Team A0: Tactile Chess

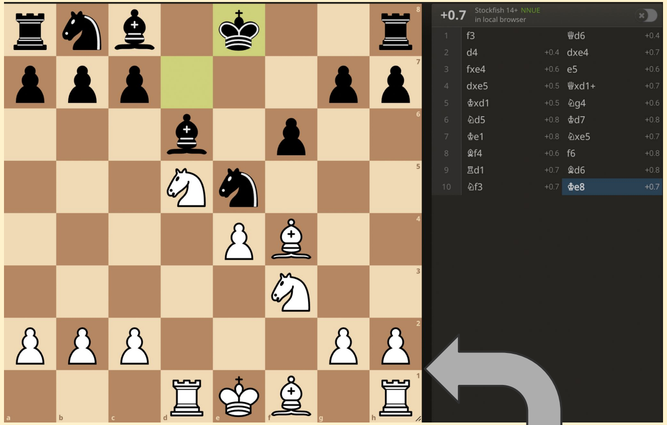
Mukundh Balajee, Edison Aviles, Juan Mejia

Introduction & Project Summary

- Create a smart chess board that allows the visually impaired to play online chess, as well as having the ability to train beginner and novice chess players
- Provide tactile and vocal cues to allow players a quick transition between moves of the physical board and the online platform.
- With the use of our own web application, players will be able to connect the board to the online platform, such as lichess.org and make moves that are reflected in the online game and recorded for players to study later.
- ECE Areas: Signal Processing, Circuits, Software Systems

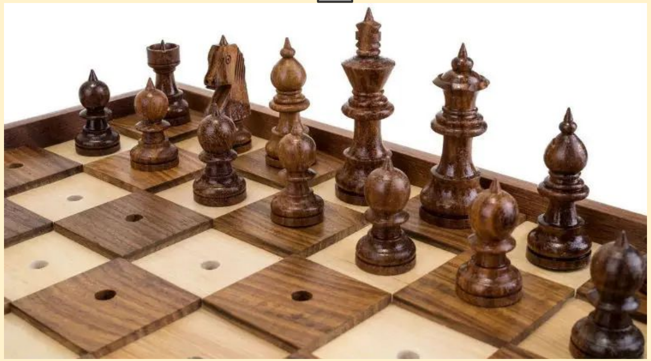
Case Requirements - User Experience and Accessibility

- Accessibility
  - Main focus for our project is to achieve a smart board that is accessible to the community of visually impaired chess players.
- Easy setup
  - Users should take about 1 minute to set up and connect the board.
- Battery life
  - Provide users a play time of 4 hours (not plugged in)
- Online Play
  - Connect to a popular online platform in order to play chess online and register moves with max 2 seconds delay.



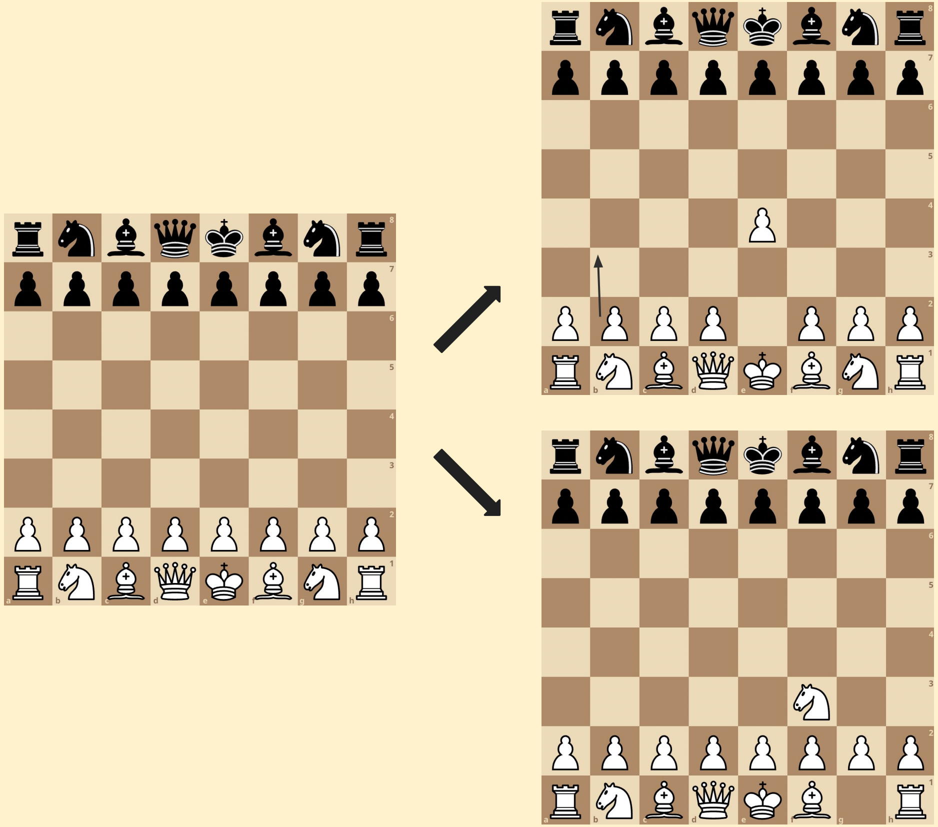
+0.7 Stockfish 14+ NNUE
in local browser

1	f3		♕d6	+0.4
2	d4	+0.4	dxe4	+0.7
3	fxe4	+0.6	e5	+0.6
4	dxe5	+0.5	♕xd1+	+0.7
5	♔xd1	+0.5	♘g4	+0.6
6	♘d5	+0.8	♔d7	+0.8
7	♔e1	+0.8	♘xe5	+0.7
8	♗f4	+0.6	f6	+0.8
9	♖d1	+0.7	♗d6	+0.8
10	♘f3	+0.7	♔e8	+0.7



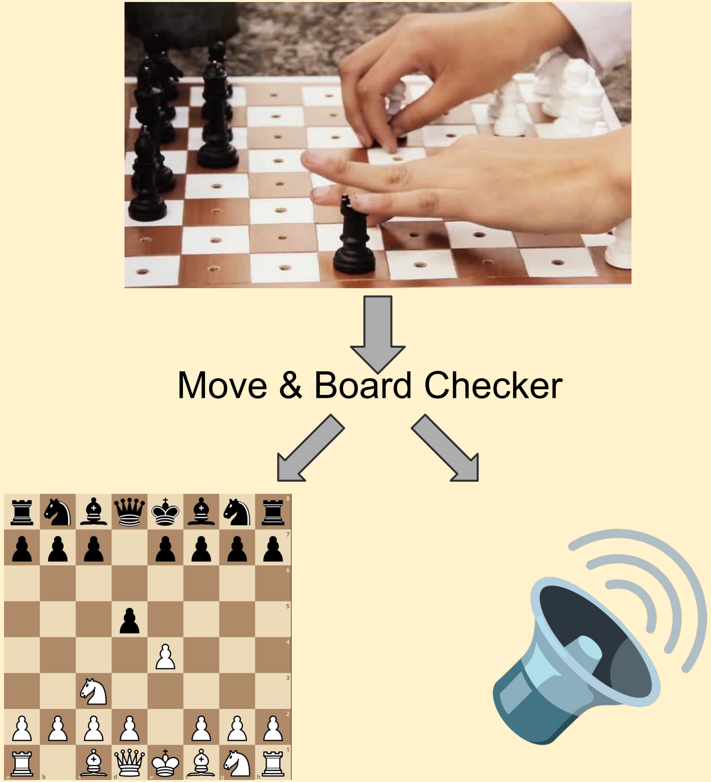
Case Requirements - Piece Detection & Board integrity

- We store the starting state of the board.
- We store the moves in for every game
  - Store only one game’s data which can be exported to a PDF file
- Distinguish between white and black pieces with 100% accuracy.
- Determine if a move is legal.
- We know the status of each square through our sensors



Use Case Requirements - Accuracy & Latency

- We want our board to record and upload moves with 100% accuracy
  - Inaccuracies lead to wrong computations and results.
- Our goal for move vocalization is also 100% accuracy
  - Inaccuracies will result in users receiving wrong inputs and hence, a wrong state of the board
- Latency should take at most 2 seconds for registering user inputs.



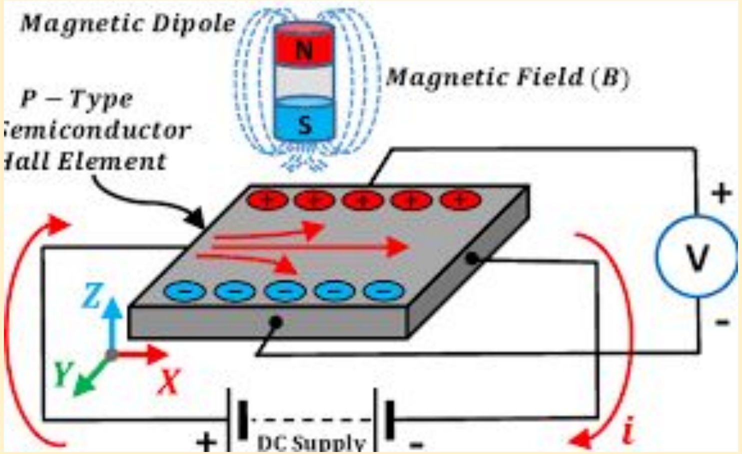
Move & Board Checker

Technical Challenges

- Distinguish the White and the Black pieces
- Low latency in detecting pieces, registering moves, and vocalising moves
  - Data collection from board, legality check, move vocalisation, state logic
- Keep track of initial board state and moves performed
  - Use an object to keep track of every current board state
- Vocalization of opponent moves.
- Allow board to be connected seamlessly

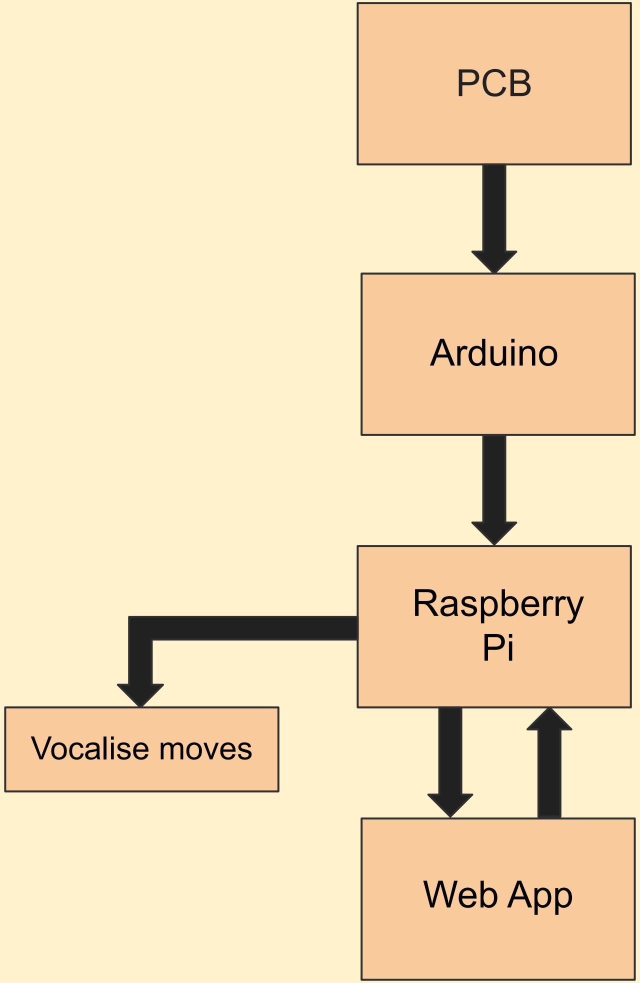
Solution Approach - Hardware

- Custom PCB + Hall Effect Sensors
  - Hall effect sensors will be used to differentiate pieces on the board by using varying strengths of magnetic fields between the pieces and board
- Arduino (convert analog to digital)
  - Use internal sensors in arduino to communicate with sensors on board
  - Convert signals from sensors to digital, then send it to the Raspberry Pi (under 15 ms to acquire and send data)
  - Data received and processed in Raspberry Pi in under 50 ms
    - Leaves over 90% of goal time



Solution Approach - Software

- Use a software approach to test and track gameplay
  - Store board status in a matrix and a list of legal performed moves
  - Store current board state as an object
  - Vocalise results of moves/validity of moves/opponent's moves
- Web Application
  - Will make calls to lichess.org's public API in order to register the user's move.
  - After inputting a move it will wait for the opponent's move to register and then communicate with the Raspberry Pi.



Testing, Verification, & Metrics

- System Tests
  - Unit Tests
    - Sensors
      - piece detection
    - Raspberry Pi
      - connecting to hardware and software systems
      - Collecting and transmitting data (maintaining integrity)
    - Web Application
      - Receiving data from board
      - Storing moves
      - Formatting and connecting requests to API
  - Integration Tests
    - Piece detection and data transmission to the Raspberry Pi
    - Raspberry transmits information to web application
    - Web application makes correct API calls and stores board information accurately
    - Want to make sure this is a seamless process under 2 seconds
- Quantitative Tests
  - Latency tests (entire system)
  - Battery lifetime and power usage
  - Accuracy and integrity of gameplay

Tasks & Division of Labor

Juan Mejia (Circuits, Signal Processing)

- Circuits, CAD design, Arduino and Raspberry Pi

Edison Aviles (Software Systems)

- Web Application, API calls and Raspberry Pi

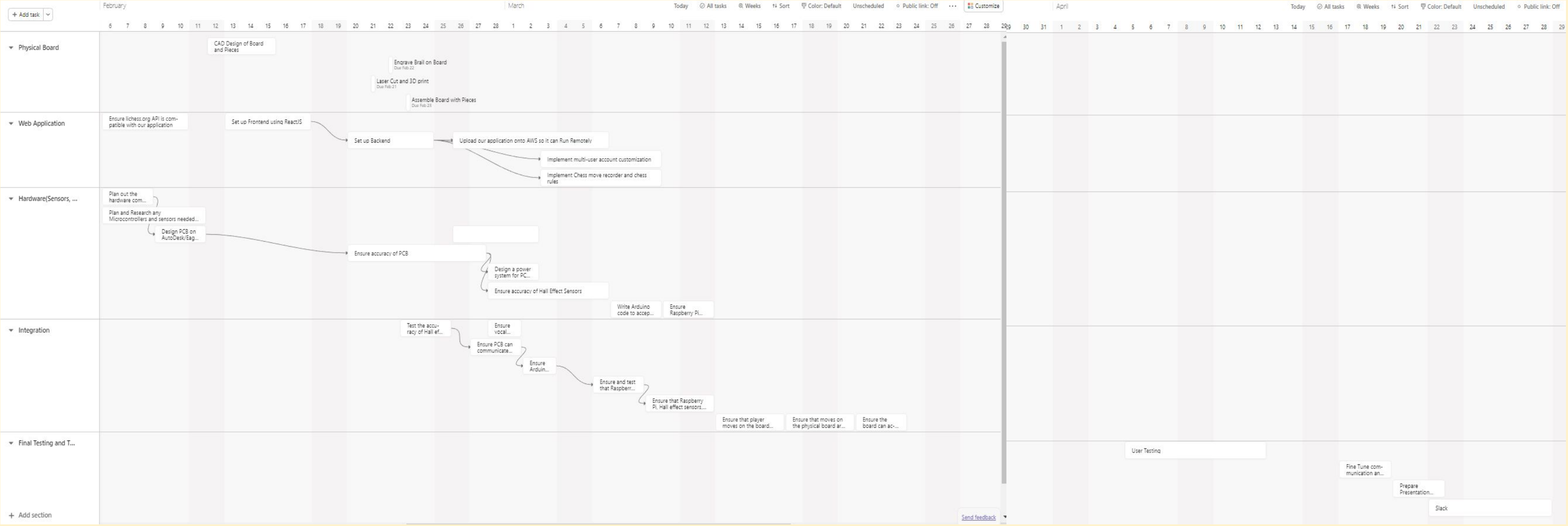
Mukundh Balajee (Software Systems, Hardware Systems)

- PCB, Sensors and Debugging

Entire Group

- Integration, Testing

Schedule



Physical Board
- CAD Design of Board and Pieces
- Engrave Brail on Board (Due Feb 22)
- Laser Cut and 3D print (Due Feb 21)
- Assemble Board with Pieces (Due Feb 23)

Web Application
- Ensure lichess.org API is compatible with our application
- Set up Frontend using ReactJS
- Set up Backend
- Upload our application onto AWS so it can Run Remotely
- Implement multi-user account customization
- Implement Chess move recorder and chess rules

Hardware(Sensors, ...
- Plan out the hardware com...
- Plan and Research any Microcontrollers and sensors needed...
- Design PCB on AutoDesk/Eag...
- Ensure accuracy of PCB
- Design a power system for PC...
- Ensure accuracy of Hall Effect Sensors
- Write Arduino code to accep...
- Ensure Raspberry Pi...

Integration
- Test the accuracy of Hall ef...
- Ensure vocal...
- Ensure PCB can communicate...
- Ensure Arduin...
- Ensure and test that Raspberr...
- Ensure that Raspberry Pi, Hall effect sensors,...
- Ensure that player moves on the board...
- Ensure that moves on the physical board ar...
- Ensure the board can ac-...

Final Testing and T...
- User Testing
- Fine Tune communication an...
- Prepare Presentation...
- Slack