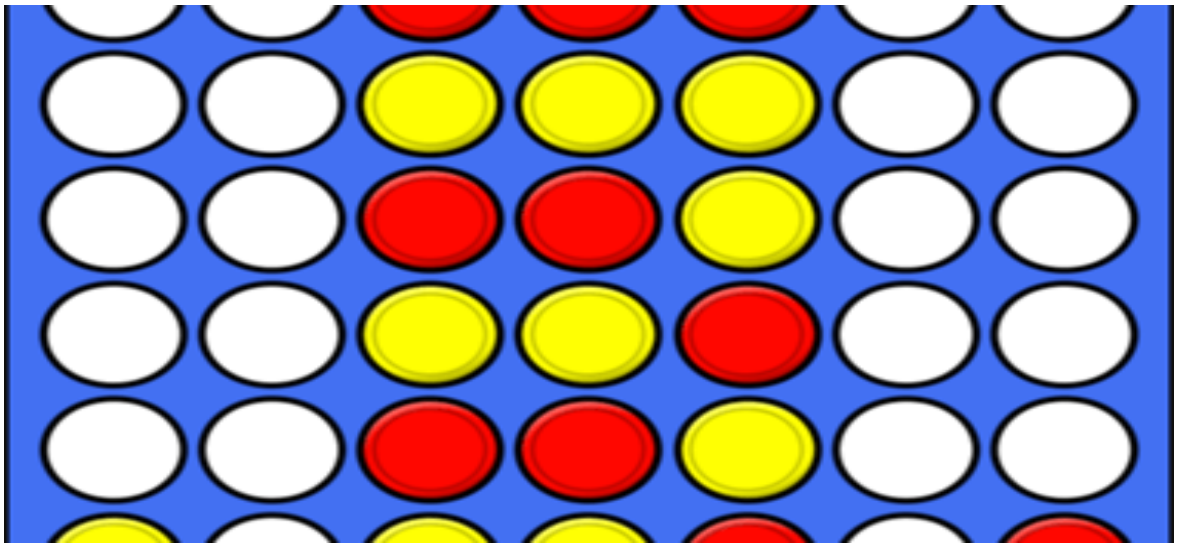
Connect 4



Team Members

Abdallah Mostafa Barakat (Ai) Section 2
Abdallah Sayed Awad (Ai) Section 2

Supervised By:

Eng / Youssef El-Baroudy

1. Project Overview

This project is a Connect 4 game where a human player can compete against an AI agent.
The AI uses Adversarial Search (Minimax with Alpha-Beta Pruning) to make strategic moves.

Objectives:
- Implement a playable game with GUI.
- Use an AI agent that makes intelligent moves.
- Demonstrate understanding of adversarial search strategies.

2. Files Description

board.py

This file contains the Board class which manages the game state.

Class: Board
- __init__(self) : Initializes an empty 6x7 board.
- drop_piece(row, col, piece) : Places a piece (1 = Human, 2 = AI) at the specified position.
- is_valid_location(col) : Checks if a column is available for a move.
- get_next_open_row(col) : Returns the lowest empty row in the given column.
- winning_move(piece) : Checks if the given piece has 4 in a row (horizontal, vertical, diagonal).
- is_full() : Returns True if the board has no empty spaces.

minimax.py

This file contains the AI logic using Minimax with Alpha-Beta Pruning.

Constants:
- AI_PIECE = 2
- HUMAN_PIECE = 1

Functions:
- score_position(board, piece) → Evaluates the board for the given piece.
- evaluate_window(window, piece) → Helper function to calculate score for 4 consecutive cells.
- minimax(board, depth, alpha, beta, maximizingPlayer) → Returns the best column

for AI.
- maximizingPlayer = True → AI turn
- maximizingPlayer = False → Human turn
- Uses Alpha-Beta pruning to reduce unnecessary calculations.

game.py
This file handles the GUI and interaction using Tkinter.

Class: Connect4GUI
- __init__(self, root) → Initializes the GUI and sets up the game.
- draw_board() → Draws the current state of the board on the canvas.
- handle_click(event) → Handles Human player clicks, updates the board, then triggers AI move.
- show_message(msg) → Displays a popup message for Game Over with a Restart button.
- restart_game() → Resets the board and redraws it for a new game.

Gameplay Features:
- Human plays with red pieces, AI plays with yellow pieces.
- Game Over popup appears for Human win, AI win, or Draw.
- Restart button allows replay without closing the window.

main.py
This file runs the game.

Explanation:
- Creates a Tkinter window.
- Initializes the Connect4GUI class.
- Starts the main loop to keep the game running.

3. How to Run
1. Make sure Python is installed.
2. Make sure all four files are in the same folder:
   - board.py
   - minimax.py
   - game.py
   - main.py
3. Open terminal/command prompt in the folder.

4. Run the game:
   python main.py
5. Click on a column to drop your piece. The AI will respond automatically.

4. GitHub Contribution

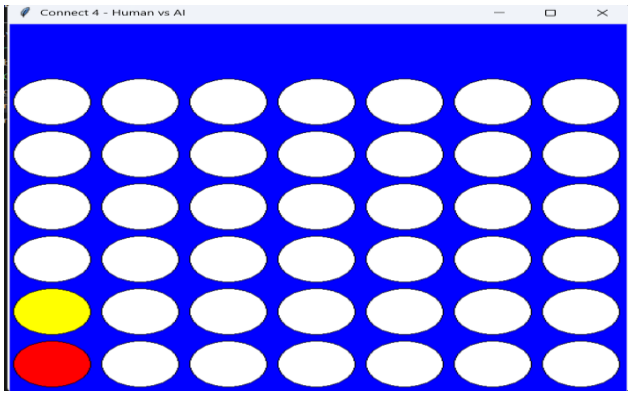
- Abdullah Mostafa: board.py + game.py
- Abdullah Sayed : minimax.py + main.py + documentation

5. Notes

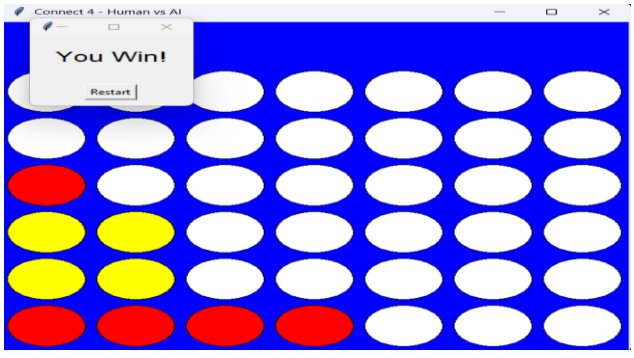
- AI uses Minimax + Alpha-Beta for fast, intelligent moves.
- The game fully supports Human vs AI with restart functionality.

6- screenshots of the game

When the user plays (the red one):



When human win:



When Ai win: