TARTAN'S GAMBIT

Lillie Widmayer, Luis Ortega, Juan Cortes

Use Case

- physical chess playing experience without the need for physical contact
- Allow professional chess players to practice specific scenarios

Requirements - mechanical hardware

- Pieces moved by gantry system should land completely within intended square bounds
- Pieces should be moved within 10 seconds of Web command
- Others pieces should not be interfered with unless they are involved in the move (i.e Castling, Capturing)
- Pieces captured will be placed by the gantry in a separate receptacle
- Gantry system will be able to reach farmost limits of the board and the separate receptacle
- Gantry should only move on its turn to not affect user

Requirements - electrical hardware

- Moves are detected with 99% accuracy as long as player moves the piece within the proper bounds
- User should be notified if piece is within multiple squares
- Piece movement should be detected and received by the web interface within 10 seconds of player dropping the moved piece
- User should be notified of turn with an LED
- User should be notified if movement was detected with an LED
- User should be notified if they are performing an illegal move

Requirements - software

- CV should be able to recognize individual pieces at each board state with 99% accuracy
- If the physical player picks up a piece and drops it in the same square the system should recognize they have not completed their turn
- System should track when pieces have been move and captured and update the interface accordingly
- Pieces should snap into place when dragging and dropping moving pieces
- User should not be allowed to capture their own side’s pieces
- Web client should
    - send the command within 5 second of movement
    - continuously indicate user of connection to the board
    - communicate with board through WIFI
    - only allow valid moves

Technical Challenges

- Response time
  - Mechanical
  - Electrical
  - Software
- Integration
  - Communication between web client and board
  - Communication between boards (eventually)
- Accuracy
  - Detecting correct piece movement
  - Moving pieces on board/taking pieces off board
- Using CV
- Calibrating timing belts for accurate movement

Solution Approach

- Raspberry Pi
  - OpenCV for piece tracking and movement
  - Django for web server interface
  - Chess.js for move validation
- Board
  - Plastic hollow chess pieces, standard size
  - Wooden chess board, standard size
- Gantry System
  - 12V Nema Stepper Motors
    - Timing belts to cover length and width of board
  - Servo motor for the z axis movement and gripper
    - Gripper is 3D printed with a grip lining
  - HD Camera in bird’s eye view attached to top of gantry
  - LED Indicators for user clarity

Testing, Verification, Metrics

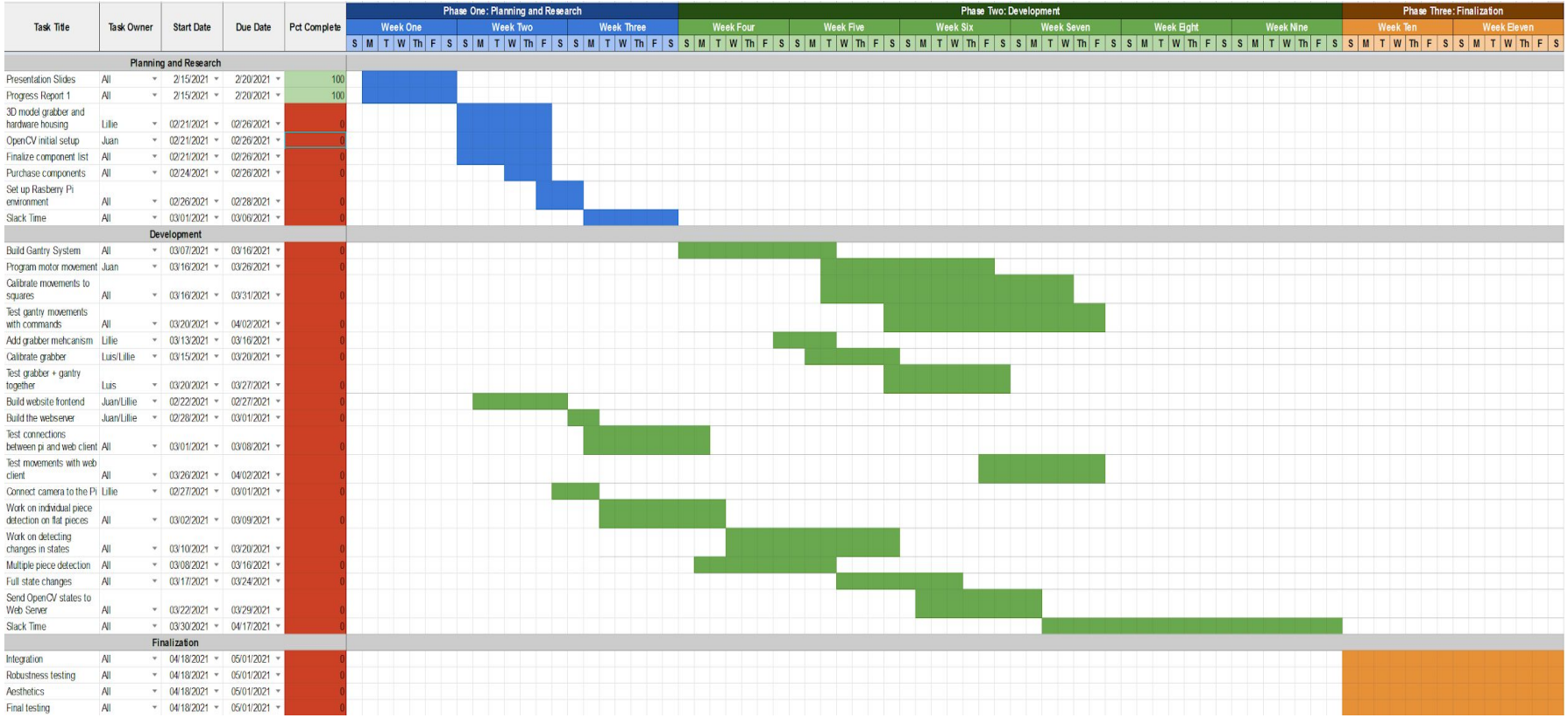
- Mechanical
  - Measure how long pieces take to move
  - Asses whether piece lands completely within intended square
  - Ensure that moving piece doesn’t interfere with other pieces
  - Test that the gantry can reach and move all pieces
  - Check if piece can be lifted and held during movement
- Electrical
  - Visual confirmation that pieces are being detected and moved properly
  - Test that LEDs turn on at appropriate times
- Software
  - Ensure that command being sent is the same as received
  - Test that the interface is functional
    - Move validation is being done
    - User is able to move pieces properly
    - Check that the web client is updating after each move

Final test - can you play a game of chess remotely?

Tasks and Division of Labor

- Lillie
  - Model 3D printed parts
  - Website development
  - Raspberry Pi web client setup
- Juan
  - Raspberry Pi web client setup
  - Website development
  - Programming timing belts
- Luis
  - Raspberry Pi setup (logic)
  - Model 3D printed parts
- All
  - Gantry system assembly and testing
  - openCV
  - Documentation

Schedule



Task Title	Task Owner	Start Date	Due Date	Pct Complete
Planning and Research				
Presentation Slides	All	2/15/2021	2/20/2021	100
Progress Report 1	All	2/15/2021	2/20/2021	100
3D model grabber and hardware housing	Lillie	02/21/2021	02/26/2021	0
OpenCV initial setup	Juan	02/21/2021	02/26/2021	0
Finalize component list	All	02/21/2021	02/26/2021	0
Purchase components	All	02/24/2021	02/26/2021	0
Set up Rasberry Pi environment	All	02/26/2021	02/28/2021	0
Slack Time	All	03/01/2021	03/06/2021	0
Development				
Build Gantry System	All	03/07/2021	03/16/2021	0
Program motor movement	Juan	03/16/2021	03/26/2021	0
Calibrate movements to squares	All	03/16/2021	03/31/2021	0
Test gantry movements with commands	All	03/20/2021	04/02/2021	0
Add grabber mehcanism	Lillie	03/13/2021	03/16/2021	0
Calibrate grabber	Luis/Lillie	03/15/2021	03/20/2021	0
Test grabber + gantry together	Luis	03/20/2021	03/27/2021	0
Build website frontend	Juan/Lillie	02/22/2021	02/27/2021	0
Build the webserver	Juan/Lillie	02/28/2021	03/01/2021	0
Test connections between pi and web client	All	03/01/2021	03/08/2021	0
Test movements with web client	All	03/26/2021	04/02/2021	0
Connect camera to the Pi	Lillie	02/27/2021	03/01/2021	0
Work on individual piece detection on flat pieces	All	03/02/2021	03/09/2021	0
Work on detecting changes in states	All	03/10/2021	03/20/2021	0
Multiple piece detection	All	03/08/2021	03/16/2021	0
Full state changes	All	03/17/2021	03/24/2021	0
Send OpenCV states to Web Server	All	03/22/2021	03/29/2021	0
Slack Time	All	03/30/2021	04/17/2021	0
Finalization				
Integration	All	04/18/2021	05/01/2021	0
Robustness testing	All	04/18/2021	05/01/2021	0
Aesthetics	All	04/18/2021	05/01/2021	0
Final testing	All	04/18/2021	05/01/2021	0

Conclusion

- Phase 1:
  - Control chess pieces on a board via web application
  - Chess piece movement detection through CV
- Phase 2:
  - Wireless interaction between 2 separate chess boards