
Chess Game — Final Report

Design

Managing Material Imbalance Challenges through Tailored Advanced Solutions

Name:

Khoa Tran

Project Methodology:

Waterfall

Reason for Choosing Waterfall:

I selected the Waterfall methodology because it allows me to clearly define, schedule, and control the timeline of each phase of the project. This structured approach ensures that every stage is completed before moving on to the next, reducing confusion and improving project organization.

Login Flow



Login System Flowchart — System 1

The Login System

Justification:

1. The login system is necessary for the chess engine because a user may have multiple accounts with different game positions, openings, or piece arrangements. The login system allows users to switch between accounts easily, compare different board positions, and analyze moves with AI assistance. This helps users learn, improve, and evaluate different strategies.
2. Although login systems are not always required in chess applications unless designed for professional or tournament players, this system is still valuable. It allows users to practice safely without fear of their game history being accessed by others. In competitive environments, some players may even hire hackers to analyze opponent move logs, so having a secure login system helps protect user data and gameplay privacy.

Testing Resources Needed

Number of Test	Description	Data	Expected Result	Actual Result	Pass (Y/N)	Comment
1	Correct username entered	“Test user name = user log”	Proceed to password	System asked for username again	N	Valid username check failed
2	Correct username entered	“Test user name = user log”	Proceed to password	System asked for username again	N	Valid username check failed
3	Correct username entered	“Test user name = user log”	Proceed to password	System asked for username again	N	Valid username check failed
4	Correct username entered	“Test username = user log”	Prompt for password	Prompt appeared	N	Invalid password

Online Availability

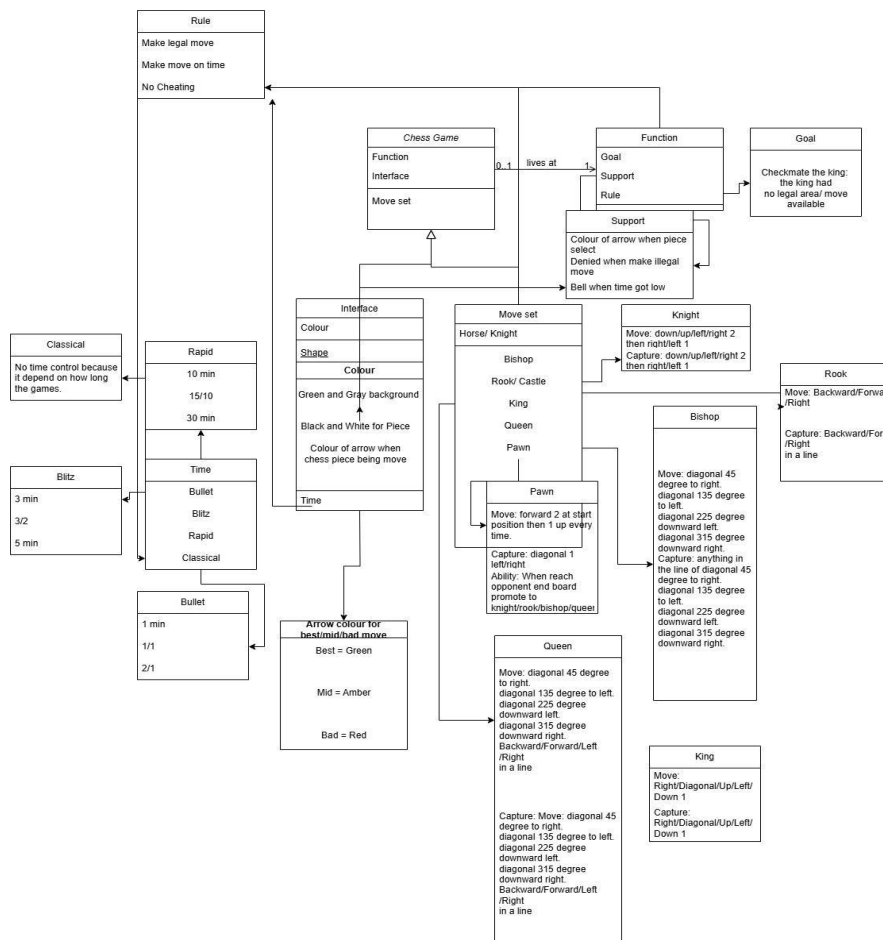
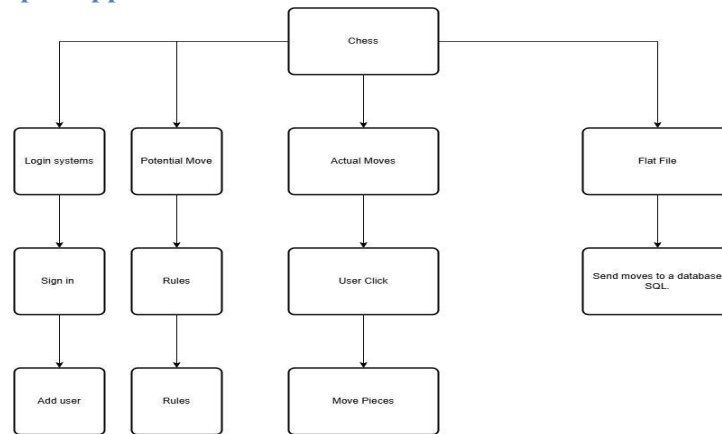
No

Justification:

I did not implement online multiplayer because there was not enough time within the project deadline, and online functionality was outside the official project scope.

Design — Login System for Chess Boards

Top Down Developer Approach

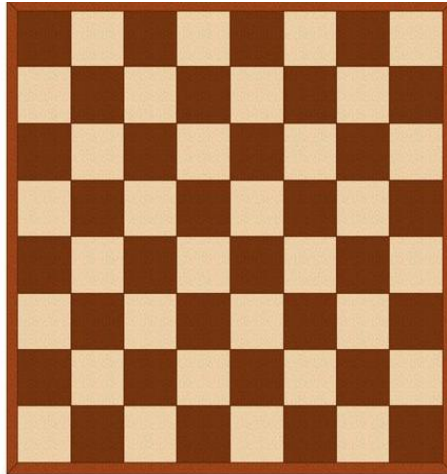


The Login System

Justification:

1. The login system is needed because users may have multiple accounts with different chess openings or board setups. This allows players to compare positions, experiment with different strategies, and analyze gameplay from multiple perspectives.
 2. The login system also ensures security by preventing unauthorized users from viewing another player's practice games or strategies.
-

Chess Board Layout and Square Design



- The board consists of **64 squares** in a standard checkerboard pattern.
- Minimum resolution: **1080 × 1080 pixels**.
- Each square is **135 × 135 pixels**.
- There are **8 rows and 8 columns**.
- Each chess piece is **100 pixels** and centered within its square.

Important Rule:

- The **White Queen** always starts on a white square.
 - The **Black Queen** always starts on a black square.
- This helps users correctly identify board orientation and prevents layout confusion.

The board is labeled **a–h horizontally** and **1–8 vertically**, allowing players — including visually impaired or blind players — to identify piece locations verbally or mentally.

This conceptual board layout was initially visualized in Python to assist in developing the final pseudocode and game logic.

Chess Piece Movement Rules

Rook (bR / wR)

- Moves any number of squares vertically or horizontally.
- Cannot jump over other pieces.
- Captures by landing on an opponent's piece in its movement path.



Knight (bN / wN)

- Moves in an “L” shape: two squares in one direction and one perpendicular.
- Can jump over other pieces.
- Captures by landing on an opponent's piece in an L-shaped move.



Bishop (bB / wB)

- Moves diagonally any number of squares.
- Cannot jump over pieces.
- Captures diagonally.



Queen (bQ / wQ)

- Combines rook and bishop movement.
- Moves horizontally, vertically, or diagonally.
- Cannot jump over pieces.



King (bK / wK)

- Moves one square in any direction.
- Cannot move into check.
- If the king has no legal moves and is in check → **Checkmate (game over)**.



Pawn (bp / wp)

- Moves one square forward.
- Captures one square diagonally forward.
- Special moves:
 - **En passant**
 - **Promotion**



Special Pawn Rules

En Passant

If an opponent moves a pawn two squares forward from its starting position, the opposing pawn may capture it diagonally as if it had moved only one square.

Pawn Promotion

When a pawn reaches the final rank (rank 8 for white, rank 1 for black), it can be promoted to:

- Queen
- Rook
- Bishop
- Knight

Although this rule existed earlier, it became widely recognized after Bobby Fischer popularized its strategic importance in competitive play.

Game Rules

Capture

A capture occurs when a player moves their piece onto a square occupied by an opponent's piece, removing it from the board.

Check

A king is in check when it is under direct attack by an opponent's piece.

Checkmate

If a king is in check and has no legal moves to escape, the game ends.

Stalemate

If a player has no legal moves but is not in check, the game is a draw.

Time Control

Examples:

- **Blitz 3/2** → 3 minutes + 2 seconds per move
- **Blitz 3/0** → 3 minutes, no increment
- **Rapid 15/10** → 15 minutes + 10 seconds per move
- **Classical 60/0** → 60 minutes, no increment

Control System

When a piece is selected:

1. Legal moves are highlighted.
2. Illegal moves are rejected.
3. The piece can be dragged with the mouse.

Illegal moves cause the piece to snap back and display:

"The move is incorrect, please try again or learn the move."



Board Rendering Using Pygame

I used **pygame** because it includes built-in mouse detection features such as `MOUSEBUTTONDOWN`, which simplifies debugging and interaction handling.

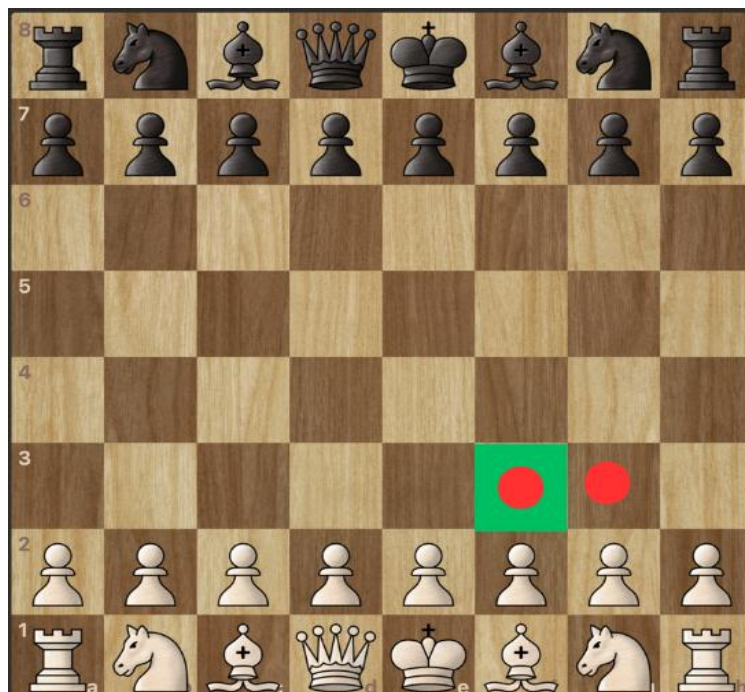
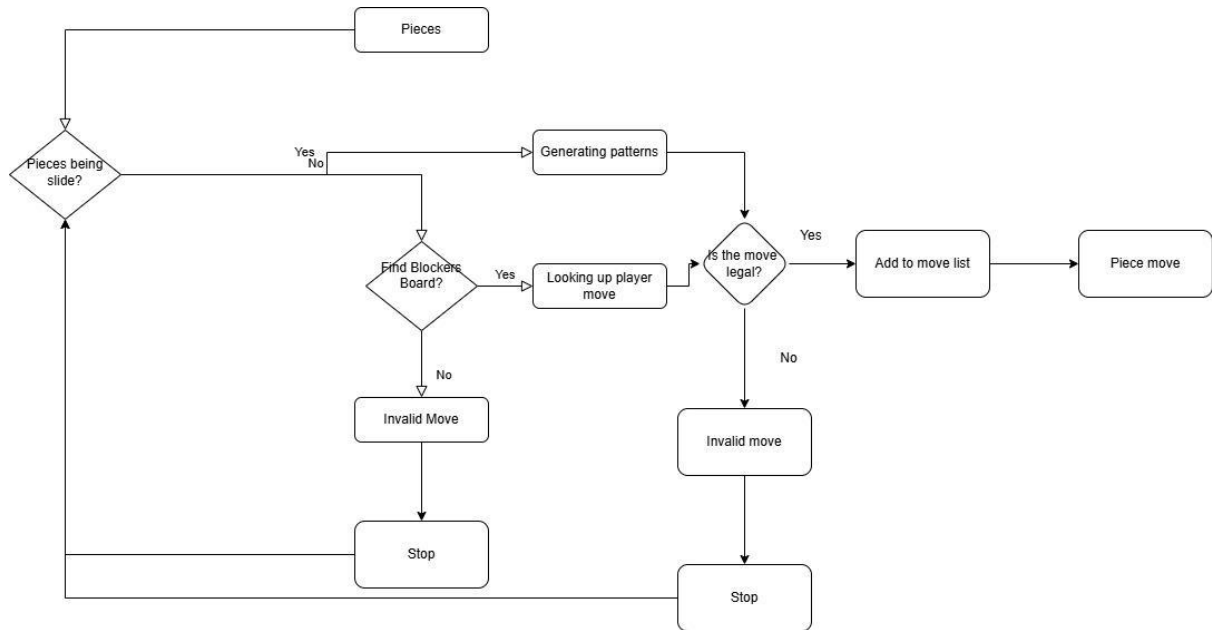
- Board size: **1080 × 1080**
- Square size: **135 × 135**
- Board stored as a 2D array with rows and columns.
- Pieces loaded as images from a local folder:
 - wR, wN, wB, wQ, wK, wp
 - bR, bN, bB, bQ, bK, bp

Legal Move Generation Algorithm

Each piece type has its own movement function:

- Pawns → `PawnMoves()`
- Rook → `StraightLineMoves()`

- Bishop → DiagonalLineMoves()
- Knight → KnightMoves()
- Queen → Combination of rook + bishop moves
- King → KingMoves()







Game Evaluation System

A basic material evaluation score is calculated:

Piece	Value
Pawn	1
Knight	3
Bishop	3
Rook	5
Queen	9
King	0

White adds points; Black subtracts points.

Iterations and Testing

Iteration 1 — Board Visualization

- Implemented an 8×8 checkerboard.
- Displayed all pieces in correct starting positions.
- Used $(r + c) \% 2$ to alternate square colors.

Result:

- Initially failed, then fixed and passed.
-

Iteration 2 — Movement and Legality

- Implemented mouse click movement.
- Added legal move highlighting.
- Prevented illegal moves.

Test Results:

- Pawn E2 to E4 → Pass
 - Knight B1 to C3 → Pass
 - Illegal pawn move → Rejected correctly
-

Iteration 3 — Advanced Rules

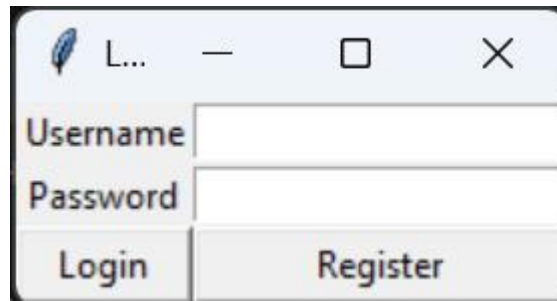
Added:

- Castling
- En passant
- Pawn promotion
- Check detection
- Checkmate
- Stalemate

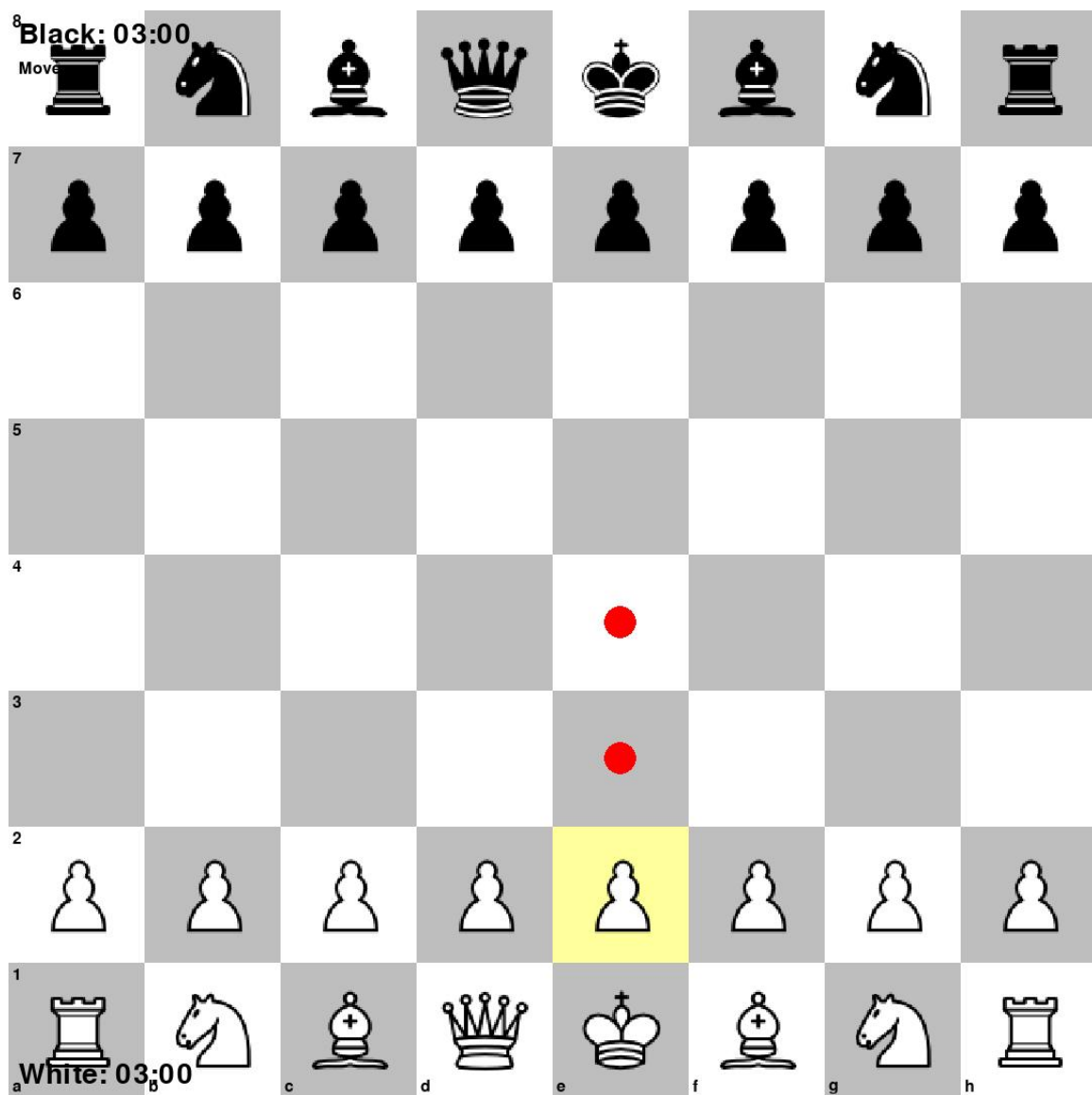
All tests passed after fixes.

Iteration 4 — Database Login

Used a CSV file to store usernames and passwords, later imported into SQLite.



A web form with a title bar containing a feather icon, the text 'L...', and standard window controls (minimize, maximize, close). The form has two input fields: 'Username' and 'Password'. Below the input fields are two buttons: 'Login' and 'Register'.



8
Black: 02:58

Move

1. e2

2. e3

7

6

5

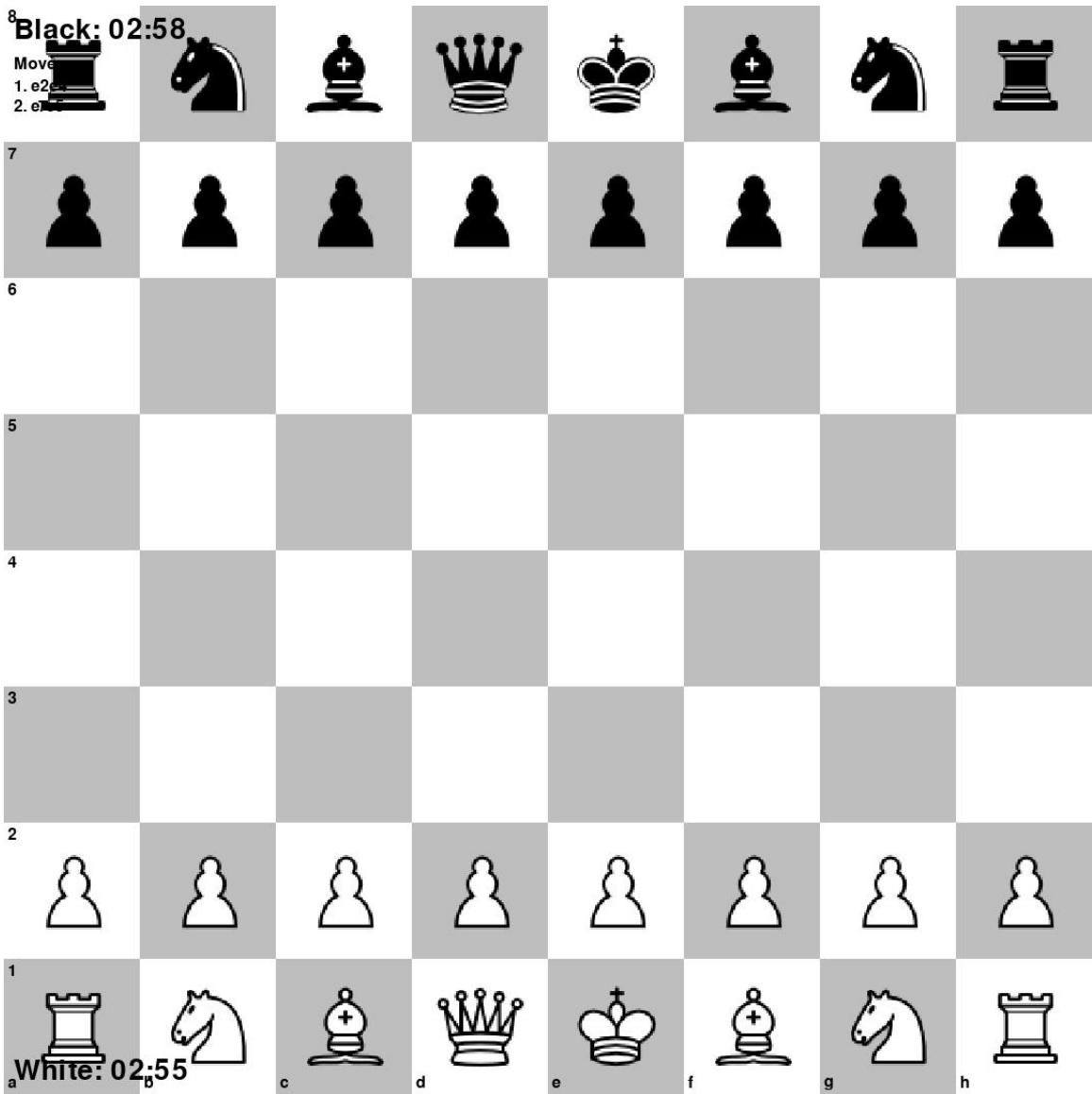
4

3

2

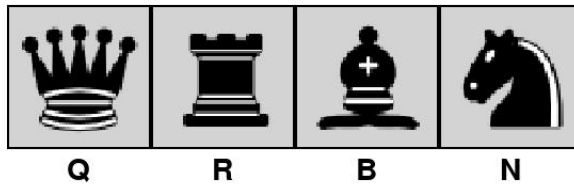
1

a White: 02:55



Checkmate - White wins

Choose promotion piece:



Play Again

New Game

Press R or N for Restart



External References

- **Pygame Documentation:** <https://www.pygame.org/docs/>
 - **Tkinter Documentation:** <https://docs.python.org/3/library/tkinter.html>
 - **FIDE Laws of Chess:** <https://www.fide.com/FIDE/handbook/LawsOfChess.pdf>
 - **Chess Game Tutorial:** https://youtu.be/EnYui0e73Rs?si=HHI7W8qY_hEOmjsw
 - **SQLite Guide:** https://www.w3schools.com/sql/sql_insert.asp
 - **Python SQLite:** <https://www.geeksforgeeks.org/python/python-sqlite-connecting-to-database>
-

Final Note

If you want, I can:

- format this directly into **your Word document**, or
- convert this into a **formal academic report style**, or
- rewrite it in **perfect university-style English**.

Just tell me your preference.