MaGomoku Final Presentation

Team **C4** – Shuailin Pan, Sizhe Chen, Zipiao Wan

Presenter : Shuailin Pan

Problem Statement & Use Case

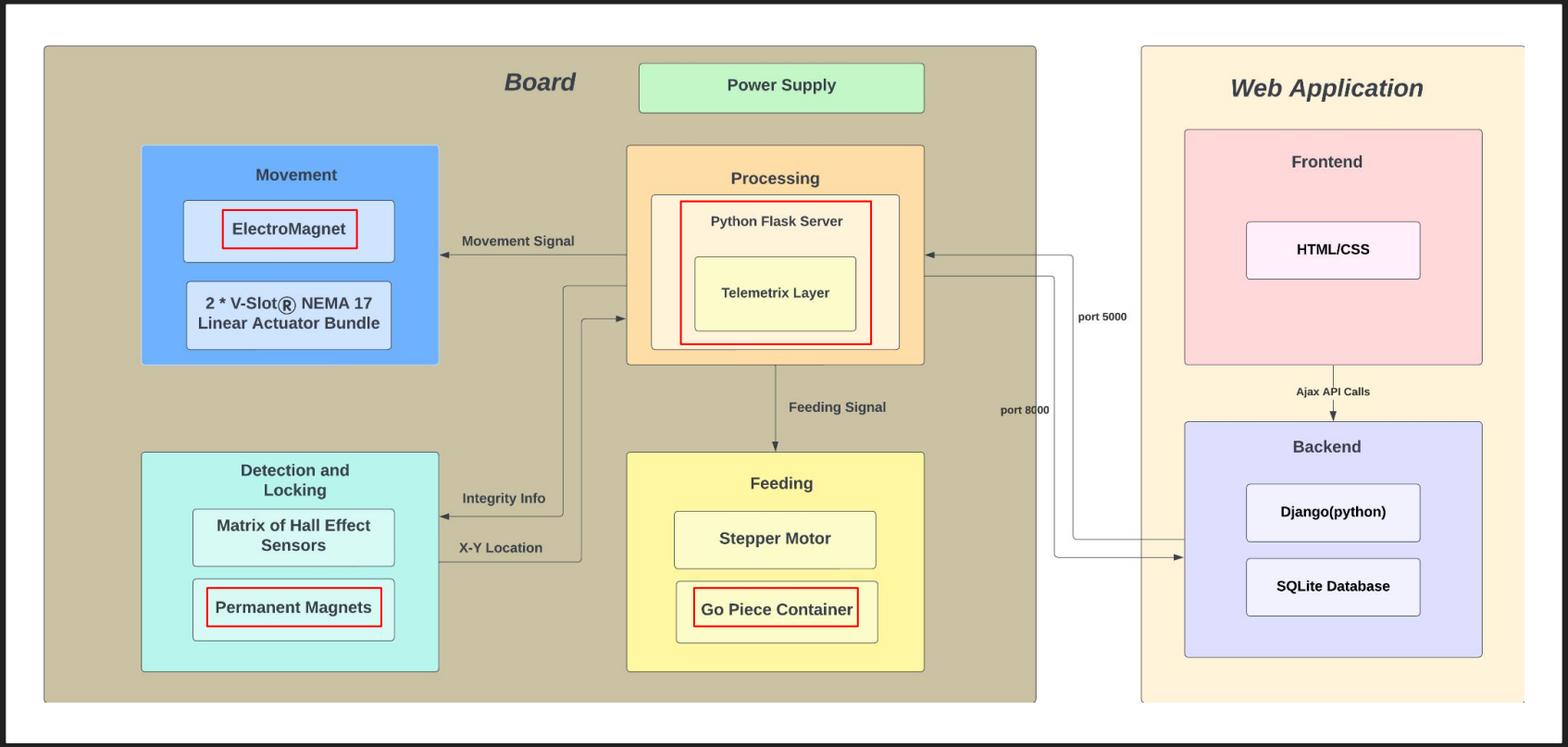
An automated Gomoku game board (game set) made for:

- Gomoku Lovers who want to level-up their experience
- Want to enjoy the physical game with online friends
- Elderly Users who are not familiar with online Gomoku games
- Tech enthusiast who want to try out magnetic controlled devices
- Interesting game that can kill time and boredom

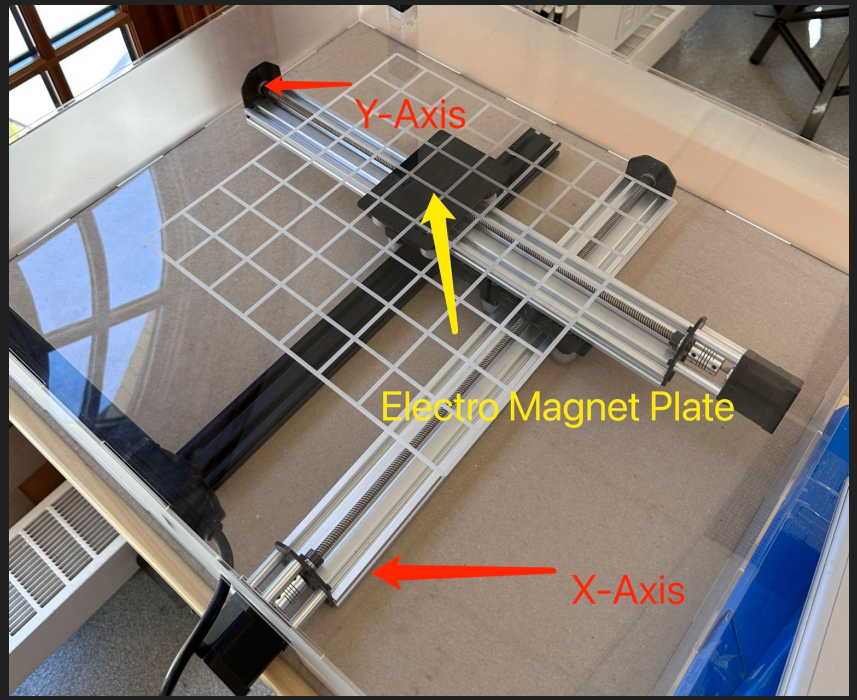
Design Requirements

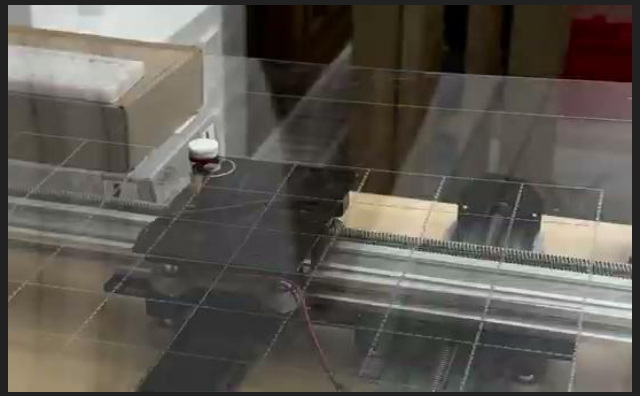
Use Case	Requirements
Detection & Board Integrity	• Distinction between black and white piece (aim for 95%) • Detection of piece presence on the board (aim for 95%) • Use sensors and software to ensure board integrity
Movement	• ~~Stable magnetic levitation pick-up of piece (90% success)~~ • Stable piece transportation (90% success) • Accurate piece landing (within 5mm of the center) • Fast feeding and movement (within 13s)
User Experience	• Easy setup (game setup done in app only) • Easy to play (only need to deal with his/her own piece) • Low latency (maximum latency of 1 second)

Solution Approach



Movement





Use an xy-gantry system powered by stepper motors and a electromagnet to drag a piece from point A to point B.

Movement

Challenges & Trade-offs

Linear Guide Rails vs Belt Driven

Maglev vs Electromagnet

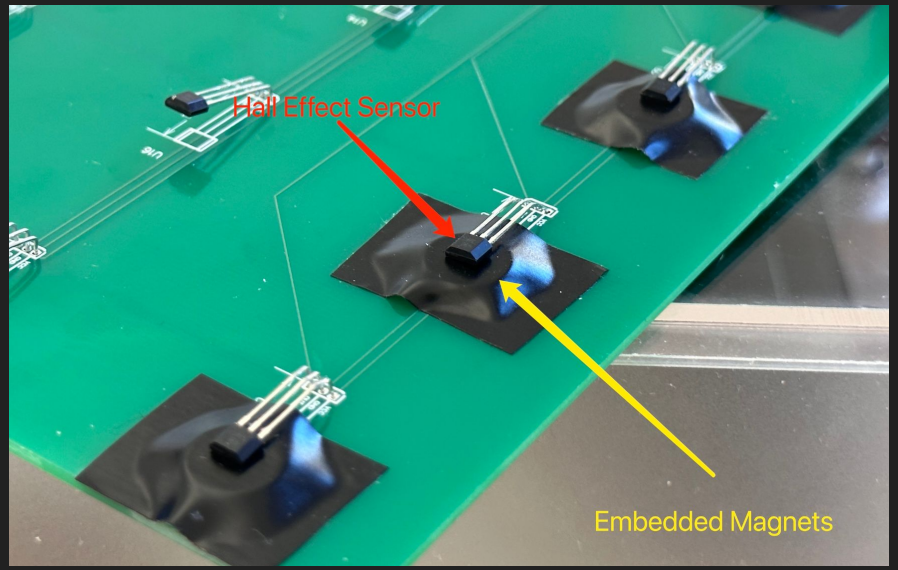
5V vs 12V Electromagnet

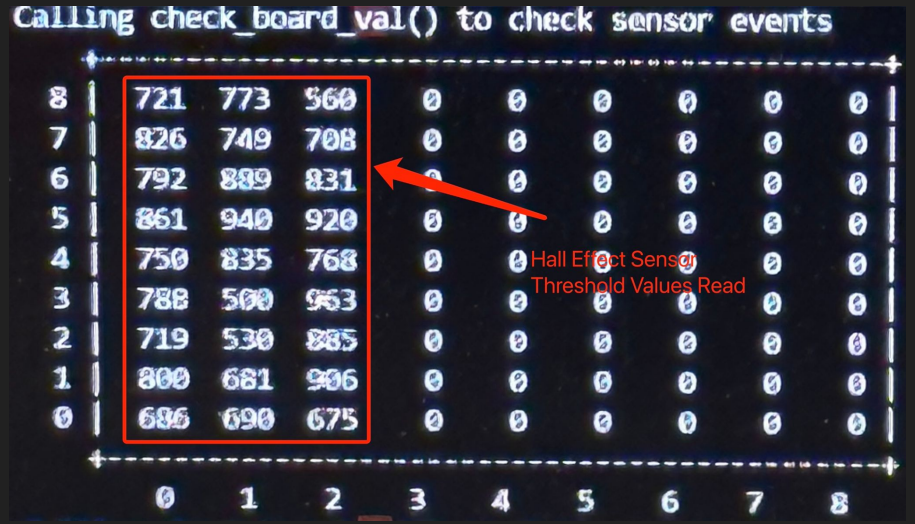
Grid Size -> 40mm (Locking & Detection)

Testing

Maglev Pickup Success Rate	(w/ human) ~33%	(w/o human) 0%
5V EM White Piece Success Rate	8 / 10	
12V EM White Piece Success Rate	10 / 10	
Longest Path Timing	11s	

Detection & Locking





A matrix of Hall Effect sensors to detect pieces

Small fixed position permanent magnets for locking

Detection & Locking

Challenges & Trade-offs

Electromagnet vs Permanent Magnet

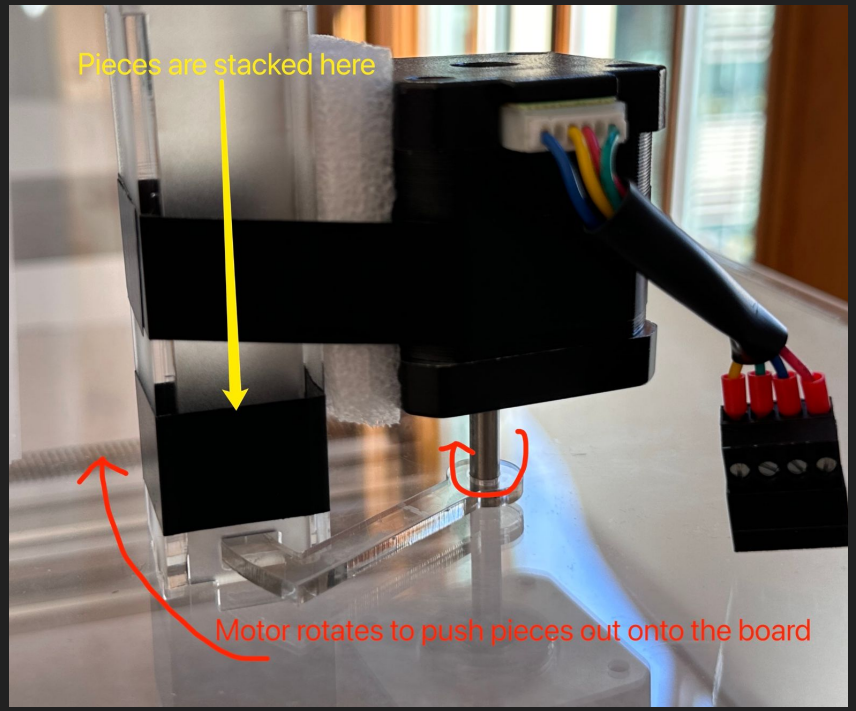
PCB physical layout

3 layer vs 2 layer piece design (Feeding)

Testing

Locking Success Rate	100%	
Piece Presence Detection Accuracy	50 / 50	
White/Black Distinguish Accuracy	3 layer (~80%)	2 layer (49 / 50)
Matrix Detection Latency	??	

Feeding



Challenges

Spring Loaded (Horizontal)

vs

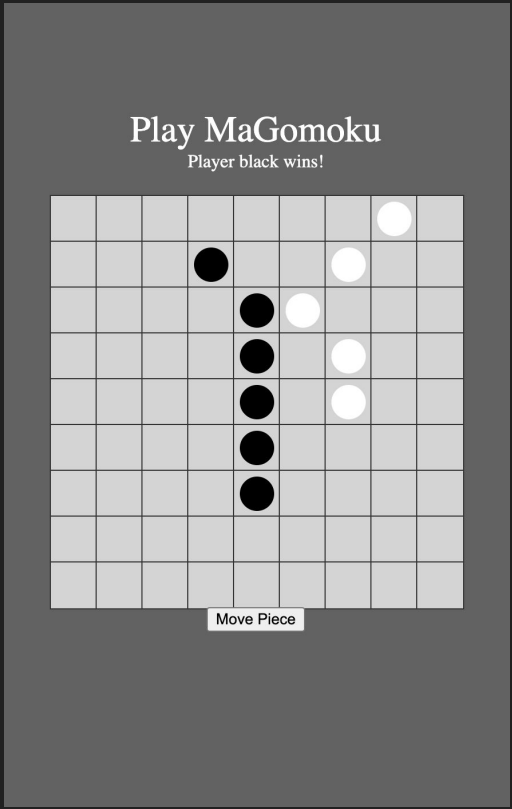
Gravity (Vertical)

Testing

Feeding Success Rate (w/o EM)

Black Piece	17 / 20
White Piece	20 / 20

Software



- Use Django(python) as the backend and html/css as the frontend to achieve the game state control and communication with the online gomoku platform.

- Use Flask for local communication via api endpoints between hardware program & webapp program

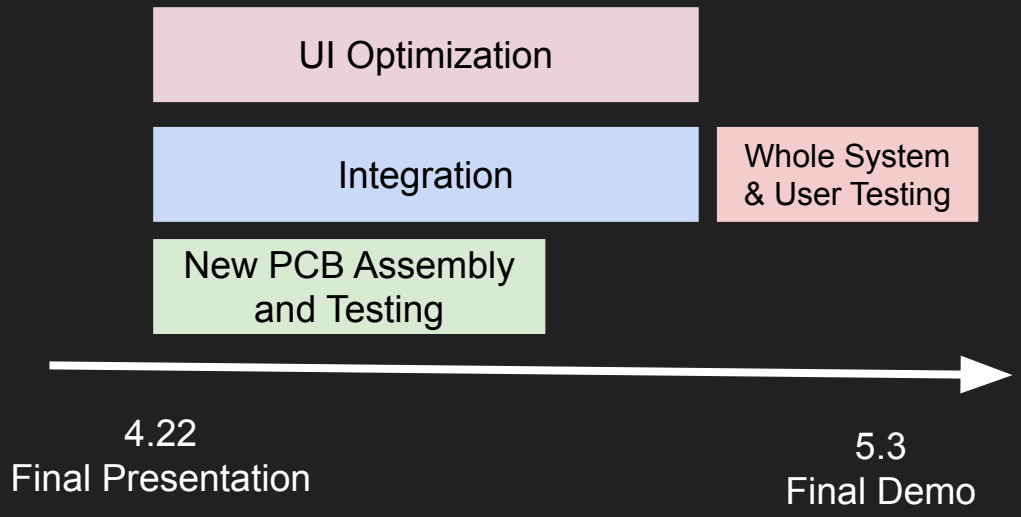
Project Management

Shuailin Pan
- Feeding system development
- Board assembly, Laser Cutting
- Hall-effect sensor development

Sizhe Chen
- Piece detection algorithm development
- Hall-effect PCB design
- Gantry interface development

Zipiao Wan
- Web application development
- Arduino interface Development
- Gantry system development

Next Steps



What we learned

1. **Importance of separation of responsibilities and the interface**
   - We set a clear bound of duties between software and hardware, and spent lots of time specifying data standards between the two, which made the coordination and integration easy.
2. **Importance of feasibility testing of core components ASAP**
   - We spent lots of time assuming magnetic levitation works, but it actually not, so we have to resort to another solution.
3. **Importance of design review of high risk items before execution**
   - Ordering and shipping PCB board takes a long time, need to double check the design is feasible before placing orders.
4. **Learning new knowledge through practice and non-standardized source**