**Project Title:**  
Connect 4 Evolution

**Submitted By:**  
22k-4322 Owais Aamir

22k-4562 Hassaan Akhtar

22k-4310 Anas Ashfaq

**Course:**  
AI

**Instructor:**  
Sir Abdullah Yaqoob

**Submission Date:**  
March 10, 2025

**1. Project Overview**

* **Project Topic:**  
  This project involves developing an innovative version of Connect 4 with two distinct modes:
  1. A three-player variant featuring two human players and one AI (with modified rules and a multi-player minimax approach)
  2. A traditional two-player mode with AI vs. human.  
     The aim is to enhance the complexity and strategic depth of Connect 4 by introducing unconventional turn sequences and modifying the win-detection and evaluation heuristics accordingly.
* **Objective:**  
  The primary goal is to implement strategic AI opponents using a modified minimax algorithm with alpha‑beta pruning to handle both conventional two-player scenarios and a more complex three-player environment. The project will explore heuristic evaluation modifications and multi-player search strategies (using a “paranoid” approach) to balance play in the three-player variant.

**2. Game Description**

* **Original Game Background:**  
  Connect 4 is a classic two-player connection game in which players take turns dropping colored discs into a vertically suspended grid. The objective is to be the first to connect four discs in a row horizontally, vertically, or diagonally.
* **Innovations Introduced:**
  + **Dual-Mode Gameplay:**
    - **Three-Player Mode:** In this variant, the game supports two human players and one AI. The board remains a 7-column by 6-row grid, but the turn sequence is modified so that each player (with distinct colors) plays in rotation. The AI uses a modified minimax algorithm that assumes the two human opponents act in unison to minimize the AI’s advantage.
    - **Two-Player Mode:** This mode retains the classic Connect 4 rules but pits a human against an AI opponent using conventional minimax with alpha‑beta pruning.
  + **Impact on Complexity and Strategy:**
    - In the three-player mode, the introduction of an additional human player and a modified turn order increases the branching factor of the game tree and forces the AI to evaluate threats from two opponents simultaneously.
    - The heuristic evaluation functions are adjusted to assess board positions in this multi-player scenario, creating a more challenging and unpredictable gameplay experience.

**3. AI Approach and Methodology**

* **AI Techniques to be Used:**
  + **Minimax Algorithm:**
    - For the two-player mode, a standard minimax algorithm with a depth-limited search is used.
    - For the three-player mode, a modified (paranoid) minimax approach is implemented, where the AI assumes that both human opponents collaborate to minimize its outcome.
  + **Alpha‑Beta Pruning:**
    - Both modes use alpha‑beta pruning to reduce the search space and optimize the decision-making process.
  + **Optional Techniques:**
    - Future iterations may explore reinforcement learning or Monte Carlo Tree Search (MCTS) to further enhance AI performance.
* **Heuristic Design:**
  + **Evaluation Function:**
    - The evaluation function scores board positions based on the number of connected pieces (windows) for both the AI and its opponents.
    - Special emphasis is placed on center column control and threat detection (e.g., three connected pieces with an open space).
  + **Multi-Player Consideration:**
    - In the three-player mode, the heuristic penalizes positions where either human opponent is close to forming a winning line, assuming that they may work together against the AI.
* **Complexity Analysis:**
  + The time complexity of the minimax algorithm grows exponentially with depth. In the three-player variant, the branching factor increases due to additional opponent moves.
  + Optimizations using alpha‑beta pruning and a careful selection of search depth (e.g., depth 4–5) are critical to maintain real-time performance.
  + Challenges include designing robust heuristics that fairly evaluate positions in a multi-player context and managing increased computational complexity.

**4. Game Rules and Mechanics**

* **Modified Rules:**
  + **Three-Player Mode:**
    - The board remains a standard 7×6 grid.
    - Three players (Player 1: Human with red pieces, Player 2: Human with yellow pieces, AI: Green pieces) take turns in a fixed rotation.
    - When it is a human player’s turn, the game registers mouse input for move selection.
    - The AI computes its move using the modified minimax algorithm.
  + **Two-Player Mode:**
    - Traditional Connect 4 rules are followed, with one human player facing the AI.
    - Turns alternate between the human and the AI.
* **Winning Conditions:**
  + A player wins by connecting four of their pieces in a row horizontally, vertically, or diagonally.
  + In the three-player mode, the game immediately terminates when any player forms such a connection.
  + A draw is declared if the board is filled with no winning move.
* **Turn Sequence:**
  + **Three-Player Mode:**
    - The turn order is fixed (e.g., Player 1 → AI → Player 2) with the order randomized at game start.
  + **Two-Player Mode:**
    - Turns alternate between the human player and the AI.

**5. Implementation Plan**

* **Programming Language:**  
  Python
* **Libraries and Tools:**
  + **Pygame:** For building the graphical user interface and handling user input.
  + **NumPy:** For efficient board representation and manipulation.
  + **Optional AI Libraries:** TensorFlow or Scikit-learn may be considered for future reinforcement learning experiments.
* **Milestones and Timeline:**
  + **Week 1-2:**
    - Finalize game design and rules for both modes.
    - Create initial prototypes for board representation and basic move logic.
  + **Week 3-4:**
    - Develop and test the minimax algorithm with alpha‑beta pruning for the two-player mode.
    - Extend the algorithm to support three-player gameplay using a paranoid approach.
  + **Week 5-6:**
    - Implement and refine the graphical user interface using Pygame.
    - Integrate game mechanics with the AI logic.
  + **Week 7:**
    - Comprehensive testing of both game modes.
    - Fine-tune heuristics and AI decision-making.
  + **Week 8:**
    - Final testing, debugging, and documentation.
    - Prepare the final project report and presentation materials.

**6. References**

* Russell, S., & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach.*
* Pygame Documentation: <https://www.pygame.org/docs/>
* NumPy Documentation: <https://numpy.org/doc/>
* Research articles on minimax algorithm, alpha‑beta pruning, and multi-player game AI strategies.