**Project 7: Intelligent Connect-6 Player Documentation Report**

**Team members:**

**Introduction:**

introduced in 2003 by Professor I-Chen Wu at Department of Computer Science and Information Engineering, National Chiao Tung University in Taiwan as his personal master's thesis, is a two-player strategy game like Gomoku. It became very famous in the year 2006 as it was included in the 2006 Computer Olympiad as one of the games there. In which Professor Wu, won gold in the Connect6 tournament at the 11th Computer Olympiad. (Yang\_Ming\_Chiao\_Tung\_University, 2024)

**Project Idea in Detail**

The Connect 6 game is a modern twist on the traditional Connect-N games like Connect 4 and Go Moku. Designed for two players, rules: the game is played on a grid where each player alternates turns placing two stones on the board, except for the first move where the first player places only one stone. The objective is to form a line of six consecutive stones (horizontally, vertically, or diagonally) while preventing the opponent from doing the same. Game fairness: Since each party has one more piece of the board than the other party after each move, the game can naturally reach a balanced state, which greatly improves fairness. Unlike many chess types such as backgammon, chess, and chess, playing first has some advantages. Complexity: If Connect6 uses an infinite board, both state-space and game-tree complexities are infinite as well. If a19 standard board is used. The game-tree complexities for connect 6 is much higher than those in Go and Renju, since many more moves are possible placing two stones rather than one. The number of movies increases exponentially with board size. (Yang\_Ming\_Chiao\_Tung\_University, 2024)

The project aims to develop a software implementation of Connect 6 with an emphasis on interactivity and strategic depth using 3 main algorithms minimax, alphabet, and heuristic. It will feature an intuitive user interface, an AI opponent with varying algorithms, and options for multiplayer mode

**Main Functionalities:**

* **Simple Gui for selecting algorithms:** a decent UI that will let you interact flawlessly with the application.
* **Interactive Game Board Interface:** A visually appealing, grid-based GUI where players can place their stones and view the progress of the game in real time.
* **Single-player Mode:** An AI opponent with selectable difficulty levels that uses strategic algorithms to simulate intelligent gameplay.
* **Multiplayer Mode:** A local multiplayer mode allowing two players to play on the same device.
* **Game Rules Validation:** Automatic enforcement of game rules, including the placement of stones and detection of illegal moves.
* **Win Detection Algorithm:** Efficient algorithms to detect winning conditions, ensuring that games conclude correctly and promptly.
* **Custom Settings:** Options to configure grid size, game rules and visual themes for stones (can be changed in the code itself).

**Similar Applications in the Market:**

Several applications and board games share similarities with Connect 6 but there is very few working connect 6 applications or website that lets you try and play the actual game:

Apart from games like:

* **Go**: A highly strategic game with an emphasis on territory control. Although more complex, it shares elements of strategic depth with Connect 6.
* **Go Moku**: A Connect-5 variant that involves forming a line of five stones, often played on a 15x15 board. Connect 6 expands on this by requiring six stones and allowing two moves per turn, creating more dynamic strategies.

**Grand games arena (**[**https://arena.grandgames.net/en/connect6 )** : was](https://arena.grandgames.net/en/connect6%20)%20:%20was)the only website that lets you play vs an ai or an algorithm with various difficulty

**There were sites like:**

* **(**[**https://www.yucata.de/en/GameInfo/Connect6**](https://www.yucata.de/en/GameInfo/Connect6) **)**
* **(** [**https://www.ludoteka.com/clasika/games?game=connect6&hizk=en**](https://www.ludoteka.com/clasika/games?game=connect6&hizk=en)**)**
* **(**[**https://boardgamegeek.com/boardgame/22847/connect6**](https://boardgamegeek.com/boardgame/22847/connect6) **)**

All claim to have the game but none of them was working which made the process of trying and knowing more about this game challenging.so I was forced to go and look at any other source that can help me.

**Review of Academic Published papers:**

The following academic papers provide insights and methodologies that helped me through the development of the Connect 6 game:

* **Relevance-Zone-Oriented Proof Search for Connect6:** This was written by the inventor of the game himself which helped me a lot to under stand the base lines of the game from rules to strategies and algorithms that can be used. (I-Chen Wu & Ping-Hung Lin, 2010)
* **Deep learning approaches to the game of Connect6:** this paper helped me to understand how the minimax algorithm can be used in this game. ( Yang , Yen, & Kavak, 2040)
* **Comparing Approaches to Connect6: Temporal difference vs. Minimax:** The paper discusses the minimax algorithm and how it could be used in this game (Wilson, 2007)
* **Alpha-Beta Pruning in Mini-Max Algorithm -An Optimized Approach for a Connect-4 Game:** Tho this paper ins not implementing the connect 6 game but still is was very useful to help me better understand the alpha b-beta pruning and how it could be implemented. (Chaudhary, Pandey, & Khera, 2021)
* **Study of Road-based Versus Pattern-based Game Strategies in Connect6:** this paper helped me to understand how I can check if any player have won the game and helped me in implement some strategies that the ai can use to win. (Cao, Zhang, Jia, Shang, & Wang, 2024)
* **Connect6:** this paper helped me assembling the whole project together and understand the game more deeply. ( Wu, Huang, & Chang, 2005)

**Dataset Employed:**

The game’s AI does not require a pre-existing dataset but instead uses heuristic rules and search-based algorithms to determine its movements. For more advanced AI, synthetic data can be generated by simulating thousands of games between AI agents to identify optimal strategies.

**Details of the Algorithm(s)/Approach(es) Used and Results of Experiments**

* **Algorithms:**
  + **Minimax:** This algorithm evaluates possible moves up to a certain depth, aiming to maximize the AI’s chances of winning while minimizing the opponent’s opportunities. However, the computation time was found to be approximately **60 seconds** to determine two moves, making it less practical for real-time gameplay.
  + **Alpha-Beta Pruning:** An optimization of the Minimax algorithm, it reduces the number of nodes evaluated in the search tree. This significantly improved performance, reducing decision-making time to **16 seconds** for two moves without compromising strategic quality.
  + **Two Heuristic-Based Strategy:** A lightweight approach focusing on controlling the center of the board and following the player lead. It evaluates board states based on predefined criteria, prioritizing central positions for better control. This method achieved the fastest performance, taking only **13 seconds** to compute two moves, making it ideal for quicker gameplay.
  + **Win Detection Algorithm:** A linear-time algorithm scans the board to identify six consecutive stones in all possible directions (horizontal, vertical, and diagonal).

**Development Platform:**

* **Programming Language**: Python
* **Libraries and Tools**:
  + **Tkinter**: For creating the graphical user interface.
  + **NumPy**: For efficient grid management and operations.

**Project overview:**

The project was written fully on python, using oop and file imports:

The main files are:

* **UI:**

This part handles the graphical user interface for the game. It includes two main classes:

UI: Manages the menu and the general game setup.

Connect6: Represents the board and the gameplay, with logic to handle different game modes.

* **Board:**

This class is responsible for creating the game board. It is used in other classes to manage the board's state and facilitate gameplay.

* **Logic:**

Contains the core game logic, such as switching players, placing pieces on the board, and other essential operations. This is the central part of the app, ensuring smooth gameplay.

* **Minimax:**

Implements the Minimax algorithm to determine the best move for the AI. The algorithm evaluates the game board and looks for the optimal position for the AI, considering a depth of 3. The evaluation of the board is done by the evaluate board method, and the AI makes decisions based on this.

* **Alphabet (Alpha-Beta Pruning):**

An optimization of the Minimax algorithm, reducing the time complexity by pruning branches that do not need to be explored. This method is much faster than Minimax while providing almost the same output, making it more efficient.

* **Heuristic 1 and Heuristic 2:**

These are improved versions of the evaluate board method used in Minimax and Alpha-Beta algorithms.

Heuristic 1: Focuses on controlling the center of the board, increasing the number of open ends, and thereby improving the chances of winning.

Heuristic 2: Focuses more on the opponent's move patterns, aiming to block or counter their strategic placements.

The project emphasizes efficiency, with the use of smarter and faster algorithms like Alpha-Beta pruning and Heuristics to improve the AI's decision-making process. This design ensures a smoother, faster game experience while maintaining strategic depth.

**Reflection and improvements:**

At the end of developing this game, I learned a lot about game development and AI, and why it's so hard to win against it. This project was a huge success for us because we:

* **Learned how to create game logic**: We successfully implemented the algorithms and improved them over time.
* **Gained insights into AI's effectiveness:** We understood why AI is so strong at games, as it can anticipate and calculate moves ahead, making it tough to beat.

**However, we faced several challenges:**

* **AI Strategy**: We struggled with determining the exact steps the AI should take to win and correctly implementing the win conditions and defense strategies.
* **Performance Issues:** The program runs slower compared to other games, mainly due to:
  + **Python's slower processing power:** Being a higher-level language, Python isn't as efficient as lower-level languages.
  + **Brute-force approach in Minimax:** The Minimax algorithm evaluates every possible outcome, which makes it slow and inefficient.

In terms of improvements, I recognize areas where we could enhance the game:

* **Optimizing AI Calculation:** We could have the AI return two moves at once to improve time efficiency, but this would not give us the best 2nd move.
* **Function Optimization:** Some functions were being called multiple times in a row, which caused latency. We could optimize these functions to reduce the load and improve performance.

Soon, I plan to:

* **Fix performance issues:** By optimizing the code, refining algorithms, and exploring ways to improve the AI's efficiency.

A diagram of a computer program

Description automatically generated**Diagrams:**

A diagram of a game

Description automatically generated

A diagram of a algorithm

Description automatically generated

A diagram of a diagram

Description automatically generated

A diagram of a diagram

Description automatically generated

# Works Cited

Wu, C., Huang, D.-Y., & Chang, H.-C. (2005, 1). Connect6. *The Journal of the Computer Games Community*, pp. 235-242.

Yang , J.-K., Yen, S.-J., & Kavak, S. (2040). Deep learning approaches to the game of Connect6. *Entertainment Computing*.

Cao, Y., Zhang, M., Jia, S., Shang, Y., & Wang, X. (2024, 4 10). Study of Road-based Versus Pattern-based Game Strategies in Connect6. *Frontiers in Computing and Intelligent Systems*, pp. 38-41.

Chaudhary, S., Pandey, S., & Khera, H. (2021). *Alpha-Beta Pruning in Mini-Max Algorithm -An Optimized Approach for a Connect-4 Game.* Galgotias University.

I-Chen Wu, & Ping-Hung Lin. (2010). Relevance-Zone-Oriented Proof Search for Connect6. *IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES*, 191-207.

Wilson, I. (2007). *Comparing Approaches to Connect6:Temporal difference vs. Minimax.*

Yang\_Ming\_Chiao\_Tung\_University. (2024, 12 15). *connect6.org*. Retrieved from connect6.org: http://connect6.org/index.php/mainmenu-27/mainmenu-32?task=view&id=20