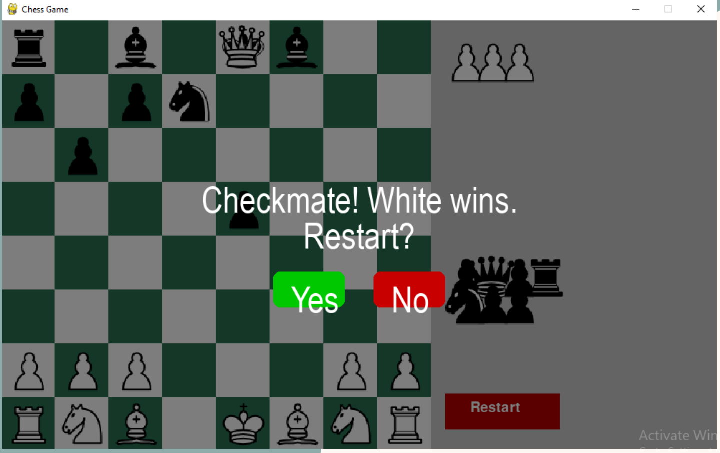
AI-Powered chess game

BY: Youssef Yasser 22-101089
Badr Mohamed 22-101178
Omar Khaled 22-101132
Youssef Elmenshawy 22-101241



Checkmate! White wins. Restart?

Challenges & Possible Enhancements:

-Challenges: Building a chess game from scratch without using chess libraries requires custom implementation of complex rules, AI, and UI, while ensuring accurate move validation, checkmate detection, and performance optimization.

-Enhancements: Adding features like improved AI, special moves, multiplayer mode, difficulty levels, mobile support,also optimizing the agent's decision-making speed—can make the game more engaging, realistic, and accessible for players of all skill levels.

How It Works:

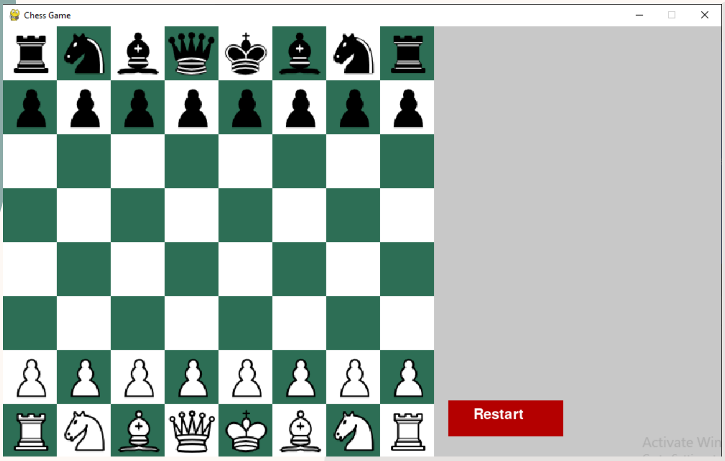
-Chessboard: The game initializes with a standard 8x8 chessboard using the Pygame library

-Rules Function: Validates moves based on chess rules, ensuring pieces move correctly and don't capture allies

-AI Agent: Uses Minimax with Alpha-Beta pruning to evaluate moves based on material value, piece activity, king safety, pawn structure, mobility, and central control.

-Check & Checkmate: After each move, the game checks if the king is in check or checkmate, displaying alerts or a "Game Over" message with restart/quit options.

-Game loop: Continuously processes player moves, AI moves, and checks for check/checkmate while updating the board.



Features:

1-Interactive Chessboard

2-Sidebar tools: Restart button, Captured pieces

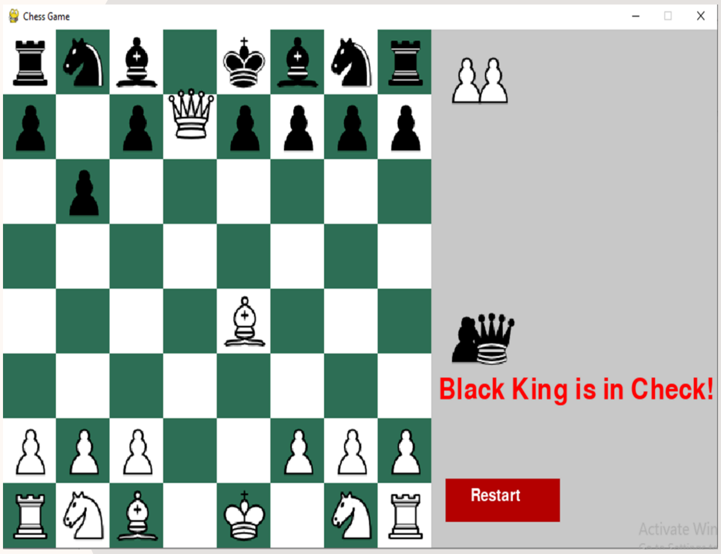
3-AI opponent

4-Checkmate & Check alertions

5-Evaluation Metrics

Objective

Our objective was to develop a chess game that incorporates AI algorithms learned in our course, such as the alpha-beta pruning algorithm and heuristic evaluation, to create a challenging and intelligent opponent. By implementing these techniques, we sought to demonstrate a deep understanding of game tree search and decision-making processes in AI, while also providing an interactive and visually engaging user experience through Pygame.



Black King is in Check!