/4k/4.jpg
AccuracyViolationError('Corner Check with 0.25 -> Failed to detect 4 corners!')
Found 4 corner circles!
Found 4 corner circles!
Found 4 corner circles!
Found 4 corner circles!
Found 4 corner circles!
Found 4 corner circles!
Found 4 corner circles!
```

INTERFACE AND ACCEPTANCE TESTING

- Interface Testing
  - Manual UX Testing
  - Communication Testing for the Move and Image
- Acceptance Testing
  - Level 1 - Combine Interfaces to Verify Functional Requirements
    - Passed at a prototype level
  - Level 2 - Full User Testing with Dr. Zambreno and other avid chess players
    - Never reached

LESSONS, SKILLS, AND KNOWLEDGE

- Importance of Modular Development
- Simplify the Problem and Work to Remove Assumptions Later – Computer Vision Pipeline
- Requirements Developments
- Engineering Standards
- Project Design
- Running Executable in Background of Mobile Application
- Computer Vision Basics using OpenCV
  - Line Detection
  - Point Detection
  - Pre-processing techniques
  - Color Masking
- Running Python on Android
- Running Executables in the background of an Android Application and giving said executable commands
- Speaking and Presenting a Technical Project

EXISTING CHALLENGES AND NEXT STEPS

Increasing Accuracy Detection

- Shadows and Glare Due to Lighting
  - Solution: Lighting Detection and Calibration of CV Module before starting the game
- Hidden Pieces (Ex: Queen Covers Pawn)
  - Depending on board state and changes, we can infer the hidden piece is still there
  - Retake Image
- Pieces don't cover the red dot completely
  - Refine how many pixels we look at to determine if space is occupied
- Problematic Pieces – Varying shades and reflective amounts

Removing Assumptions

- Green and Red Dots
- Non-Standard Chess Set
- Put on Glasses at any point in the game

Expand to Other Tabletop Games

THANK YOU

Dr. Zambreno

sdmay21-05@iastate.edu

AR Chess Advisor

TECHNOLOGY CONSIDERATIONS – AR GLASSES

	Google Glasses	Microsoft Holo Lens (Version 1 or 2)	Vuzix Blade
Price			
Functionality			
Team Fit			
Discrete			

TECHNOLOGY CONSIDERATIONS – COMPUTER VISION

	Open CV	MATLAB CV Toolbox
Library Size and Number of Available Algorithms		
Fit for our Team Members		
Available Support		
Industry Standard		

TECHNOLOGY CONSIDERATIONS – CHESS ENGINES

	Stockfish	Komodo	Cuckoo	Lenna Chess Zero	Shredder
Price					
Documentation & Examples					
Ease of Use