|  |  |  |  |
| --- | --- | --- | --- |
| **Project Title** | Connect\_4 Game | | |
| **Project Code** | **AIS201** | **Course Name** | **Artificial Intelligence** |
| **Professor** | **Hala Zayed** | | |
| **TA** | **Ali Abdelmageed** | **Mentor Name** | Text. |
| **Team Name** | **Achieve Intelligence** | | |
| **Team Members** | **Aya Elneanaei Fouda** | Text. | Text. |
| **Nada Hussain mokhtar** | Text. | Text. |
| **Problem Summary** | We implemented a Connect 4 game with three difficulty levels: easy, medium, and hard. At the easy level, the game uses a random algorithm to make moves. The computer player selects a random column to drop its token into, without considering any strategic decision-making. At the medium level, the game uses the alpha-beta pruning algorithm. This algorithm enhances the search process by evaluating the best possible moves based on a scoring system. It explores the game tree, considering both maximizing and minimizing moves, and discards branches that are guaranteed to be worse than previously explored branches. On the hard level, the game uses the Minimax algorithm. This algorithm exhaustively evaluates all possible moves and their outcomes by simulating the game until its conclusion. It considers all possible moves and their consequences in choosing the best move for the computer player, assuming optimal play from both players. These difficulty levels provide varying levels of challenge for players, with the hard level employing a more sophisticated strategy to maximize the computer player's chances of winning. | | |
| **Methodology** | We implement a Connect Four game with a graphical user interface (GUI) implemented using the Tkinter library. The game has three levels (easy, medium, hard), and there is also an option to play a typical game against an AI opponent or with another player or play Ai against Ai.  **The random algorithm at an easy level:**  A random algorithm is a simple approach to playing Connect 4, requiring minimal strategy and experimentation. It's suitable for beginners and It's important to note that a random algorithm is not likely to be very effective against more skilled players or more advanced algorithms that use strategic decision-making. Here is how it works:  The computer player selects a random column to drop its token into, without considering any strategic decision-making. chances of winning.  **alpha-beta pruning algorithm at the medium level:**  The alpha-beta pruning algorithm is a powerful technique for searching game trees and finding optimal moves in games with multiple alternative moves. It reduces the search tree size and eliminates the need to analyze all potential game pathways.  Here is how it works:  The algorithm starts by exploring the game tree, evaluating each node using a heuristic function. It tracks alpha and beta values, which represent the best maximum and minimum values. If a node has a score below beta, it prunes that branch, and if it finds a higher alpha value, it prunes that branch. The algorithm updates alpha and beta values until it reaches the maximum depth or prunes unlikely branches. It then chooses the move with the highest score based on the evaluation function and returns it as the best move for the current player.  **Minimax algorithm on the hard level:**  The minimax algorithm is used in game theory and decision-making to determine the best course of action provided that the adversary plays effectively as well.  Here is how it works:  The algorithm constructs a game tree representing possible moves and outcomes from the current board state. It evaluates each node using a heuristic function, assigning a score based on the current position of pieces. The algorithm alternates between maximizing and minimizing the score at each level, choosing the move with the highest score for the maximizing player and the lowest score for the minimizing player. The algorithm explores the tree recursively until it reaches a terminal state, where it assigns a score based on the heuristic function. The algorithm propagates the scores back up the tree, choosing the move with the highest score for the maximizing player and the lowest score for the minimizing playe | | |
| **Achievements and Skills Gained** | **1. Learning How to use the Tkinter library**  **2. Learning algorithms [ minimax- alpha and beta ]**  **3. GUI from Connect 4 game.** | | |

|  |  |
| --- | --- |
| **Project Title** | Connect\_4 Game |
| **Main Results** | Our GUI:  You can play with Ai or another player for each level.  connect 4_1  GUI for playing Connect Four offers three difficulty levels: easy, medium, and hard  connect 4_2  final |
| **Discussion and Conclusion** | The GUI for playing Connect Four offers three difficulty levels: easy, medium, and hard. Each level uses different algorithms and strategies to provide a unique gameplay experience. The GUI allows players to engage with the game and explore strategies and decision-making. It enhances the accessibility and enjoyment of Connect Four, showcasing the potential of artificial intelligence and game theory algorithms. The GUI encourages further exploration and innovation in intelligent gameplay systems for board games like Connect Four. |
| **References** | [1] Vandewiele, G. (2019, January 7). Creating the (nearly) perfect connect-four bot with limited move time and file size. Medium.  [2] Kuo, J. C. (2022, March 30). Artificial Intelligence at Play — Connect Four (Mini-max algorithm explained). Medium.  [3] Shah, S. (2021, July 11). Alpha Beta pruning. A Blog About Our Findings and Musings.  [4] Connect 4 - AI. (2021, May 9). Devpost. |
| **Future Work and Suggestions** | We will use an algorithm at the easy level instead of a random algorithm that makes the computer player selects a random column to drop its token into, without considering any strategic decision-making. |
| **Group Photo** | C:\Users\aelne\Pictures\IMG_2395 (1).JPGIMG_2395 (1) |