

Project Presentation On ♟ Chess : Mastering the Game ♟

Course Code : CSE-3636

Submitted To :
Sara Karim
Adjunct Lecturer, Dept. of CSE, IIUC

Submitted By :
Nahian Subah Ishma_C223286
Rehnuma Tasneem_C223288
Saima Kawsar_C223297
Semester : 6th

Introduction :

Overview:

- A chess-playing game that implements all chess rules, including advanced moves.
- Demonstrates game decision-making in a strategic game environment.

Importance:

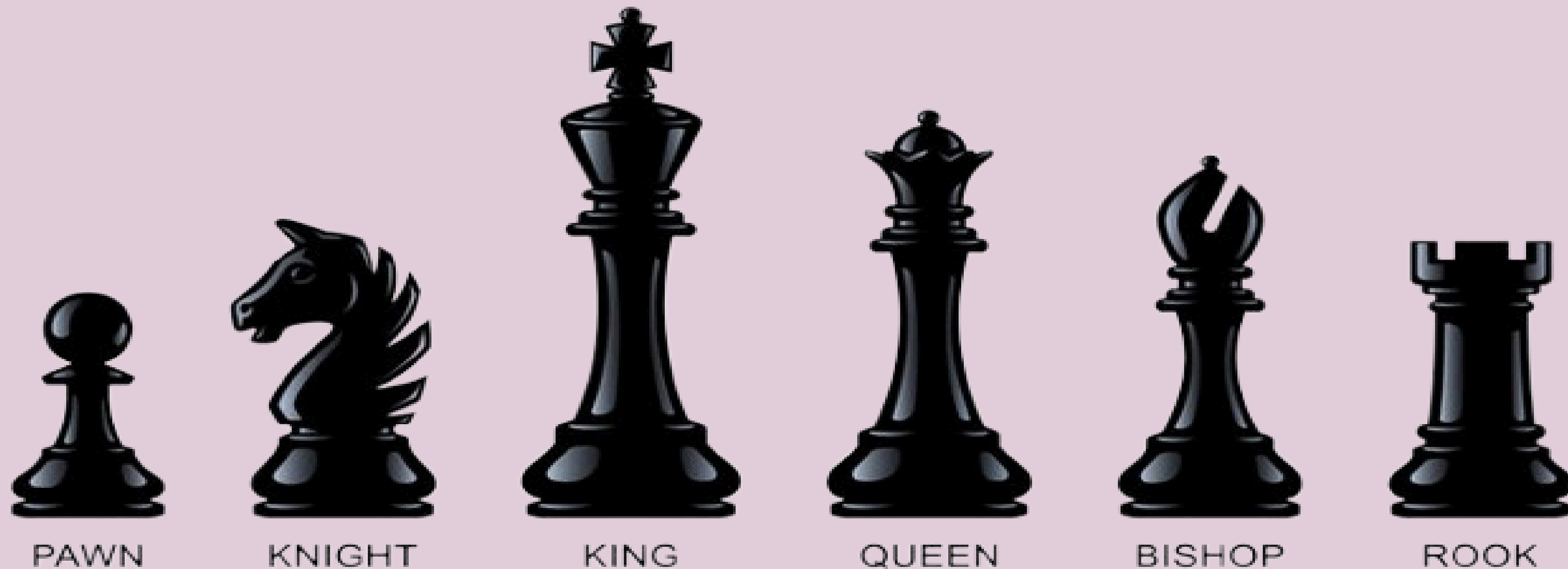
- Develops expertise in game programming and logic.
- Enhances understanding of decision-making algorithms.

Game Design and Concept

Why I Chose This Project ?

- **Hands-on AI Application:** Implementing AI algorithms in Python.
- **Challenging Complexity:** Chess offers a complex, strategic environment to develop advanced AI.
- **Skill Growth:** Enhances my programming and AI skills.
- **Real-World Relevance:** A functional chess game that demonstrates practical moves to use in gaming.

Main Character



Pawn: Moves one square forward (two squares on its first move), captures diagonally

Knight: Moves in an "L" shape (two squares in one direction, then one perpendicular)

King: Moves one square in any direction

Queen: Moves horizontally, vertically, or diagonally any number of squares

Bishop: Moves diagonally any number of squares

Rook: Moves horizontally or vertically any number of squares

Board Representation



0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0

White Pawns

0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 1 0 0 0 0 1 0

White Knights

...

0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

Black King

Board Representation



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

White Pawns

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 3 | 0 | 0 | 0 | 0 | 3 | 0 |

White Knights

...

| | | | | | | | |
|---|---|---|---|----|---|---|---|
| 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Black King

Game Play

- **Goal:** The goal is to checkmate your opponent's king, meaning the king is under threat and cannot escape capture.
- **Turns:** Players take turns moving one piece at a time, starting with the white player.
- **Winning:** A player wins by checkmating the opponent's king or if the opponent resigns.
- **Draw:** The game ends in a draw if there is a stalemate (no legal moves left), threefold repetition, or insufficient material to checkmate.

Features

- **Full Chess Rules**

Includes advanced moves: castling, en passant, and promotion

- **2-Player Mode**

Play against another person with the help of AI moves

- **Interactive Chessboard**

Real-time board updates with valid move highlighting with a status bar under the board . As pieces are captured, they are immediately added to the sidebar

- **Visual Representation**

Captured pieces are displayed on the side of the board, organized by player

Features (Continued)

The following algorithms and techniques were implemented:

1. Game Over Detection

- Checks if a King is captured to declare the winner
- Does not yet evaluate checkmate or stalemate conditions

2. Turn-Based Control

- Alternates between White and Black players using a turn variable
- Players select and move pieces during their turn

3. Piece Capture and Board Updates

- Captures opponent pieces when a valid move lands on an occupied square
- Updates the board state after each move

Features (Continued)

4. Brute Force Move Generation

Generates valid moves for each chess piece by exploring all possible directions.

a. **Rook:** Moves along horizontal and vertical lines until blocked.

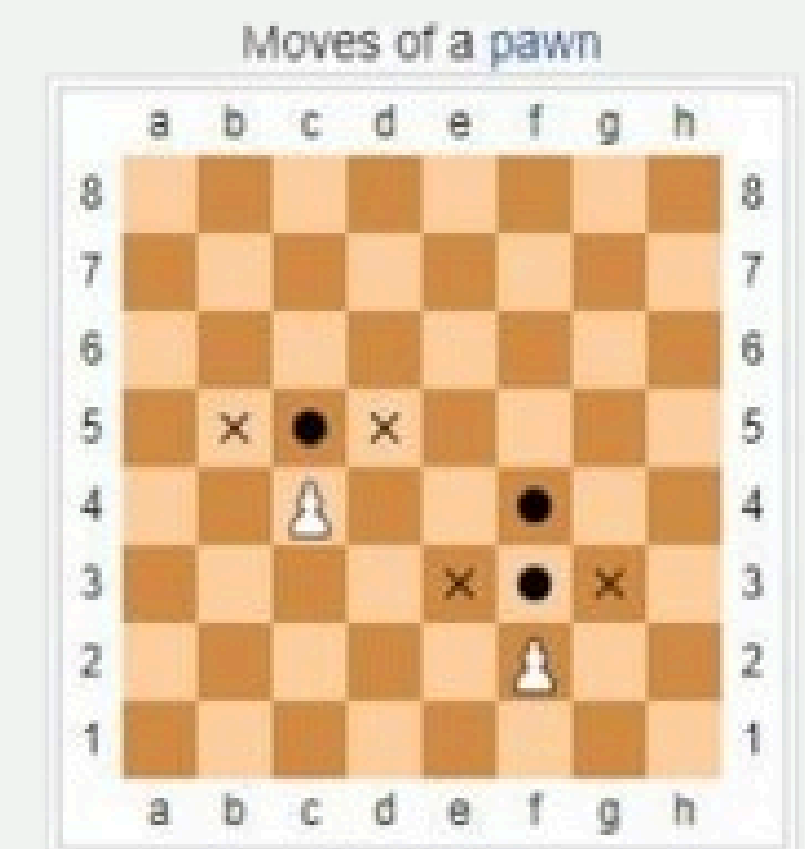
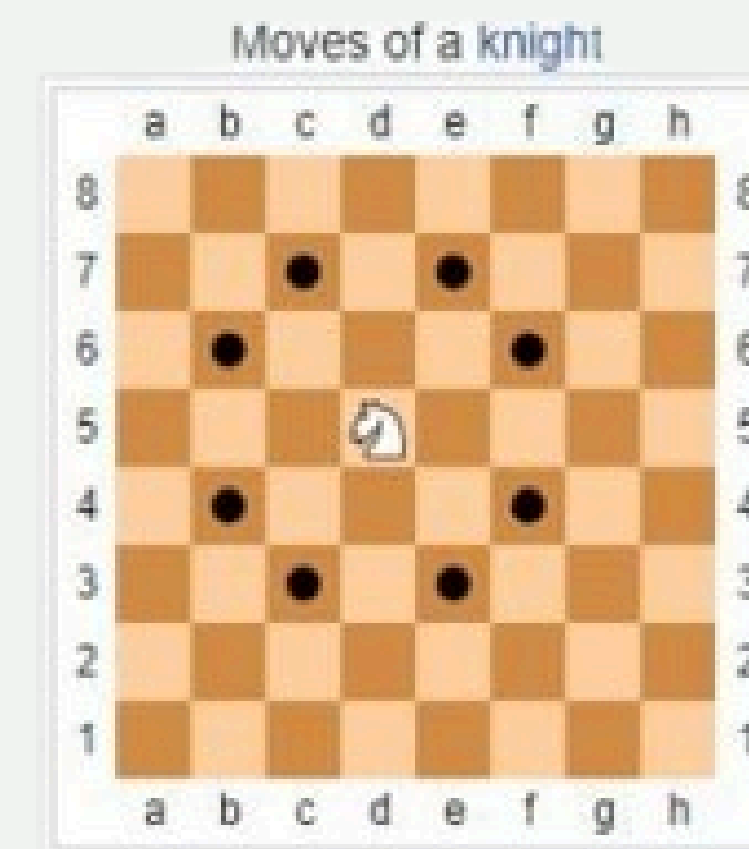
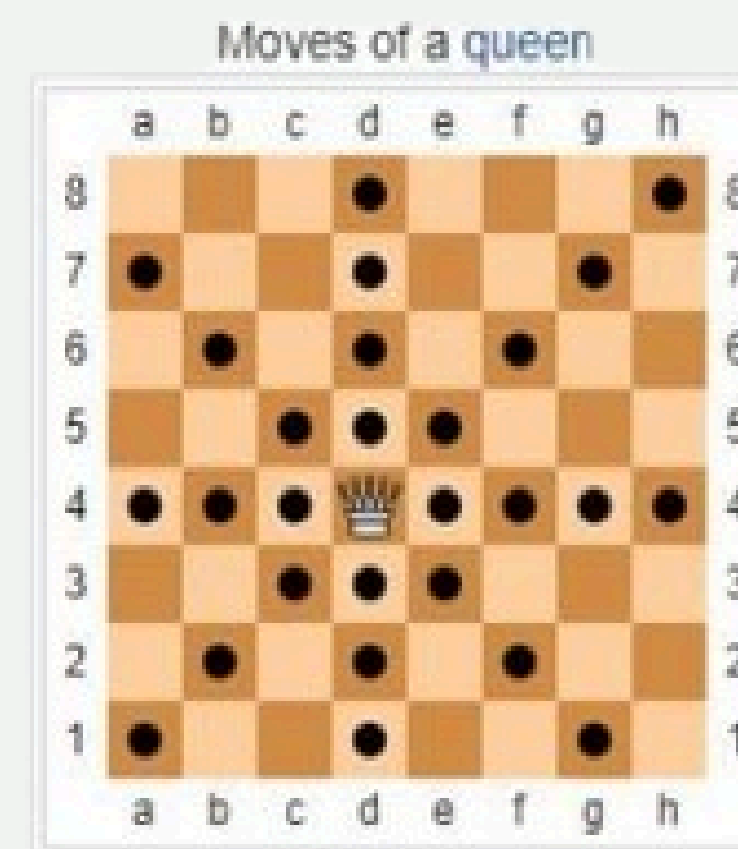
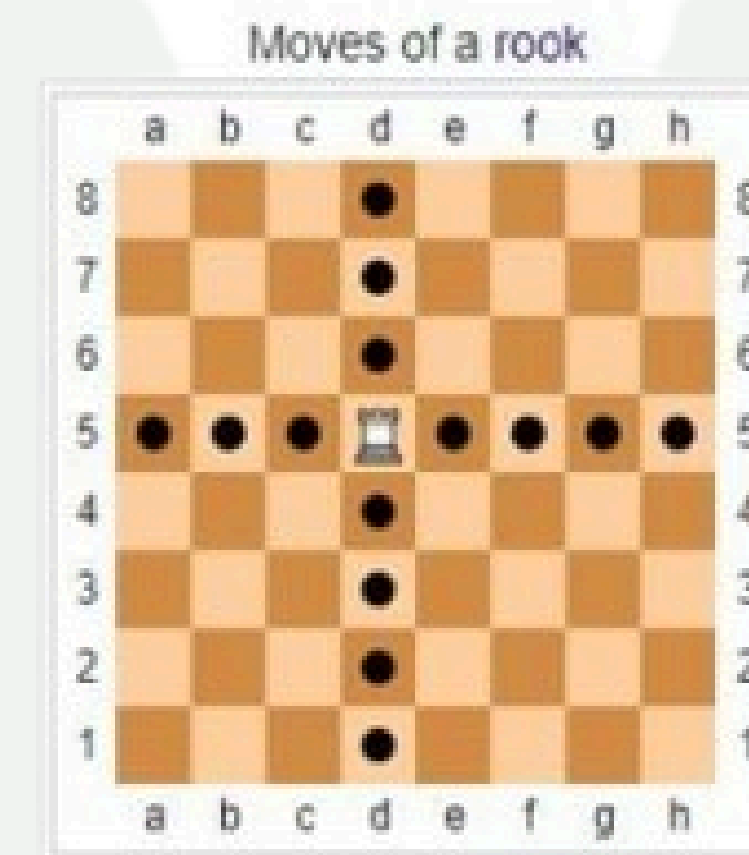
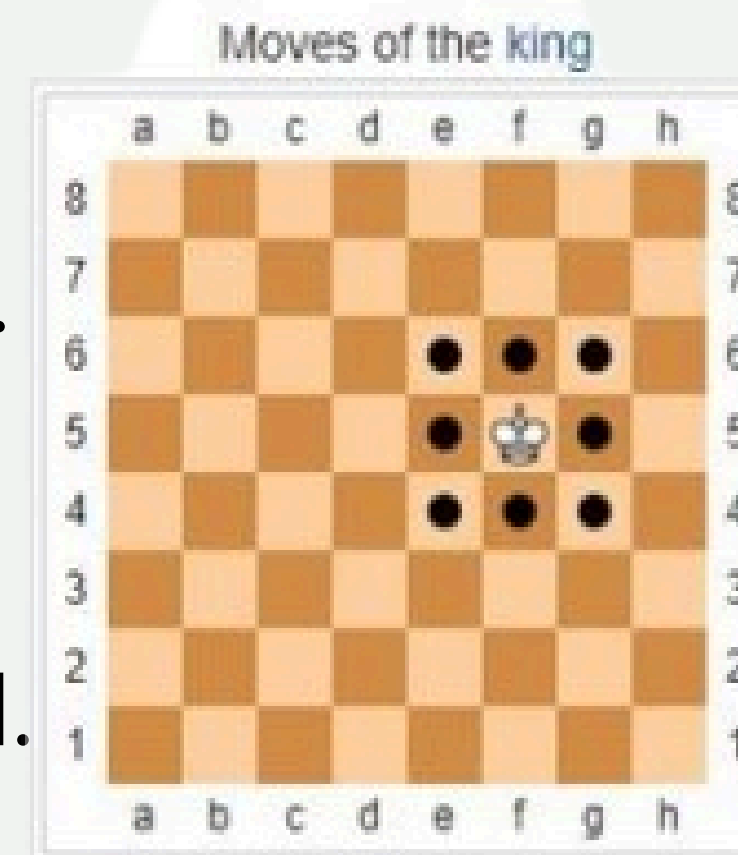
b. **Bishop:** Moves diagonally until blocked.

c. **Knight:** Jumps in an "L-shape" to specific squares.

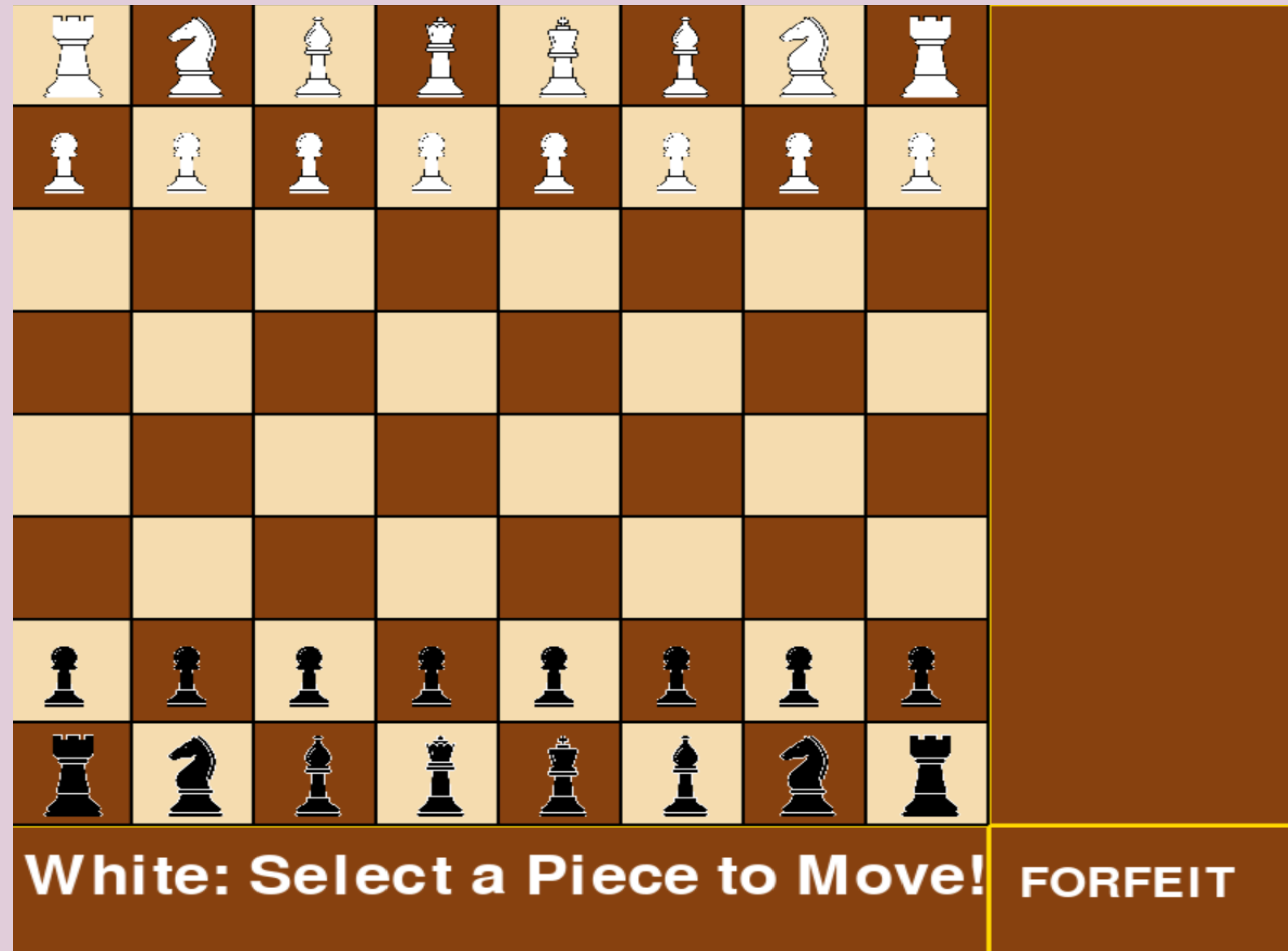
d. **Pawn:** Moves forward; captures diagonally.

e. **King:** Moves one square in any direction.

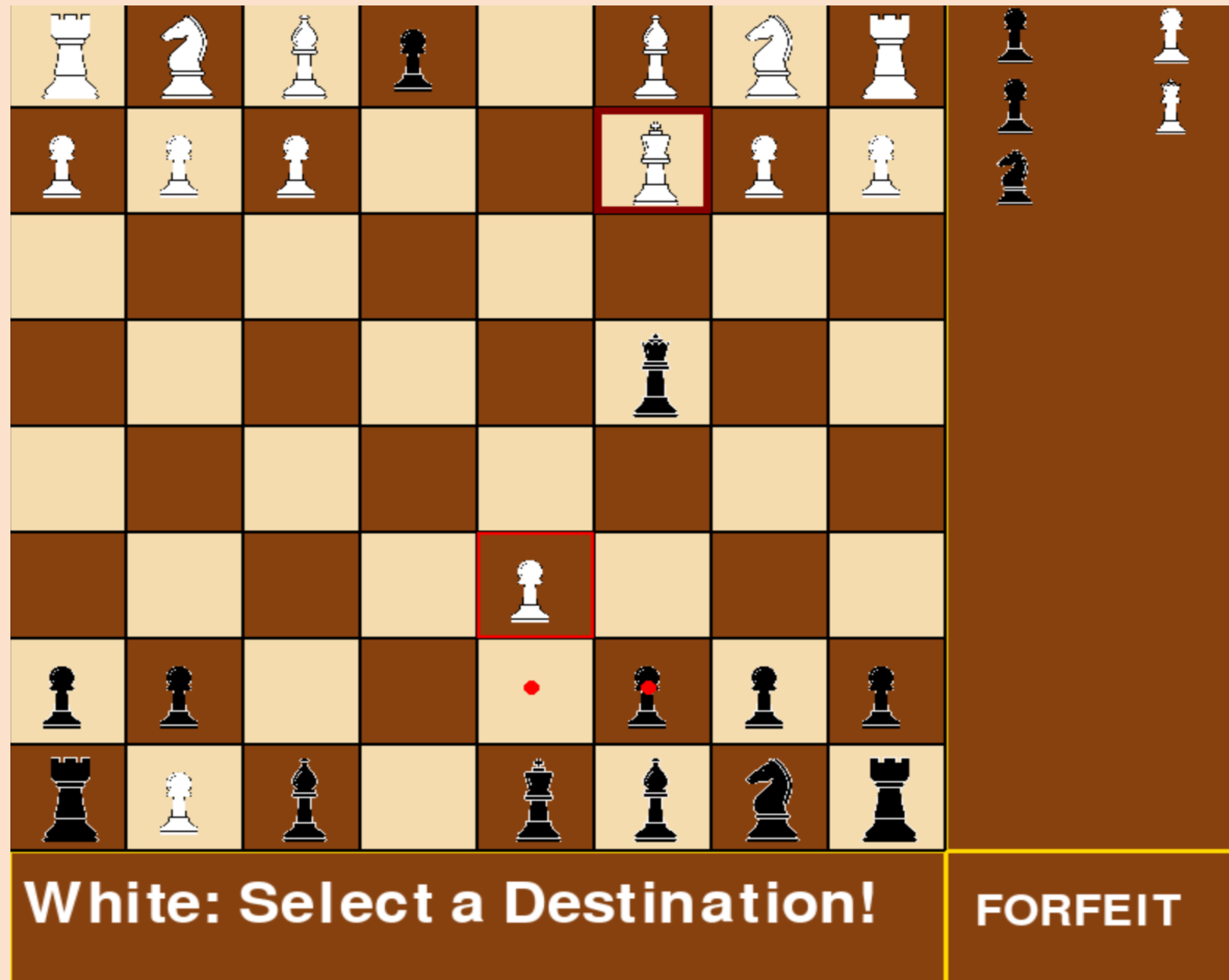
f. **Queen:** Combines Rook and Bishop moves



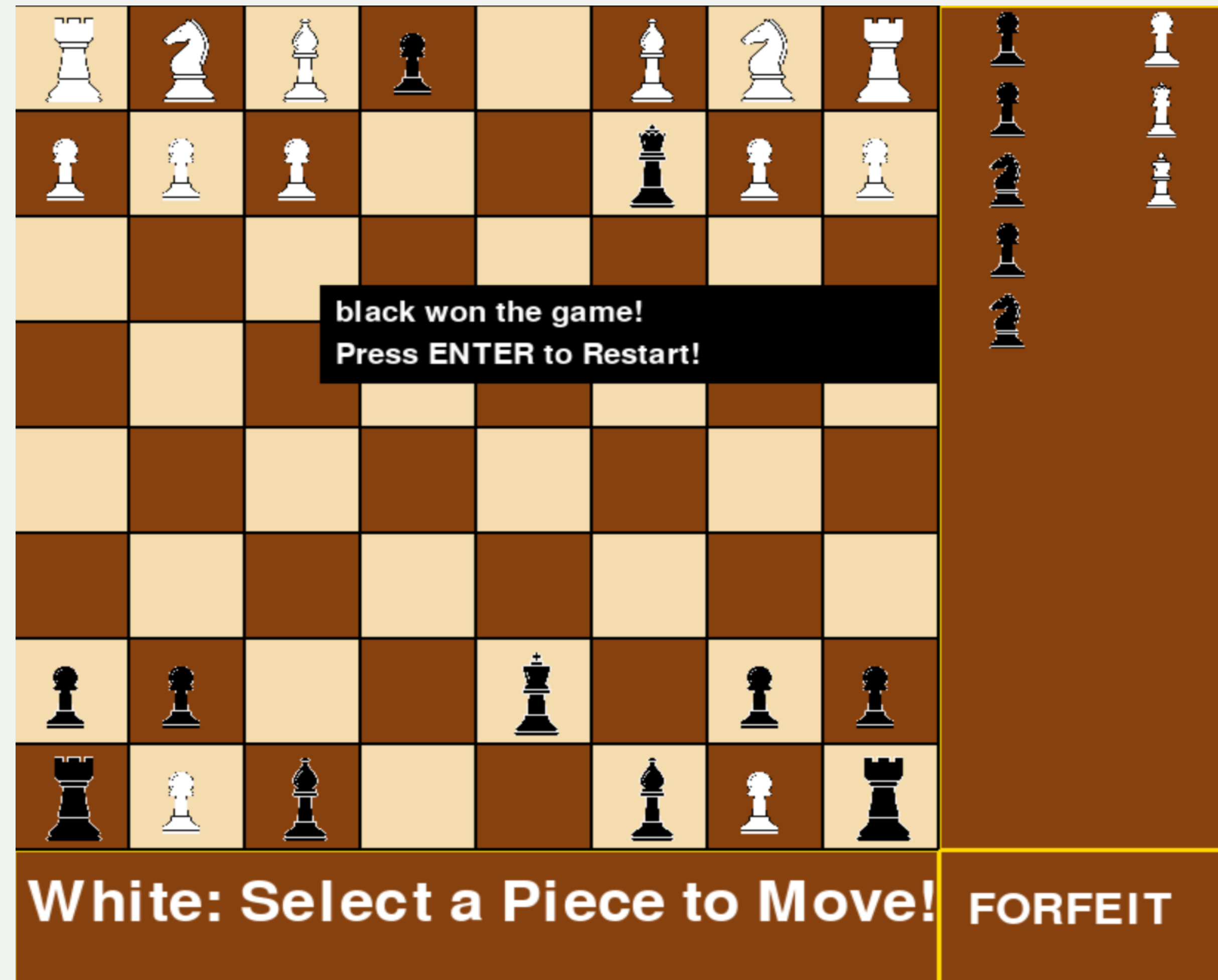
User Interface



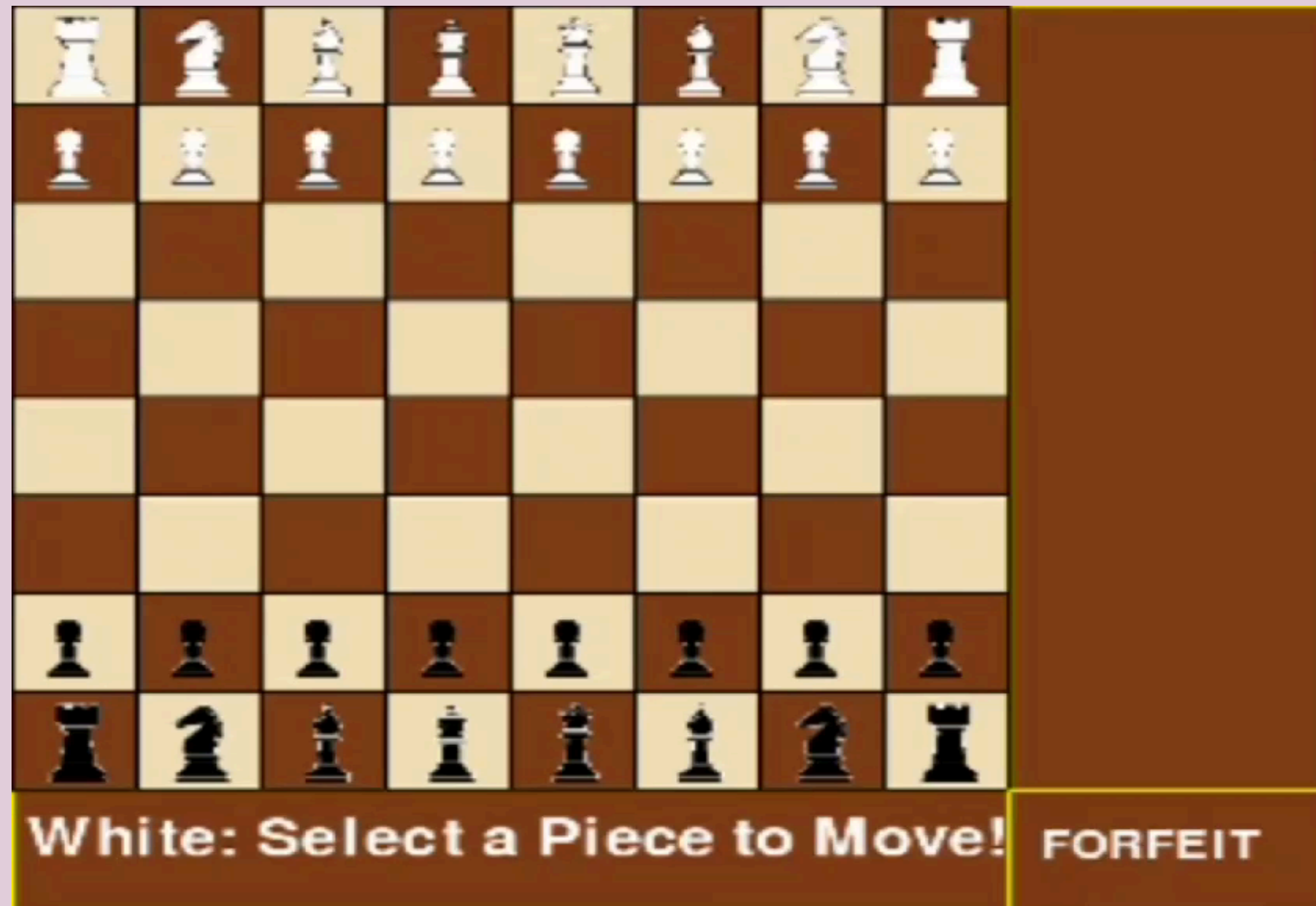
User Interface (Continued)



User Interface (Continued)



A Glimpse of the Game



Challenges

- 1.Implementing complex rules like en passant and castling.
- 2.Designing an effective heuristic evaluation function.
- 3.Optimizing AI performance for faster move calculations.

Future Work

- 1.Adding difficulty levels for AI.
- 2.Improving heuristics using machine learning techniques.
- 3.Enabling multiplayer or online gameplay features.

Conclusion

- ✓ Developed a chess-playing AI with full rule support
- ✓ Explored and implemented key algorithms for strategic decision-making
- ✓ Gained experience in game development and AI programming



Thanks for Playing!

See you next time!



Q&A Session !