

Augmented Reality Chess Advisor

sdmay21-05

Team Members: Dillon Peters, Parker Bibus, Jamie Peterson, Jake Aunan, Aidan Sherburne, Brett Santema
Faculty Adviser and Client: Dr. Joseph Zambreno

Intended Uses and User

- Play with assistance of application to learn and/or cheat
- Play against application to test skill
- Glasses can be worn by any chess enthusiast

Professional Standards

- Software Quality Assurance Process Standards for Testing – IEEE Std 730-2014
- Software Reviews – IEEE Std 1028-2008
- Developing Information for Users in an Agile Environment – IEEE 26515 - 2018

Project Resources

- Vuzix Blade - Purchased (\$700)
- Traditional Chess Set - Purchased (\$20)



Simplify Assumptions

- Product is used from game start
- Colored markers on game board (to be removed by future team)

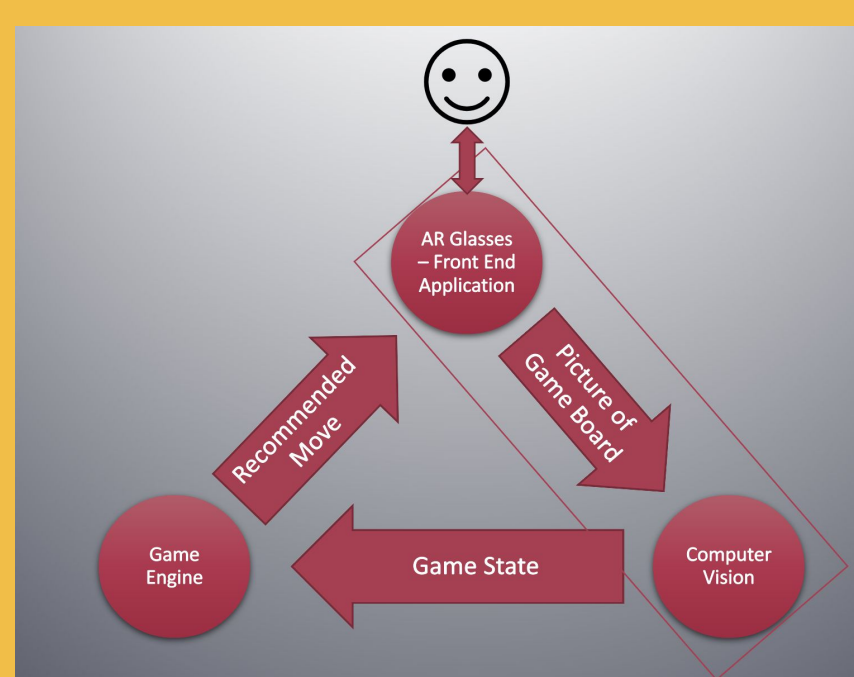
Intro/Motivation

Problem: Chess enthusiasts are always looking for ways to improve their skills. Current methods are limited to:

- Reading books/guides
- Studying game strategy
- Learning from skilled professionals
- Practicing against peers/bots

These methods are time consuming and also lack the levels of engagement and excitement many players desire to make them worthwhile, especially when it comes to younger players.

Solution: Increase engagement and excitement around learning chess and AR technology by developing an AR Glass Application to allow users to build skill while playing against or with the help of an AR game engine.



Design Requirements

Functional

- AR glasses device must be unobtrusive
- Game board state must be captured using the AR glasses
- Game board state must be processed on the device using computer vision
- Determine a recommended move using a game engine
- Indicate the recommended move to the user on the AR glasses display

Environmental/Operating Environment

- The board game should be played in a well-lit setting
- The game board should have high contrast with the surrounding environment
- The AR glasses should be kept clean, dry, and dust free for best results
- The board game should be played in a clear environment free of obstructions

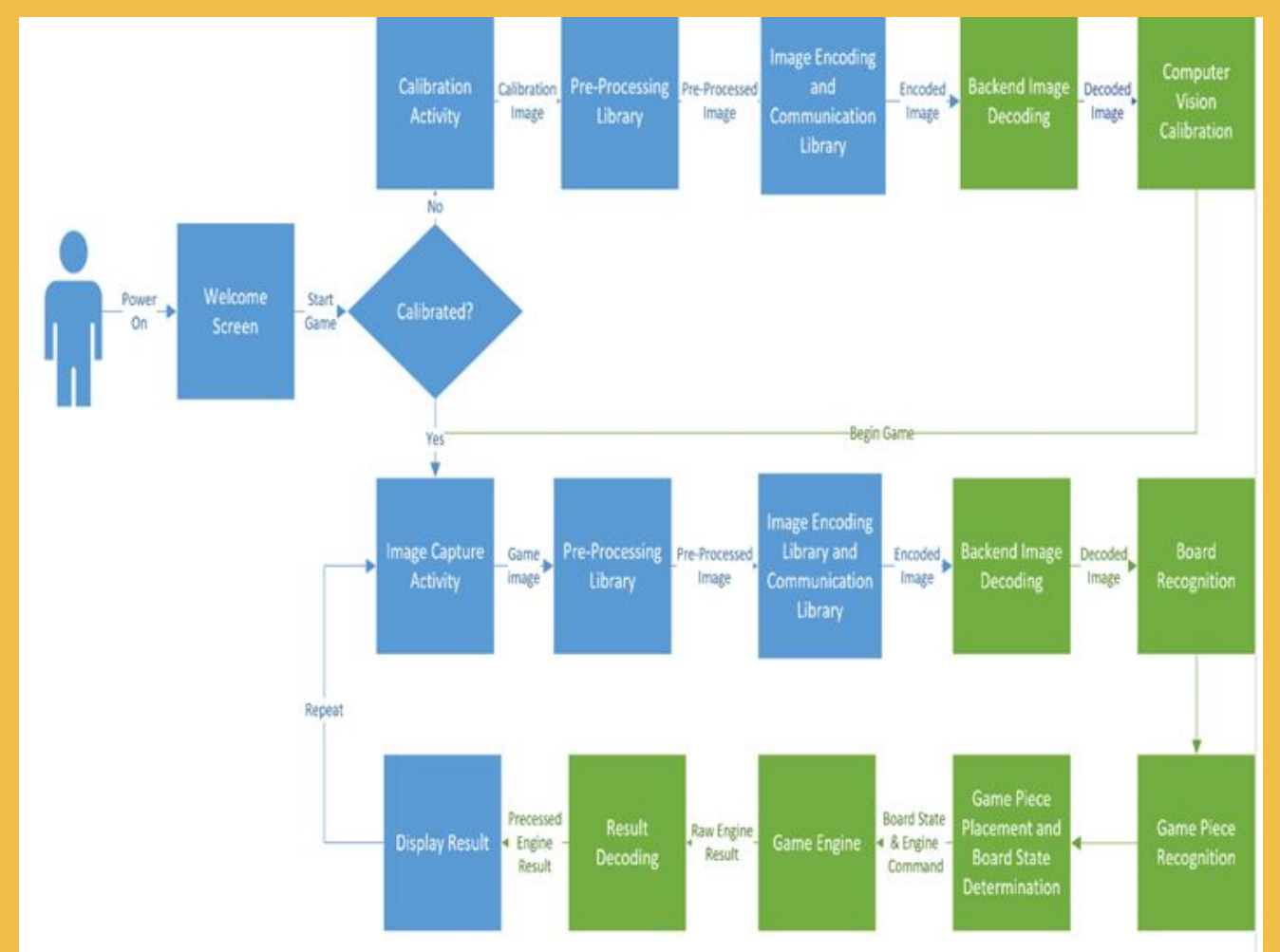
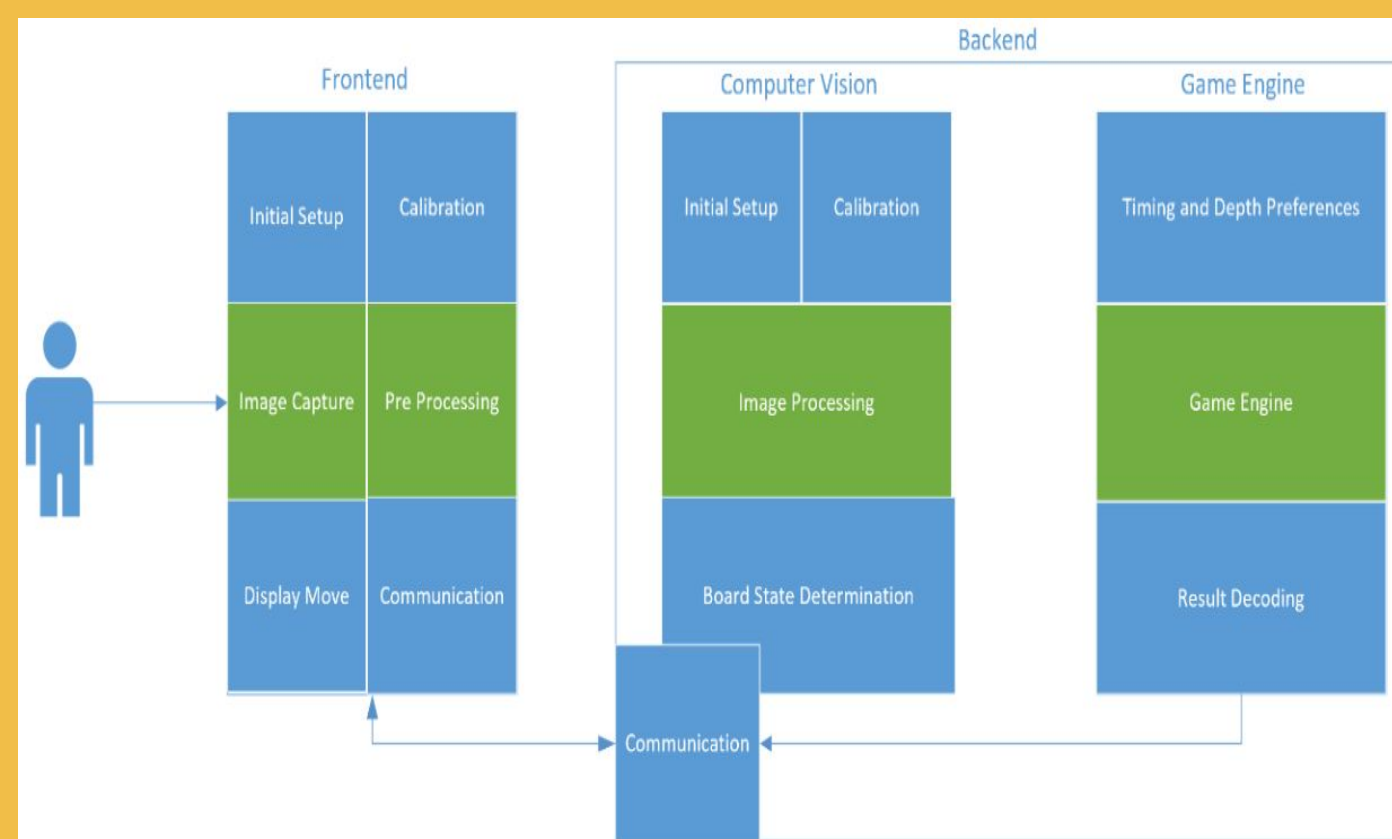
Economic

- The project must not exceed \$1000 in cost and adviser approval is required for expenditure in excess of \$300
- A proof-of-concept product should be completed by the end of the Spring 2021 semester
- "Build when you can, buy when necessary"



Design Approach

- Build with modularity in mind
 - Frontend UI/UX
 - Board & state recognition
 - Game engine integration
 - Move calculation
- Allows for easier iteration and implementation of new games
 - E.g. checkers, connect four
- Security
 - Modules easily audited
 - Little risk if compromised
 - Catastrophic failure results in failed gameplay at worst



Blue represents Front-End, Green represents Backend

Technical Details

Programming Languages

- Python
 - Computer vision
 - Board position detection
 - Piece position detection
 - Game engine integration
 - Move recommendations

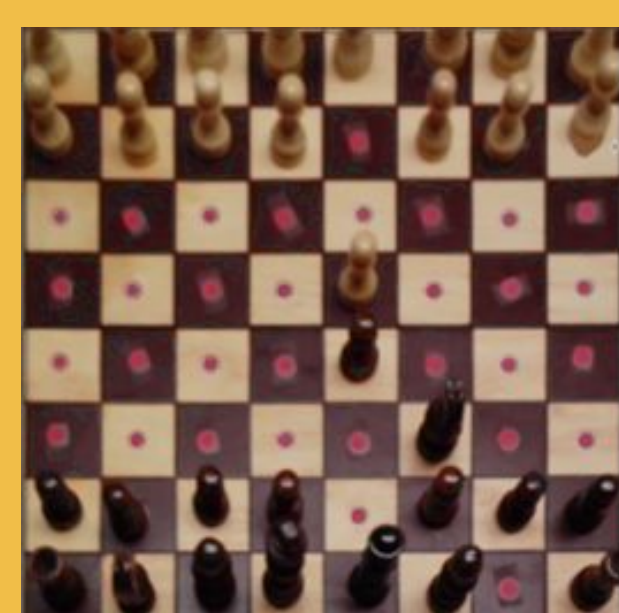
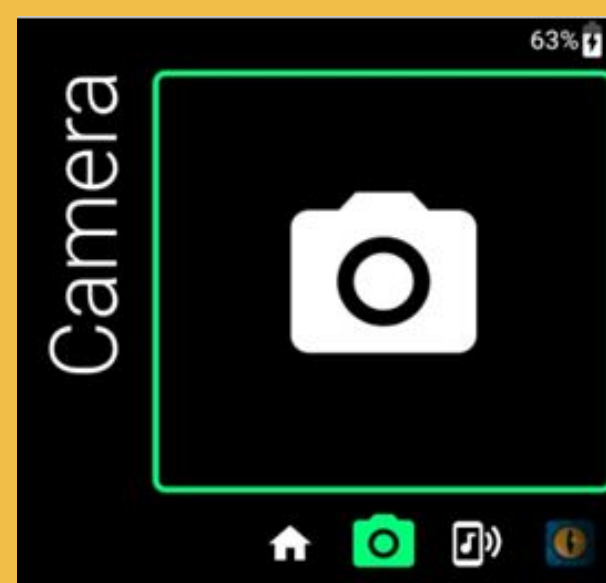
- Java
 - Android frontend
 - Image capture
 - User interface
 - Calibration

Libraries

- OpenCV
- Stockfish
- Chaquopy
- Numpy

Development Tools

- Jupyter Notebook
- Android Studio
- Vuzix Blade SDK
- Vuzix View
- Vuzix Emulator



```
Board Output:
[ [1, 1, 0, 0, 0, 0, -1, -1],
  [1, 1, 0, 0, 0, 0, -1, 1],
  [1, 1, 0, 0, 0, 0, -1, -1],
  [1, 1, 0, 0, 0, 0, -1, -1],
  [1, 0, 0, 1, -1, 0, 0, -1],
  [1, 1, 0, 0, 0, -1, -1, -1],
  [1, 1, 0, 0, 0, 0, -1, 0],
  [1, 1, 0, 0, 0, 0, -1, -1]]
1 = Light piece
0 = Open Spot
-1 = Dark piece
Access is [column][row]
```

Best Move: A7 ⇨ B7

Testing

Testing Image Set Generation

- Over 1000 images
- Variety of:
 - Lighting Conditions
 - Angles
 - Board States

Automated unit testing

- Android Application
- Computer Vision Pipeline
 - Corner Checks
 - Individual Square Checks
 - Accuracy Checks
 - Valid Move Checks
 - Individual Function Checks (Ex: get_board_diffs)
 - Move Edge Cases (Ex: En Passant)

Interface Testing

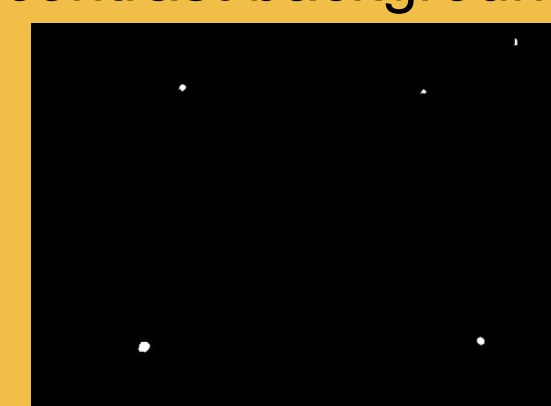
- Manual UX Acceptance Testing
- Communication Testing for Image and Move

User Acceptance Testing

- Level 1 - Combine Interfaces to Verify Functional Requirements
- Level 2 - Full User Testing with Dr. Zambreno and other avid chess players

Testing Environment

- Matches Operational Environment - Well lit area and clean with high contrast background



So far our results are consistent with our requirements for a proof-of-concept product