Checkers AI Game

CHALLENGE THE AI OR PLAY WITH FRIENDS!
BUILT WITH PYGAME

📌 Project Overview

- Classic Checkers game with AI opponent
- Modular Python structure
- Easy to run & extend
- Designed for learning AI & game development

🎮 Game Modes

- Human vs Human
- Human vs AI (AI powered by Minimax + Alpha-Beta)

AI Features

- Minimax algorithm with Alpha-Beta pruning
- Adaptive search depth (depth 4 default)
- Evaluation function considers:
  - Piece count (men vs kings)
  - Board control & positional advantage
  - Capture opportunities
- Generates all possible moves for each piece
- Chooses best move based on evaluation score

Utility Functions

- Board.evaluate():calcules board score
- Positive → WHITE advantage
- Negative → RED advantage
- Position bonus for pieces closer to promotion
- Kings weighted higher

Rules

-DIAGONAL MOVES ONLY
-MANDATORY JUMPS & MULTIPLE JUMPS
-KING PROMOTION
-TURN-BASED PLAY
-WIN CONDITIONS: NO LEGAL MOVES OR NO PIECES

AI DECISION TREE

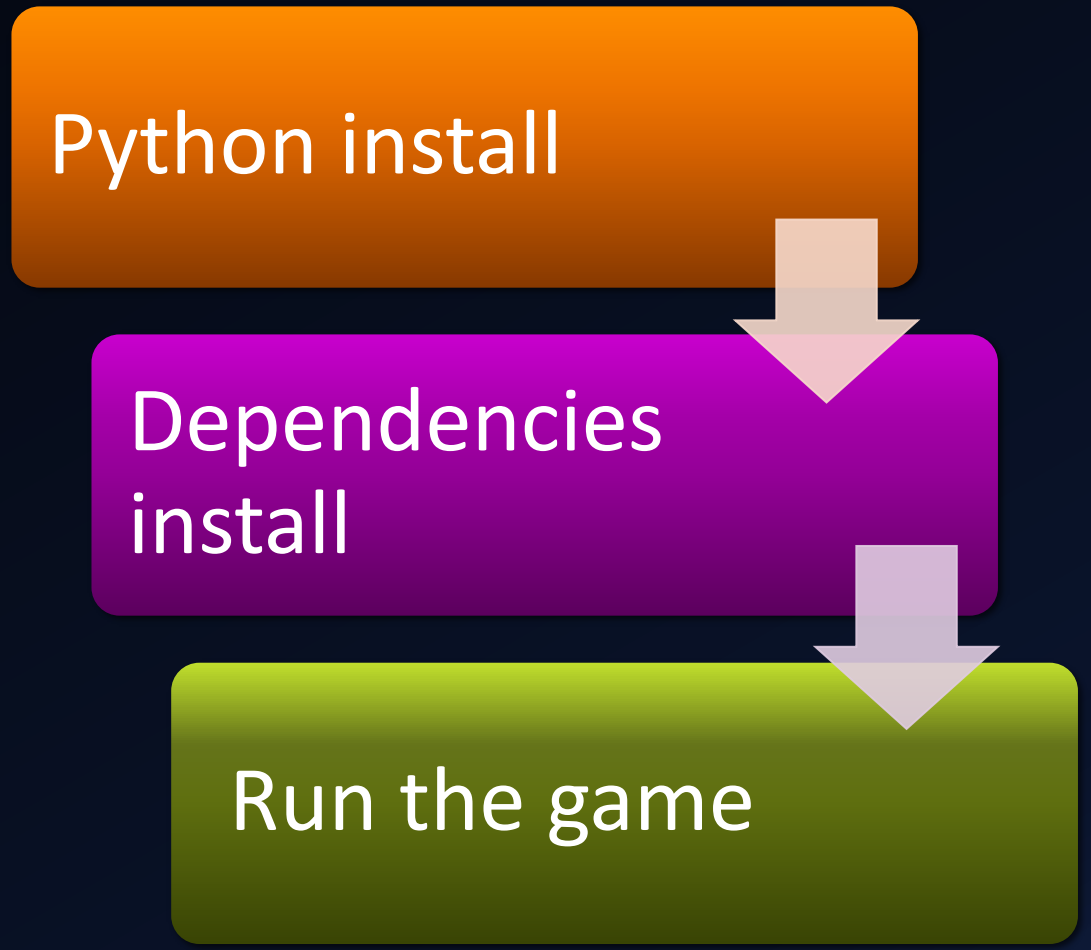
Visualize Minimax as tree:

- Root = current board state
- Nodes = possible moves
- Leaves = board evaluation

Alpha-Beta pruning cuts unnecessary branches

🚀 Getting Started

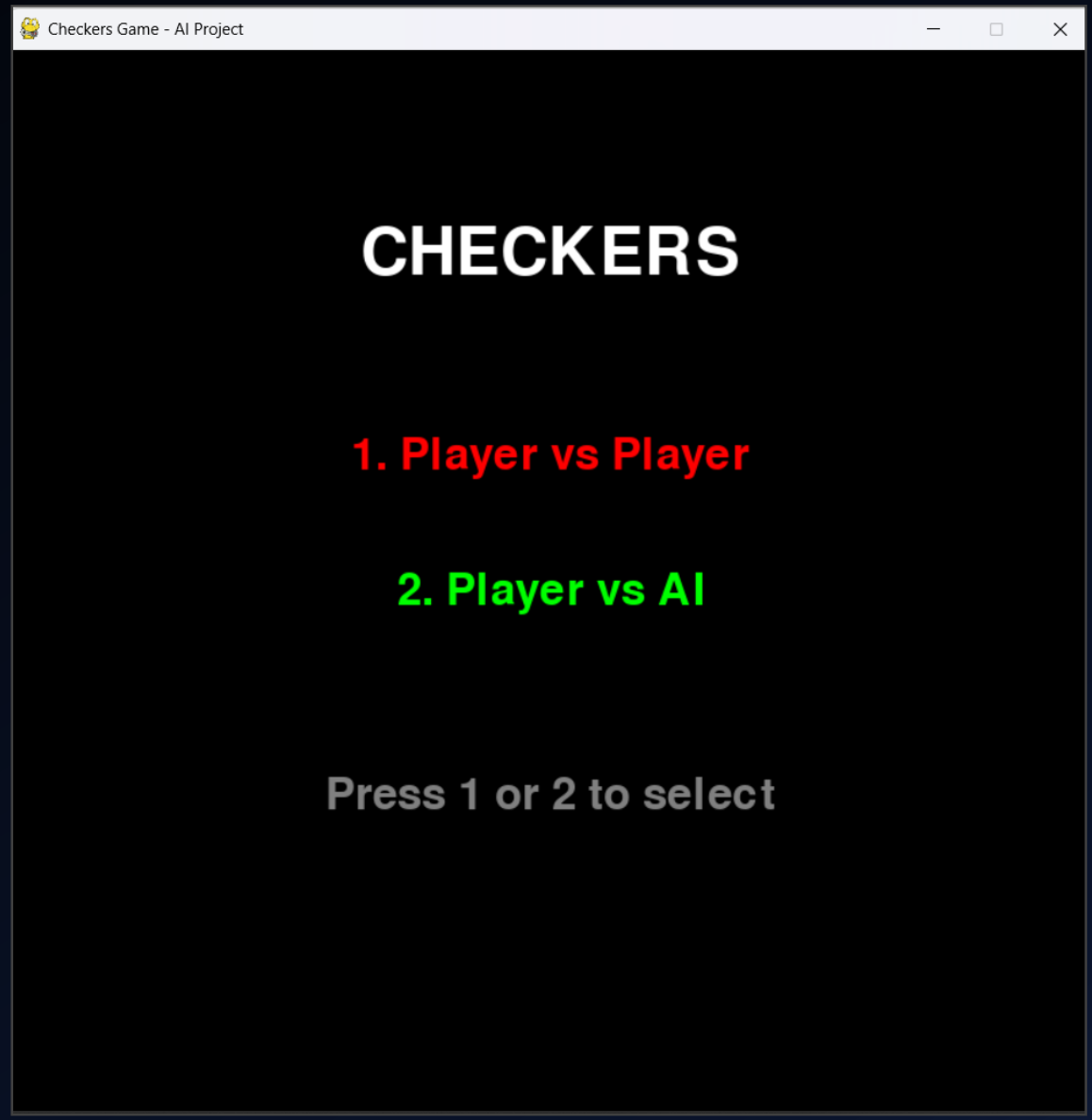
- Ensure Python 3.9+ is installed
- pip install pygameThird
- Run the game:
- python main.py



Game interface

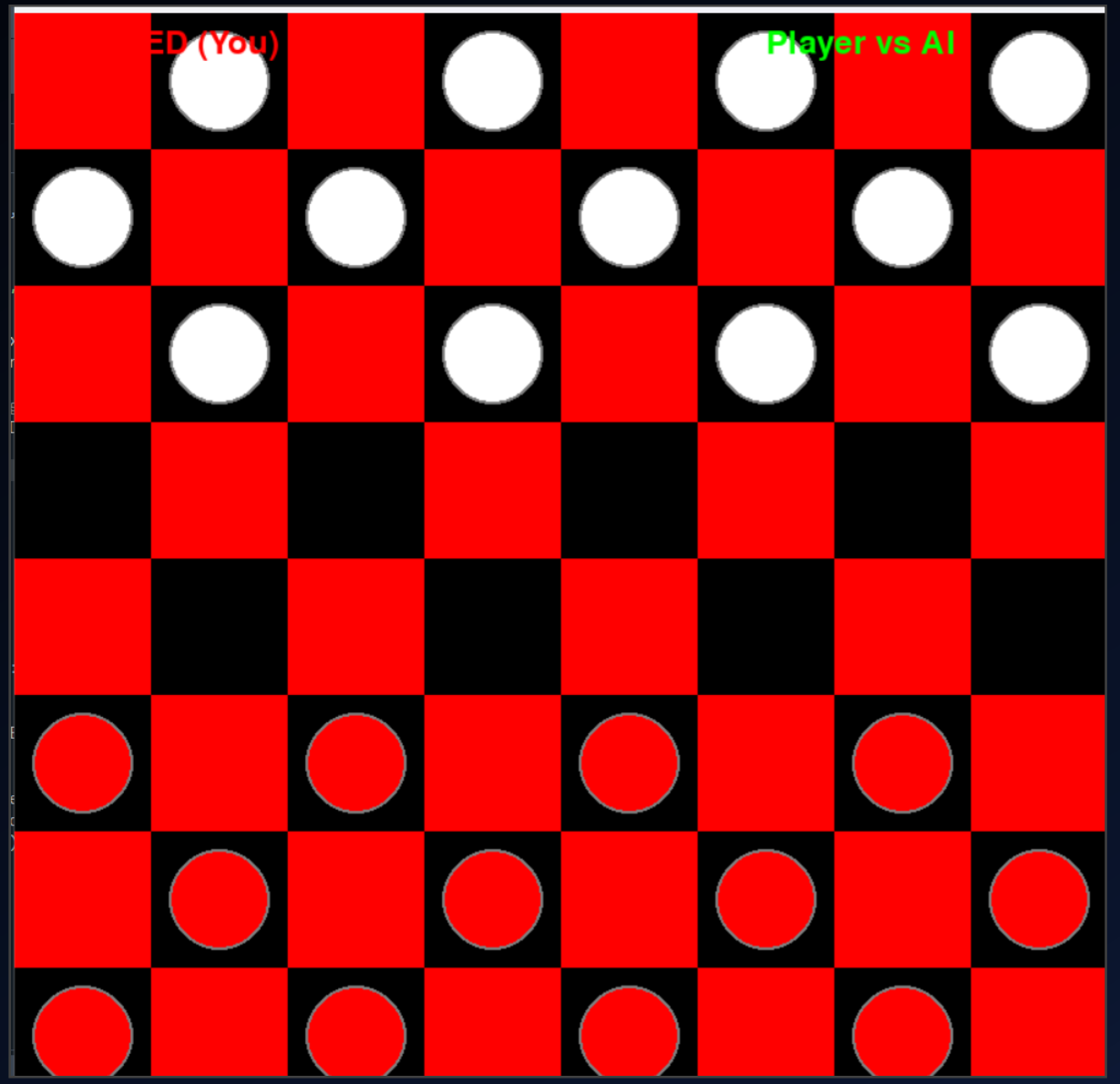
MAIN MENU

Shows the game’s main menu where players can choose the game mode (Human vs Human or Human vs AI). Clear interface with start button and title.



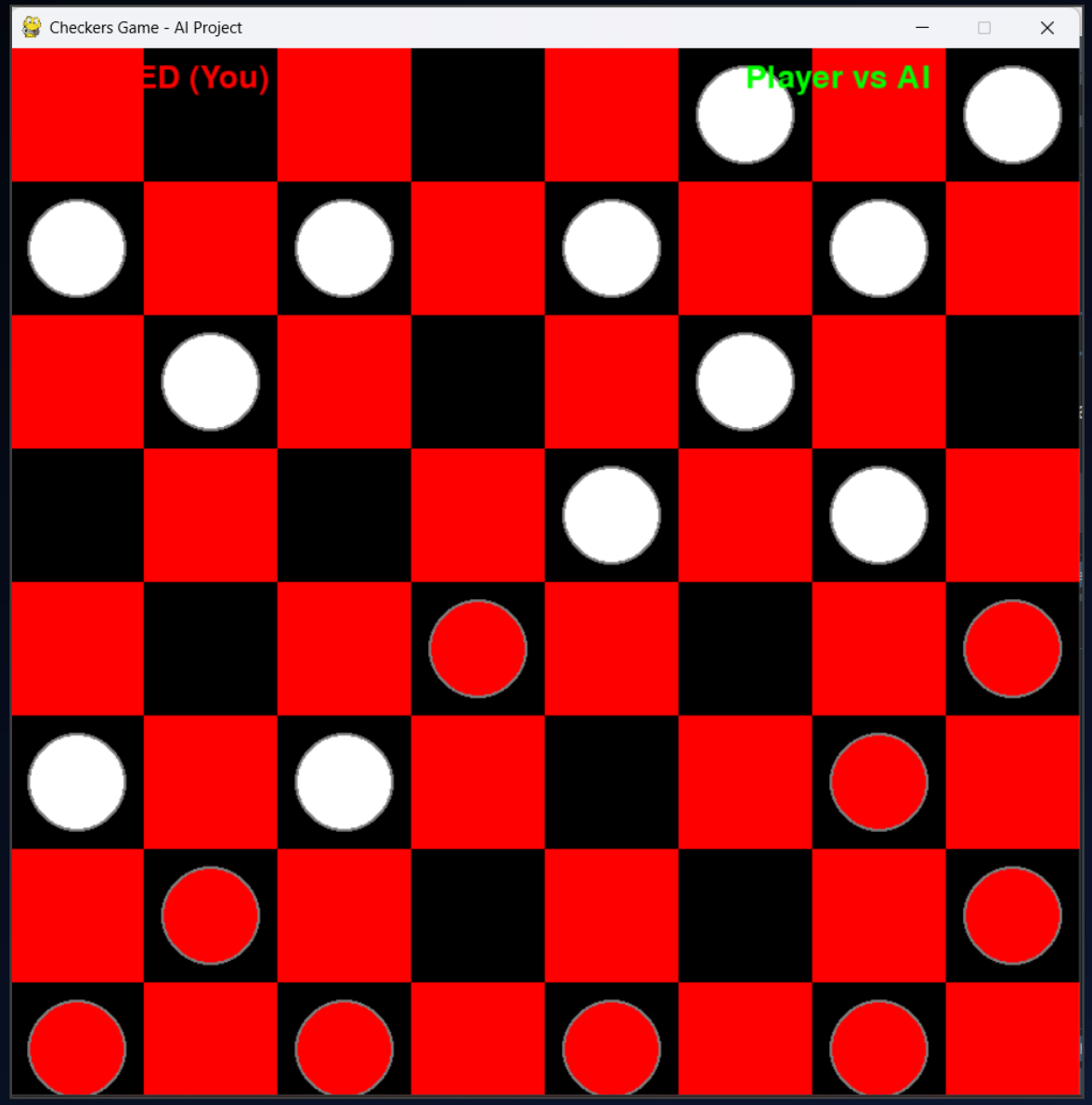
GAME PLAY

an active game session on the 8x8 Checkers board. Highlights player pieces, AI opponent moves, and possible valid moves. Demonstrates real-time gameplay mechanics.



END GAME

Shows the end of a match when one player wins or no moves are left. Highlights the winning player and displays final board state. Can include “Game Over” message or notification.



Contributions

- Doaa Ahmed
- Goda Ahmed
- Mohamed Hafez
- Zeyad Amer
- Mohamed Khaled