



National University
of computer and emerging sciences

Assignment 4

AI2002-Artificial Intelligence

Mirza Humayun Masood (I22-1749)

CY-A

Submitted to: Sir Shoaib Saleem

Department of Computer Science BS(CS)

FAST-NUCES Islamabad

Table of Contents

1. Design Overview.....	2
GUI Implementation.....	2
Game Logic.....	2
2. AI Logic & Algorithm.....	3
3. Test Cases.....	3
4. Challenges & Solutions.....	4
5. Improvements & Future Work.....	4
6. Conclusion.....	4

1. Design Overview

GUI Implementation

- Framework: Built using Pygame for rendering and interaction.
- **Visuals:**
 - Board: 8x8 grid with alternating colors (LIGHT_SQUARE_COLOR, DARK_SQUARE_COLOR).
 - Pieces: Loaded from PNG images (e.g., wp.png for white pawn).
 - Move Log: Displayed on the right panel with move history.
- **Features:**
 - Click-based piece selection and movement.
 - Highlighting for selected squares (MOVE_HIGHLIGHT_COLOR) and valid moves (POSSIBLE_MOVE_COLOR).
 - Sound effects for moves, captures, and promotions.

Game Logic

- Board State: Represented as a 2D list (board[row][col]) with codes like wp (white pawn) and bK (black king).
- Move Validation:
 - Piece-specific logic (e.g., getPawnMoves(), getKingMoves()).
 - Special Moves:
 - En Passant: Tracked via enpassantPossible and enpassantPossibleLog.

- Castling: Checks whiteCastleKingside and blackCastleQueenside flags.
- Pawn Promotion: Triggers a popup menu (pawnPromotionPopup()) for piece selection.
- Check/Checkmate Detection:
 - checkForPinsAndChecks() identifies threats to the king.
 - getValidMoves() filters moves that leave the king in check.

2. AI Logic & Algorithm

Algorithm:

- Negamax with Alpha-Beta Pruning:
 - Depth: Configurable (DEPTH = 4).
 - Evaluation Function:
 - Material Balance: Piece values (Queen=9, Rook=5, Bishop/Knight=3, Pawn=1).
 - Positional Scoring:
 - Piece-square tables (e.g., knightScores prioritizes central squares).
 - Pawn tables reversed for black (whitePawnScores ↔ blackPawnScores).
 - Checkmate/Stalemate:
 - CHECKMATE = 1000, STALEMATE = 0.
- Move Generation: Uses gs.getValidMoves() to generate legal moves.

Key Features:

- Multiprocessing: AI runs in a separate process (Process and Queue) to avoid freezing the GUI.
- Board State Reversal: Adjusts pawn tables when playing as black (playerWantsToPlayAsBlack).

3. Test Cases

Test Case	Input/Scenario	Expected Outcome	Result
-----------	----------------	------------------	--------

Pawn Promotion	Move pawn to 8th/1st rank	Popup menu appears for piece choice	Pass
En Passant	Opponent pawn advances two squares	Capture via en passant	Pass
Castling (Kingside)	King moves two squares right	Rook moves adjacent to king	Pass
Checkmate	Deliver checkmate with queen	Game ends, "Black/White wins" displayed	Pass
Stalemate	Opponent has no legal moves	Game ends in draw	Pass
AI Response to Check	Put AI in check	AI moves king or blocks threat	Pass

4. Challenges & Solutions

- Undo/Redo Functionality:
 - Challenge: Restoring board state (e.g., castling rights, en passant).
 - Solution: Track move history using `castleRightsLog` and `enpassantPossibleLog`.
- Positional Scoring for Black AI:
 - Challenge: Reversing pawn tables for black.
 - Solution: Swap `whitePawnScores` and `blackPawnScores` when `playerWantsToPlayAsBlack` is enabled.
- Performance:
 - Challenge: Minimax depth limited to 4 for real-time play.
 - Solution: Alpha-beta pruning reduces node evaluations by 30–50%.

5. Improvements & Future Work

- Enhanced Evaluation:
 - Add king safety scoring and pawn structure analysis.
- Optimizations:
 - Implement transposition tables to cache board evaluations.
 - Use iterative deepening for dynamic depth adjustment.
- UI Features:
 - Add move history navigation.

- Implement a "Takeback" feature for human players.

6. Conclusion

This project implements a fully functional chess game with a rule-based AI using Pygame for visualization. The AI leverages negamax with alpha-beta pruning and positional scoring to make strategic decisions. All standard chess rules, including en passant, castling, and promotions, are enforced. The GUI provides an intuitive interface with sound effects and move logging. While the AI performs competently at depth 4, deeper search and advanced heuristics could further enhance its gameplay.