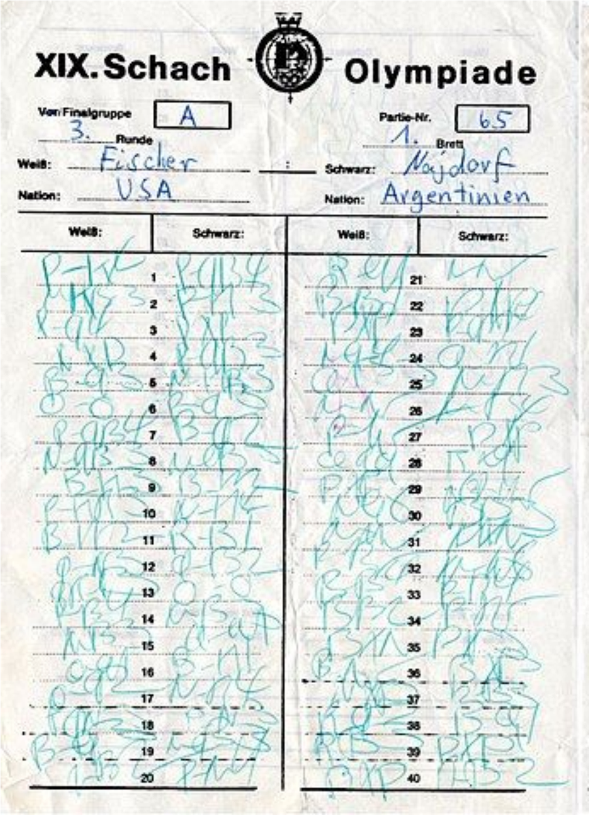
Team B0: Seamless Autonotator

Ryan Guan, Patrick Joyce, Vikram Marmer

Problem Statement/Use Case

- Notation is the way chess moves are recorded (Nc3, Bxf6)
- Notation is useful
  - Legality check in the case of a dispute
  - Reviewing previous chess games for improvement
- But, fraught with errors and time-consuming
  - Handwriting
  - Forgetting to notate
- Solution: Create a system that makes it easier for chess players to notate
  - Applicable to both professional and casual players
  - Would allow casual players to improve more



XIX. Schach Olympiade

Vor/Finalgruppe A

3. Runde

Partie-Nr. 65

1. Brett

Weiß: Fischer

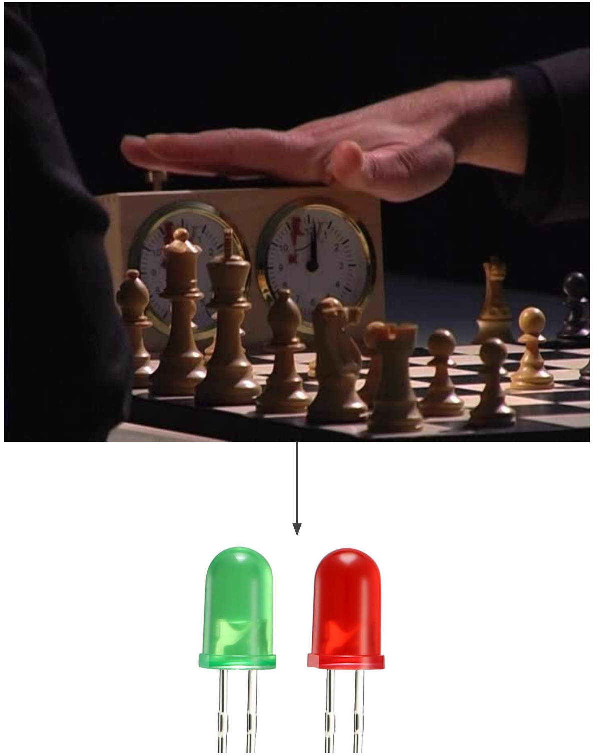
Schwarz: Najdorf

Nation: USA

Nation: Argentinien

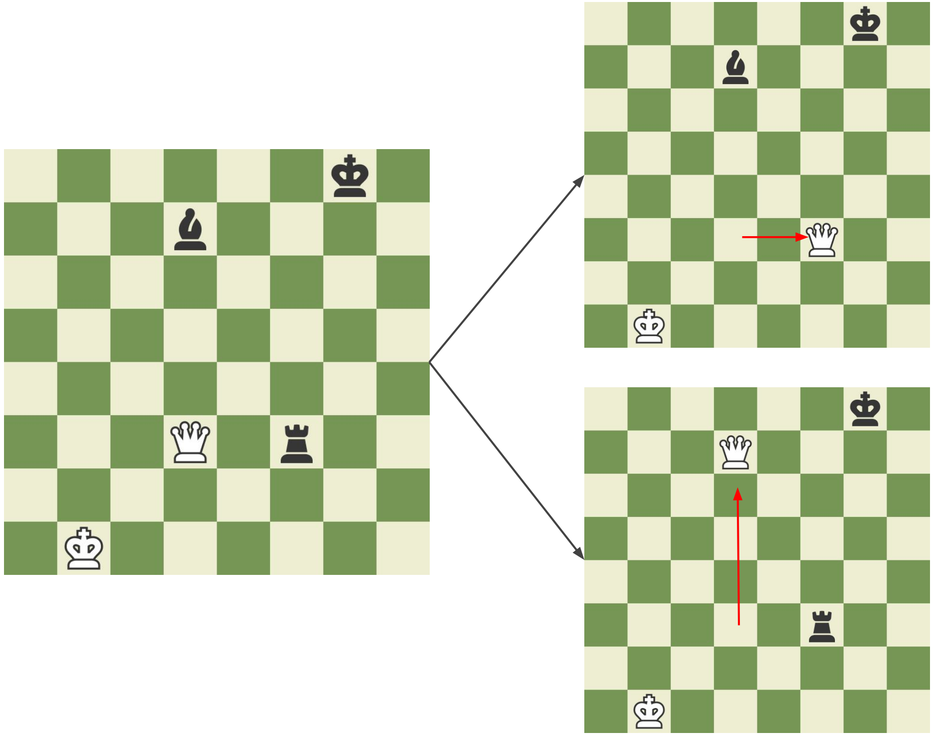
Requirements-Accuracy and Latency

- Accurately record notation
  - Aiming for **100%** accuracy
    - If inaccurate, the whole system fails
  - Inaccurate notation could result in a false positive for an inaccurate board state
- Latency from move input to legality LED response
  - **300ms** or less (rough human reaction time)
  - Important that lag is not detected through user testing



Requirements-Piece Detection

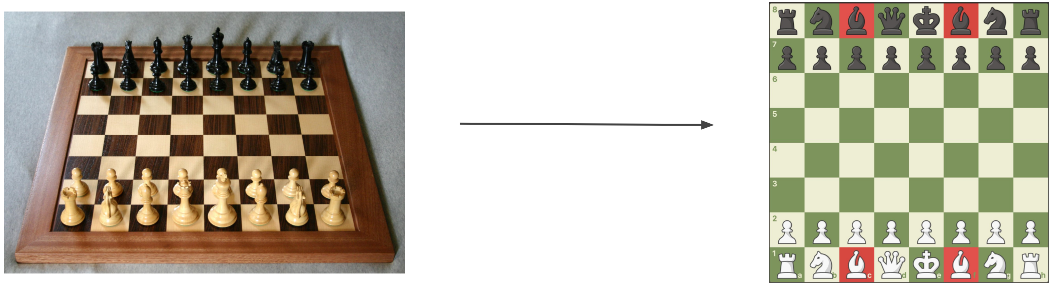
- Distinguish between white piece, black piece or no piece on a square.



- We know the starting position.
- We only need to distinguish between empty squares, white pieces, and black pieces.
- Then we can identify any move of one piece at a time.

Requirements-User Experience

- The board should be standalone and not need a camera or wall outlet
  - Camera/Wall outlet = more setup hassle and logistical issues
  - 10 hours of play time = ~1 day of high level tournament play
- Display information to the website with less than (2 seconds) of delay
- Access the previous 10 games (stored in the database) through the website
- Exportable to Chess.com

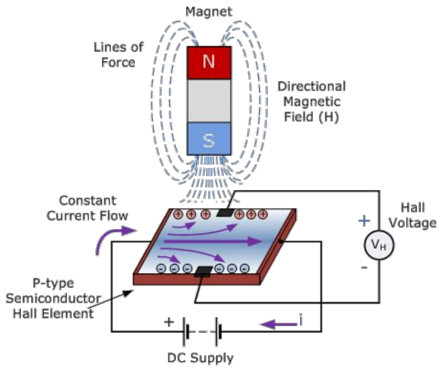


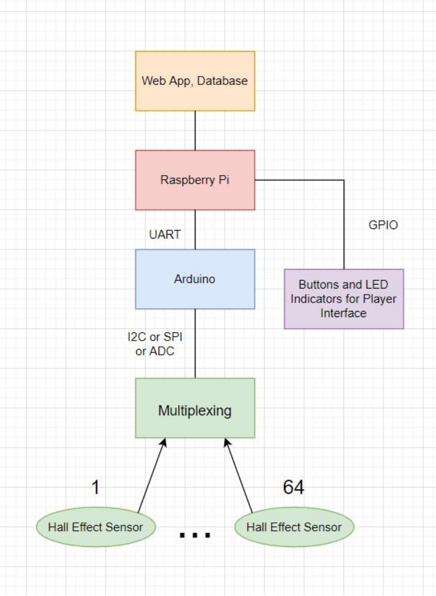
Technical Challenges

- Distinguishing between a white piece and a black piece
- Latency in differentiating pieces and legality checking should be lower than detectable human reaction time
  - Data collection from sensors, legality check, state logic
- Keep track of a correct board state
- Allow the board to be conveniently powered in a casual or competition setting

Solution Approach

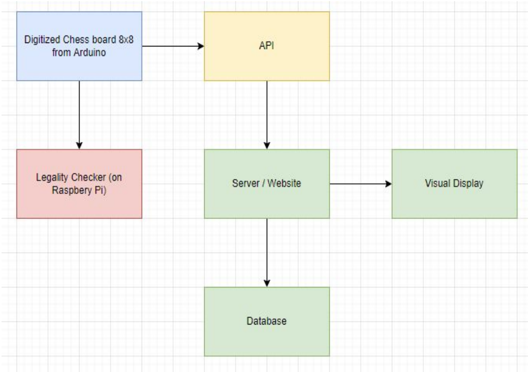
- Use ratiometric Hall-effect sensors for piece detection
  - Strength of output varies with strength of magnetic field.
  - Moderate strength magnets in white pieces, high strength magnets in black.
- Use arduino to convert analog sensor output to digital.
  - Will use the internal ADC or I^{2}C or SPI to communicate with sensors
  - Arduino will take under 15 ms to acquire and send all data
  - 2x safety factor
    - Data will be at RPi in under 30 ms
    - 90% of 300ms goal time still left for software





Solution Approach

- C++ Legality check program stores current board state in matrix.
  - Generates list of all legal moves and stores them in notation array
  - Translate new board state to move notation and check against legal moves.
  - Return result to user via green/red LED on board.
- State machine latency
  - Legal moves are pre-generated, reducing latency from user's perspective.
  - Receiving, translating, and checking a move should take less than 0.1 second with RPi 4's 1.5GHz clock.
- Use Python to write/read from databases and Web Application Backend



Digitized Chess board 8x8 from Arduino

API

Legality Checker (on Raspbery Pi)

Server / Website

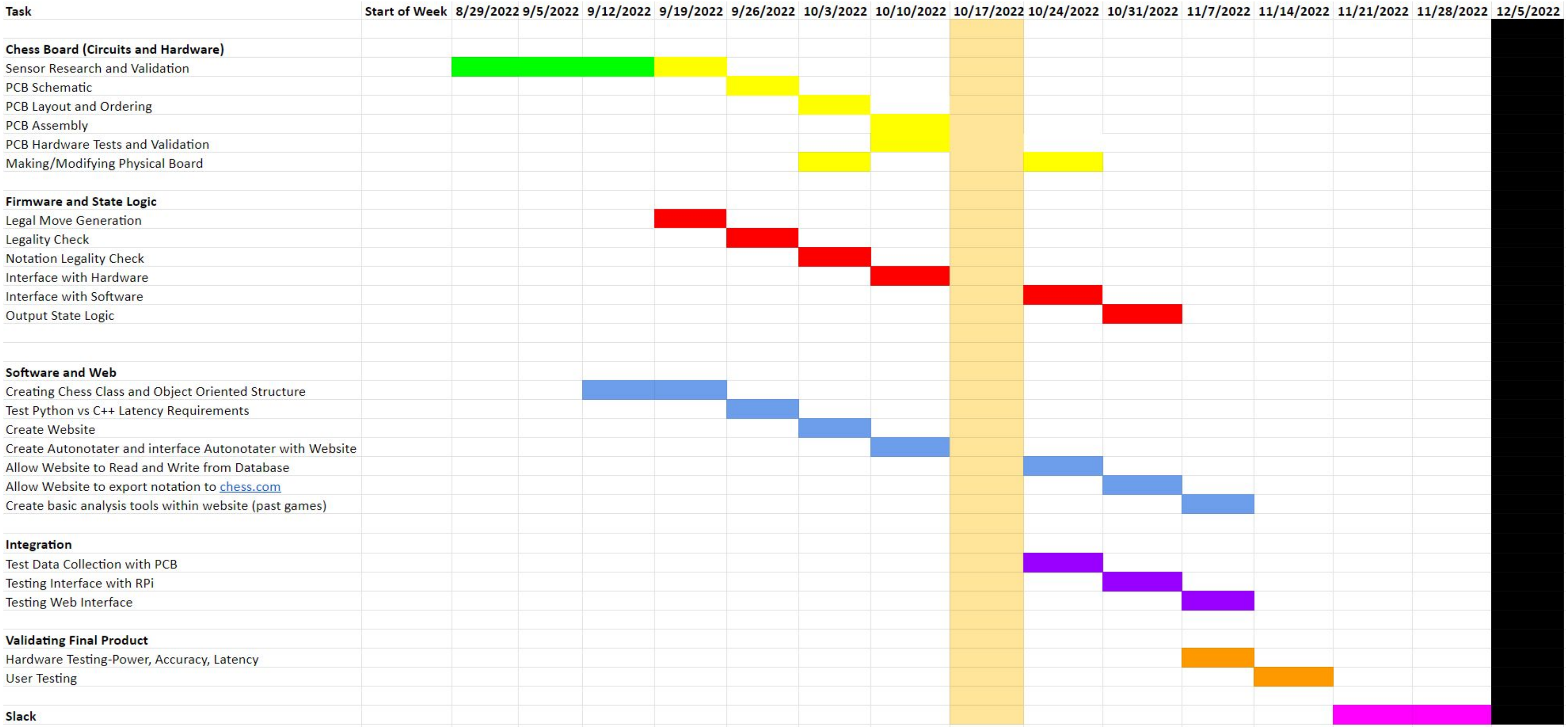
Visual Display

Database

Testing, Verification, Metrics

- Two types of user-testing
  - Initial/Informal User Testing
    - Used for testing quantitative metrics such as legality checking latency and human reaction time
  - Formal User Testing
    - Evaluate success in reducing latency
      - “Do you feel any lag between playing your move and noticing the legality checker?
      - “On a scale of 1-5 does the experience feel cohesive and streamlined”
  - Test various speeds of gameplay for accuracy
- Quantitative Tests
  - Accuracy of Board State & Piece Detection
  - Latency Tests
    - Data collection, user input to visual output and website update
  - Power Use
    - At idle and at peak computation

Schedule



Task	Start of Week	8/29/2022	9/5/2022	9/12/2022	9/19/2022	9/26/2022	10/3/2022	10/10/2022	10/17/2022	10/24/2022	10/31/2022	11/7/2022	11/14/2022	11/21/2022	11/28/2022	12/5/2022
Chess Board (Circuits and Hardware)																
Sensor Research and Validation		🟩	🟩	🟩	🟨											
PCB Schematic						🟨										
PCB Layout and Ordering							🟨									
PCB Assembly								🟨								
PCB Hardware Tests and Validation								🟨								
Making/Modifying Physical Board							🟨			🟨						
Firmware and State Logic																
Legal Move Generation					🟥											
Legality Check						🟥										
Notation Legality Check							🟥									
Interface with Hardware								🟥								
Interface with Software										🟥						
Output State Logic											🟥					
Software and Web																
Creating Chess Class and Object Oriented Structure				🟦	🟦											
Test Python vs C++ Latency Requirements						🟦										
Create Website							🟦									
Create Autonotater and interface Autonotater with Website								🟦								
Allow Website to Read and Write from Database										🟦						
Allow Website to export notation to chess.com											🟦					
Create basic analysis tools within website (past games)												🟦				
Integration																
Test Data Collection with PCB										🟪						
Testing Interface with RPi											🟪					
Testing Web Interface												🟪				
Validating Final Product																
Hardware Testing-Power, Accuracy, Latency												🟧				
User Testing													🟧			
Slack														🟪	🟪	

Division of Tasks

Vikram

- Sensor choice, Circuit & PCB design, assembly, debugging, validation

Patrick

- Magnet Research, Firmware, interface Arduino with RPi, legality check, control/state transition logic and outputs

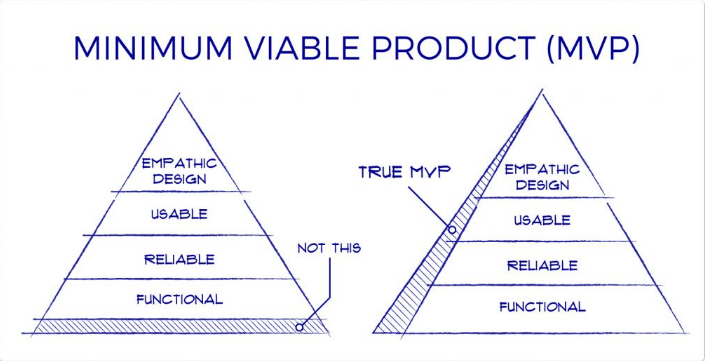
Ryan

- Web Application (HTML & CSS, JS, Server), database, auto-notation

All Group Members

- Chess board mechanical, Integration Work

Conclusion



- Autonotator (MVP)
  - Detect and record all previously played moves
  - Avoid notating illegal moves
    - Display legality through red and green LED
  - Basic website that displays notation of current game
- ECE Areas: Software and Circuits