Chess with Enhanced AI and Strategic Variations

Submitted By: Syed Hussamuddin(22K-4658) | Usman Sohail(22K-4487) | Syed Muqeet ur Rehman(22K-4477) | Ashar Zamir(22K-4241)
Course: AI
Instructor: Miss Syeda Ravia Ejaz
Submission Date: 12th March 2025

1. Project Overview

Project Topic:

This project involves developing a fully functional chess game using Python and Pygame. The AI will use the Minimax algorithm with Alpha-Beta pruning to make strategic decisions. Additionally, we will introduce a chess variant with two queens per player instead of one, adding a new layer of complexity and strategy.

Objective:

The main goal is to implement an AI-driven chess game using Minimax and Alpha-Beta pruning. The variant with two queens per player aims to challenge players with new strategic possibilities, requiring the AI to adapt accordingly.

2. Game Description

Original Game Background:

Chess is a two-player strategy game played on an 8x8 board with 16 pieces per player. The objective is to checkmate the opponent's king by placing it under direct attack with no legal moves to escape.

Innovations Introduced:

- **Two Queens Variation:** Each player starts with two queens instead of one.
- **Modified Strategy:** AI must adapt to the presence of two queens, making evaluation more complex.
- **Advanced AI with Minimax and Alpha-Beta Pruning:** The AI will optimize move selection using search-tree pruning.

- **Customizable Difficulty Levels:** AI depth and heuristics can be adjusted to suit player skill levels.

Impact on Gameplay Complexity and Strategy:

- Increased game depth due to two queens changing early-game and mid-game dynamics.
- New checkmate patterns and strategies.
- More challenging AI decision-making due to an expanded range of powerful moves.

3. AI Approach and Methodology

AI Techniques to be Used:

- **Minimax Algorithm** with Alpha-Beta Pruning for efficient move selection.
- **Heuristic Evaluation Function** to assess board positions based on material balance, piece mobility, king safety, and control of the board.
- **Optional Reinforcement Learning:** A future extension may involve training the AI using self-play.

Heuristic Design:

- **Piece Value:** Assign values to pieces (Pawn = 1, Knight/Bishop = 3, Rook = 5, Queen = 9, King = high value).
- **Positional Advantage:** Evaluate control over central squares and opponent's weaknesses.
- **Mobility:** Favor positions where more pieces have legal moves.
- **King Safety:** Prioritize moves that keep the king safe from immediate threats.

Complexity Analysis:

- **Minimax with Alpha-Beta Pruning:** O(b^d) reduced to O(b^(d/2)) complexity.
- **Challenges:**
  - Handling the added complexity of two queens.
  - Efficiently managing computational overhead while maintaining strong AI performance.

4. Game Rules and Mechanics

Modified Rules:

- Each player starts with **two queens** instead of one.
- The second queen replaces one of the original pawns at the start.
- All other chess rules remain unchanged.

Winning Conditions:

- The game follows standard chess checkmate rules.
- Stalemate and draw rules remain unchanged.

Turn Sequence:

- Players alternate turns as in standard chess.
- AI evaluates and makes optimal moves when playing in AI mode.

5. Implementation Plan

Programming Language:

- Python

Libraries and Tools:

- **Pygame** for rendering the board and GUI.
- **NumPy** for handling game state data.
- **Custom AI algorithms** implemented using Python.

Milestones and Timeline:

- **Week 1-2:** Game structure setup, board rendering, and piece movement implementation.
- **Week 3-4:** Implement Minimax with Alpha-Beta Pruning.
- **Week 5-6:** Add and test the two-queens chess variation.
- **Week 7:** Debugging, AI fine-tuning, and user interface enhancements.
- **Week 8:** Final testing, performance optimizations, and documentation.

6. References

- Python Pygame Documentation
- AI in Chess: Minimax and Alpha-Beta Pruning (Various online resources)
- Reinforcement Learning in Chess AI Research Papers
- YouTube Tutorial: Chess AI with Minimax and Pygame