Project Title: Chess with Enhanced AI and Strategic Variations
Submitted By: Syed Hussamuddin (22K-4658), Usman Sohail (22K-4487), Syed Muqeet ur Rehman (22K-4477)
Course: AI
Instructor: Miss Syeda Ravia Ejaz
Submission Date: 15th May 2025

---

1. Executive Summary

Project Overview:
This project aimed at implementing a strategic variant of traditional chess by adding a second queen to each player's starting lineup. To manage the increased complexity and provide a challenging opponent, we developed an AI model using the Minimax algorithm with Reinforcement Learning. This hybrid AI system allows the game to adapt and respond intelligently to a wider range of strategies.

---

2. Introduction

Background:
Chess is a classical two-player strategy game played on an 8x8 board, where each player starts with 16 pieces, including a single queen. We chose to modify chess because of its structured rules and AI research history. Our innovation introduces a second queen for each player at the start, significantly increasing tactical complexity and altering traditional opening and midgame strategies.

Objectives of the Project:

- Develop an AI-enhanced chess engine capable of playing with two queens per side.
- Implement and compare Minimax and Reinforcement Learning models.
- Test the AI against human players to evaluate performance under the modified rules.
- Provide adjustable difficulty levels via AI depth and RL training states.

---

3. Game Description

Original Game Rules:
Standard chess involves two players each controlling 16 pieces (1 king, 1 queen, 2 rooks, 2 knights, 2 bishops, and 8 pawns). The objective is to checkmate the opponent's king by placing it under direct attack with no legal moves left.

Innovations and Modifications:

- Each player begins with **two queens**, with the second queen replacing the d-pawn.

- This increases the game's strategic depth, especially in the opening and midgame phases.
- AI algorithms were modified to account for this additional high-value piece.

---

4. AI Approach and Methodology

AI Techniques Used:

- **Minimax**: Enables efficient search and decision-making by pruning unpromising move branches.
- **Reinforcement Learning**: Trains an AI agent using reward feedback from self-play to make dynamic decisions based on experience.

Algorithm and Heuristic Design:

- Piece values: Pawn = 1, Knight/Bishop = 3, Rook = 5, Queen = 9, King = very high value.
- Heuristics consider material advantage, piece mobility, control of central squares, and king safety.
- The evaluation function combines these factors to score board states.

AI Performance Evaluation:

- The Minimax AI performs consistently with an average move decision time of 1-2 seconds.
- The RL-based AI demonstrates adaptive behavior after sufficient training, especially in endgames and complex positions.
- Performance was evaluated via simulated matches and win rates against human testers.

---

5. Game Mechanics and Rules

Modified Game Rules:

- Each player starts with two queens instead of one.
- The second queen replaces the pawn on the d-file (4th column from the left for white).
- All standard chess rules (castling, check, en passant, promotion) remain unchanged.

Turn-based Mechanics:

- Players alternate moves.
- In single-player mode, the AI makes a move immediately after the player.

Winning Conditions:

- A player wins by checkmating the opponent's king.
- Draw, stalemate, and other standard conditions apply.

---

6. Implementation and Development

Development Process:

- The project started by implementing a standard chess engine using Python and Pygame.
- Then, we integrated Minimax for the AI.
- The game logic was modified to accommodate a second queen at startup.
- A Reinforcement Learning agent was implemented using self-play and reward-based evaluation.

Programming Languages and Tools:

- Programming Language: Python
- Libraries: Pygame, NumPy
- Tools: GitHub for version control and collaboration

Challenges Encountered:

- Handling increased branching factor in Minimax due to an extra queen.
- Designing a balanced heuristic function that accounts for unconventional strategies.
- Training the reinforcement learning agent to understand new queen dynamics without overfitting.

---

7. Team Contributions

- **Syed Hussamuddin:** Developed the Minimax and Reinforcement Learning algorithms.
- **Usman Sohail:** Designed and implemented the two-queen variant logic and board setup.
- **Syed Muqeet ur Rehman:** Focused on the user interface and integration of AI into gameplay.

---

8. Results and Discussion

AI Performance:

- The Model AI achieved a win rate of 85–90% against beginner and intermediate human players.

---

9. References

- Pygame Documentation
- Python Chess Programming Tutorials (YouTube and blogs)
- Research papers on Reinforcement Learning in board games
- Online articles on Minimax and Alpha-Beta Pruning