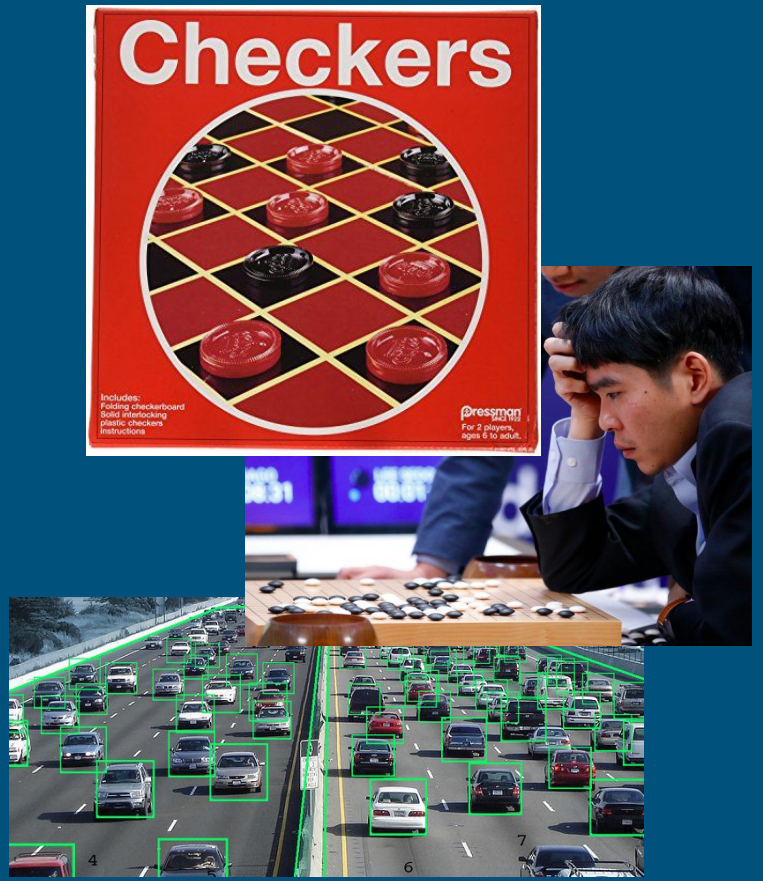
Check Yourself

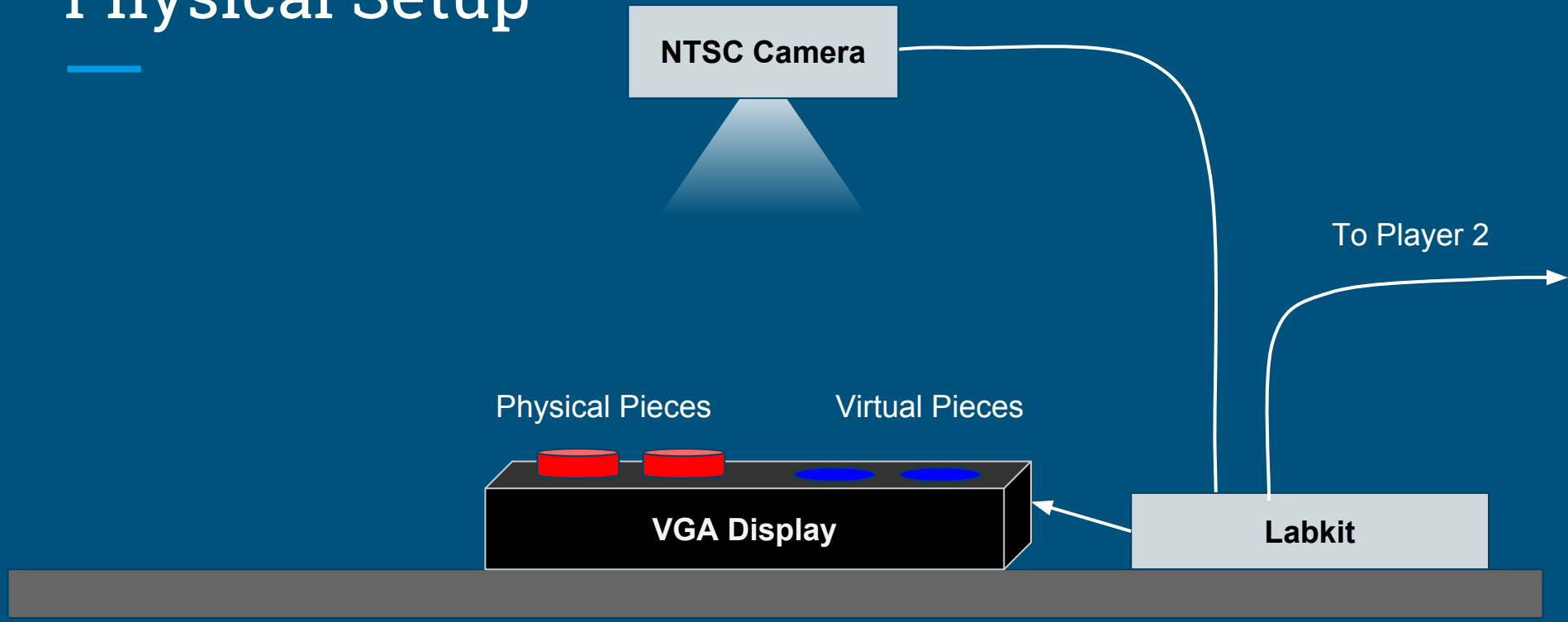
August Trollbäck, Elijah Stanger-Jones, and Suzanne O’Meara

Overview

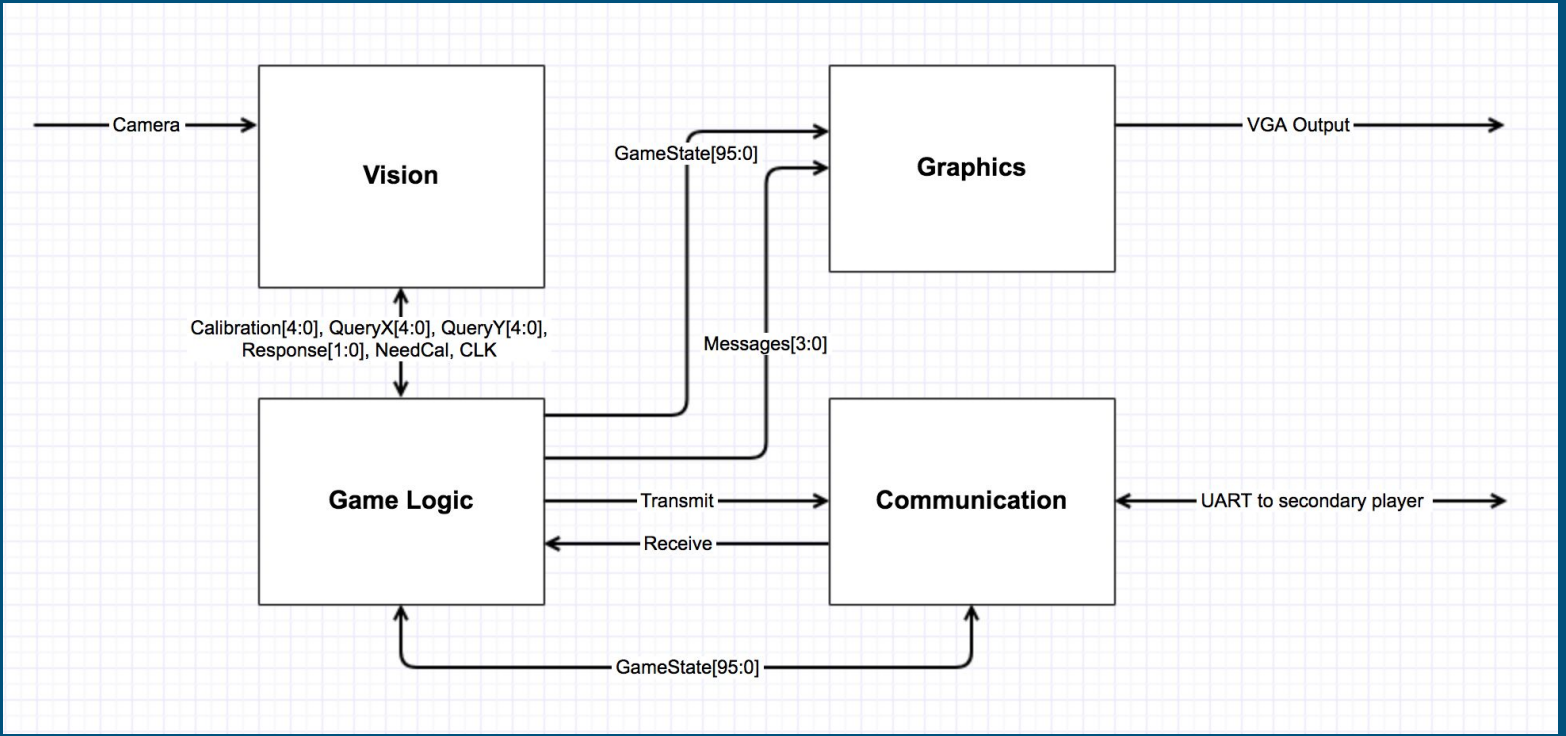
- Two-player augmented reality checkers
  - Each player has physical checkers pieces
  - Opponent's pieces displayed on monitor
  - AI to check rules and provide guidance
- Technology
  - Computer vision to read the physical piece locations
  - UART to send data between players
  - Computer monitor display



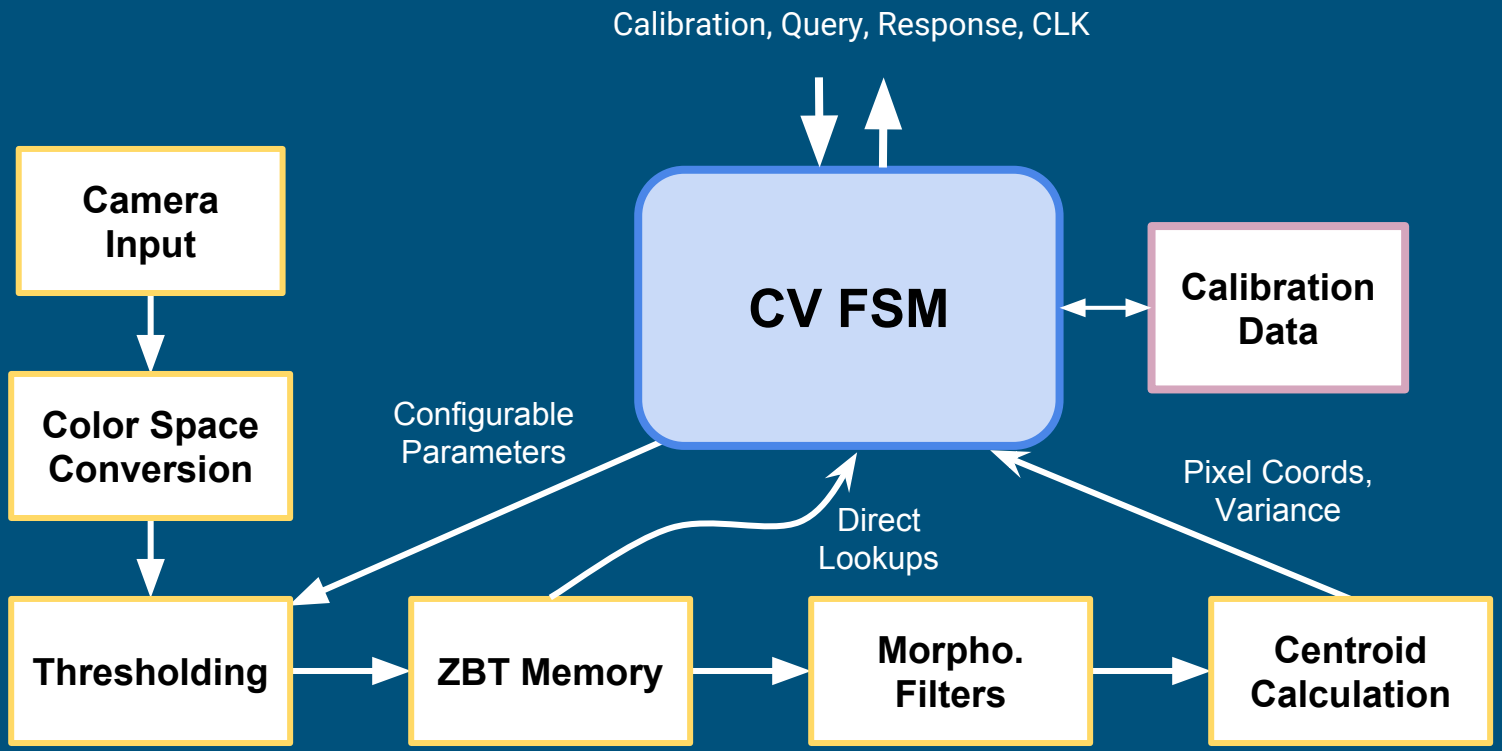
Physical Setup



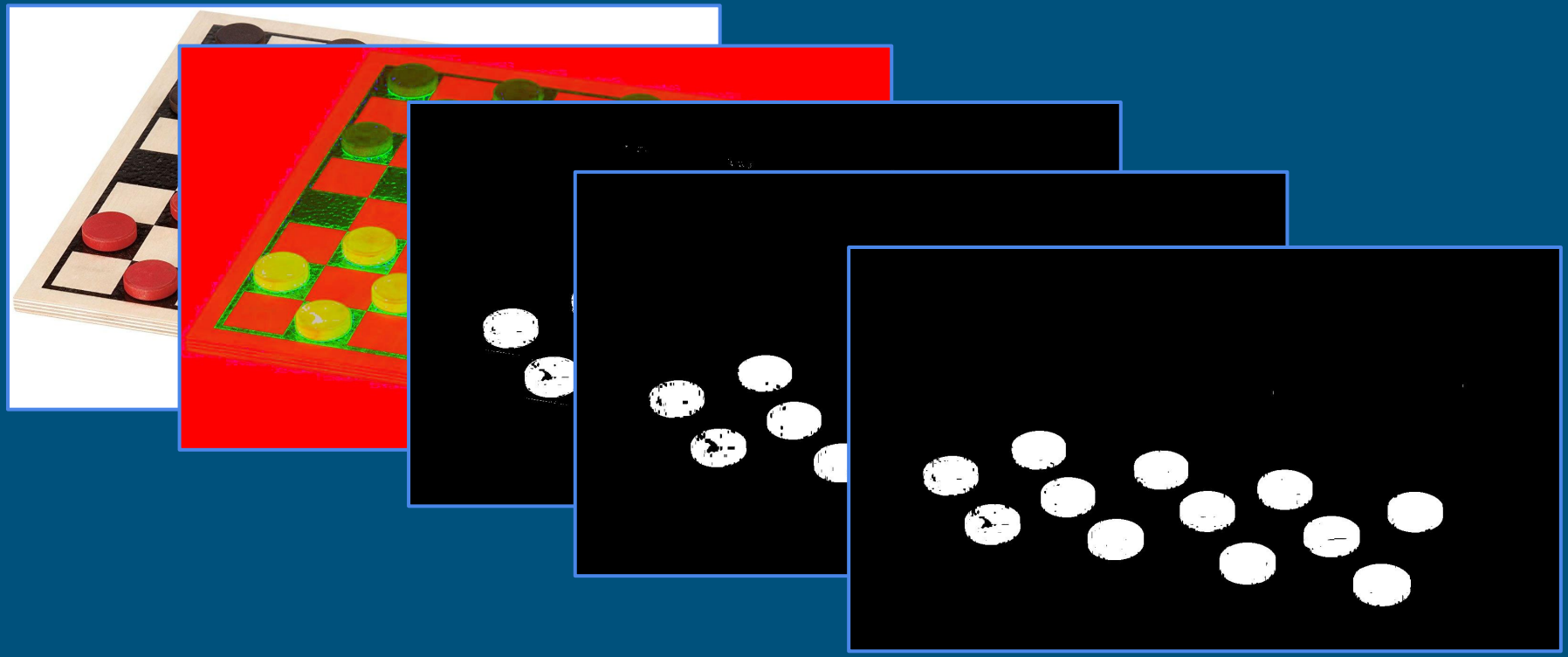
High Level Block Diagram



Computer Vision Pipeline



Computer Vision Pipeline

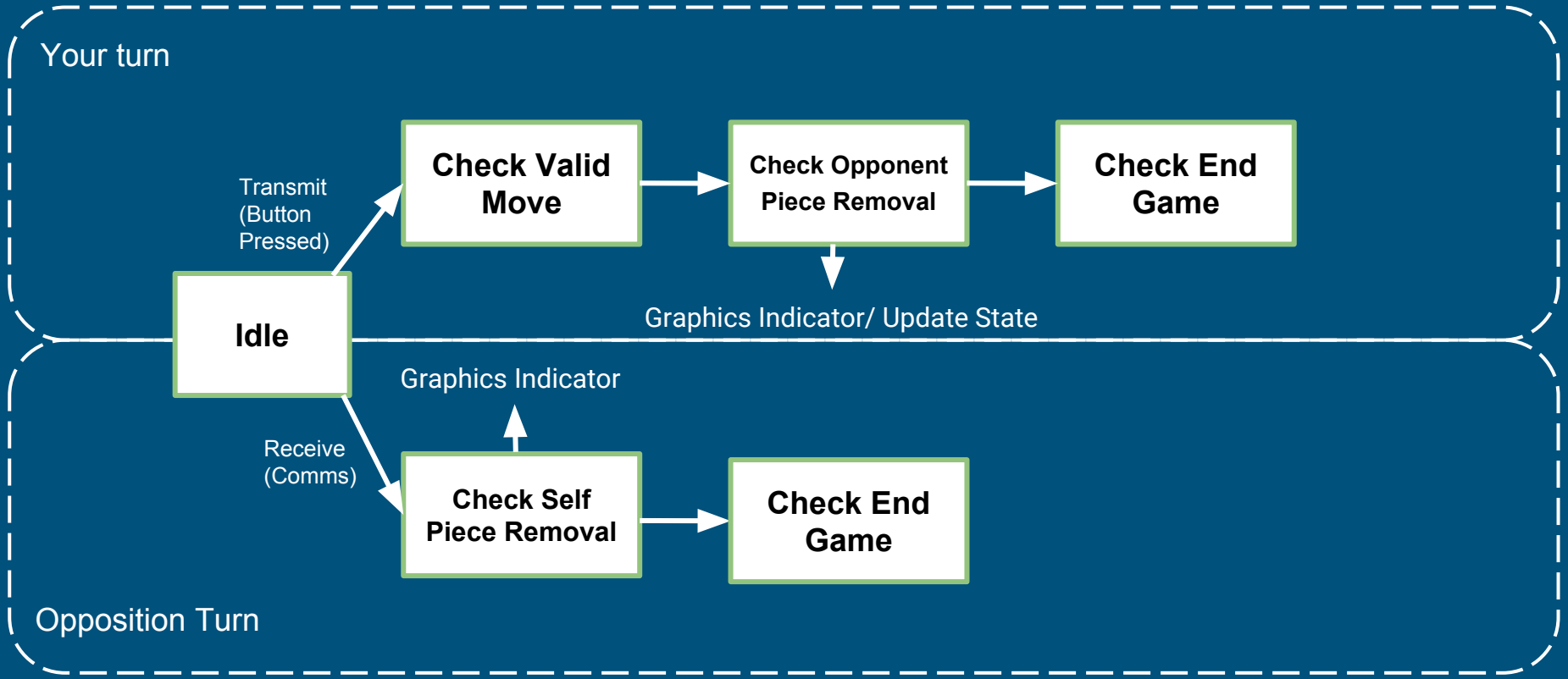


Computer Vision

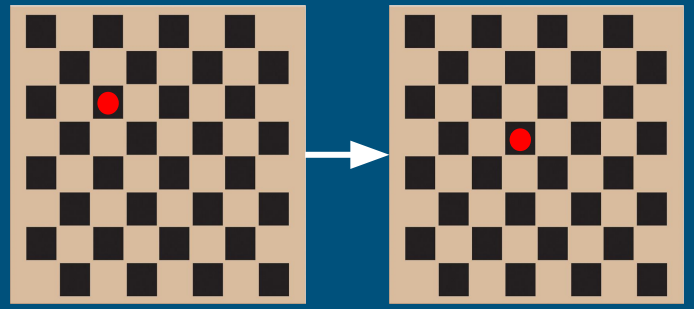
Stretch Goals

- Automatic move detection
- Account for camera distortion

Game Logic - High Level



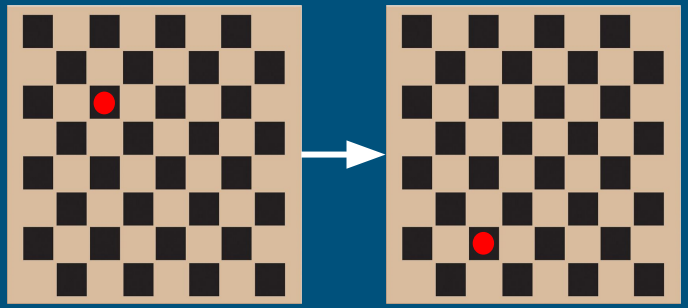
Game Logic - Detecting Valid Moves



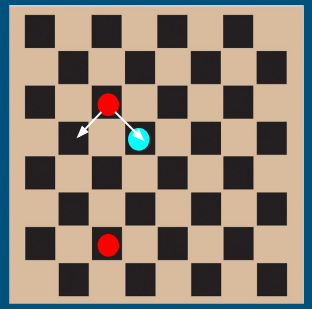
1. Detect the change:
    a. XOR the previous state with new state to produce:
    b. If more than 1 piece has moved then invalid.
2. If moved backwards then invalid.
3. Check simple non jump move
    a. i.e is the piece in the one or two valid locations for a non jump move
4. Iteratively check jump moves
    a. Check the first two valid locations
    b. If there is an opponent piece between the two positions, then valid and piece to be removed.
    c. If there is an opponent piece but a blank space instead then check the next two valid locations.
    d. Repeat to a depth of 3: the most pieces a standard checker can remove in one move

Stretch Goal: King piece logic, implement “must jump” rule

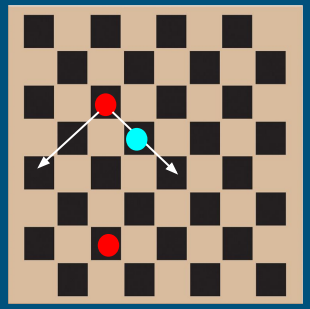
Example Jump Detection



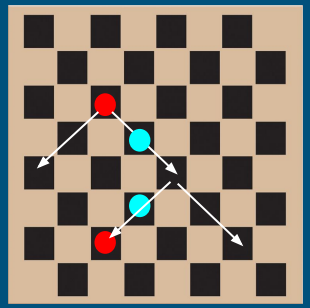
State Change



1. Simple Check



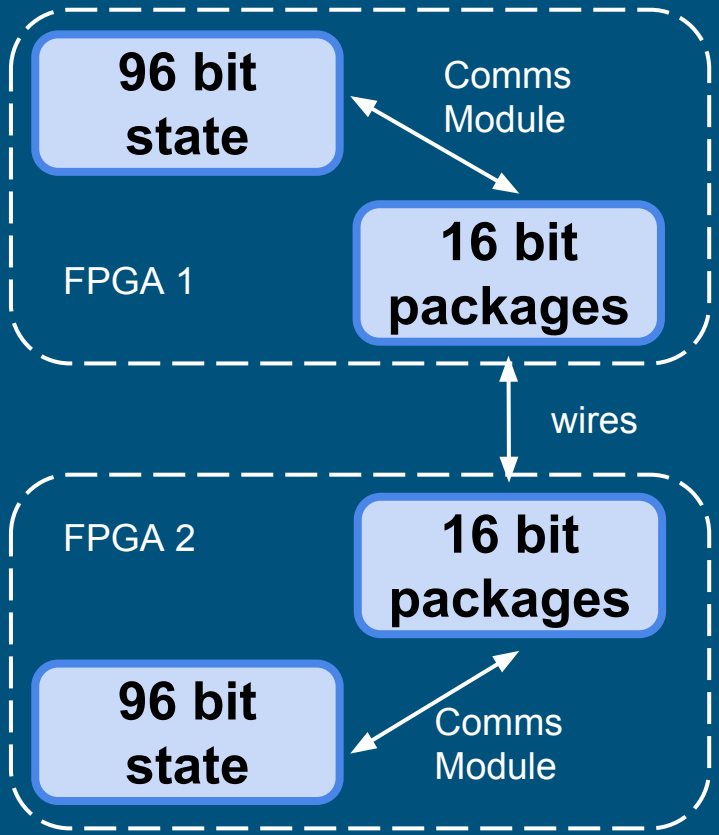
2. Depth 1 Check



3. Depth 2 Check

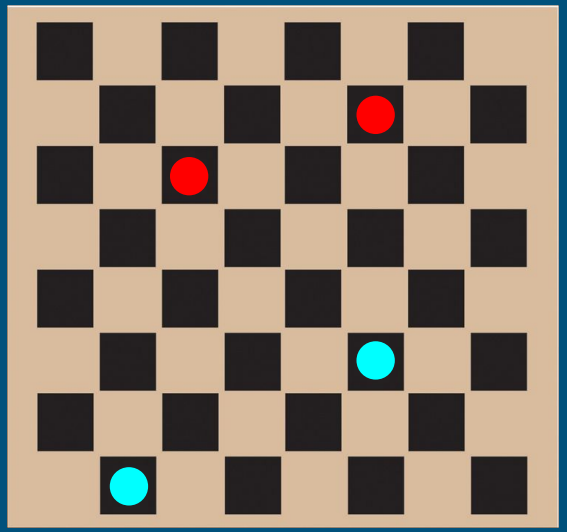
Communications

- Basic
  - UART
  - Parse custom data structures 96 bit variable into 16 bit packages
  - Data Ready bits
- Stretch
  - CRC, FEC, or other data integrity measures
  - Handshake between Game Logic and Comms Modules



Graphics- Game Board

- Basic
  - Continuously monitor game state variable which has location of each piece
  - Display basic board state
- Stretch
  - Animated piece motion
  - Better graphics on board and pieces
  - Nice theme



Graphics- Messages

- Basic
  - Continuously monitored 4-bit variable selects messages
  - Display on top of game board
- Stretch
  - Animated error messages
  - Implement timing within the graphics module

Timeline

Week	Task
11/5	FSMs complete, start coding
11/12	Finish individual code
11/19	Integrate components
11/26	Testing and adding features
12/03	Stretch goals

Any Questions?