FALL 2024

PROJECT REPORT

TERM PROJECT

PRESENTED BY :

Team YYHOS

Table Of Contents

About Project

This report documents the design, implementation, and testing of a "Connect 4" game developed using Python and the Pygame library. The game allows a single player to compete against an AI opponent, incorporating features such as a graphical user interface, game state saving, and a minimax algorithm for AI decision-making.

Motivation

The Connect 4 game was selected as a project due to its engaging nature and the opportunity it provides to explore several key programming concepts. The project combines game development, algorithm design, and GUI programming, offering a challenging yet rewarding experience for improving problem-solving and coding skills. Additionally, the project demonstrates how theoretical concepts like the minimax algorithm can be applied in practical scenarios.

Analysis & Design

Functions of project:

1)initialize_board:

Input: None
Return: A 7x6 matrix initialized with empty spaces
Task: Creates and returns an empty game board.

2)draw_board:

Input: Pygame screen object, game board
Return: None
Task: Draws the game board and player moves on the screen.

3)is_valid_column:

Input: Game board, column index
Return: Boolean
Task: Checks if a column is valid for a move.

4)get_next_open_row:

Input: Game board, column index
Return: Row index or None
Task: Finds the next available row in the given column.

5)update_board:

Input: Game board, row index, column index, piece
Return: None
Task: Updates the game board with the given piece.

6)check_win:

Input: Game board, player piece
Return: Boolean
Task: Checks for a winning condition.

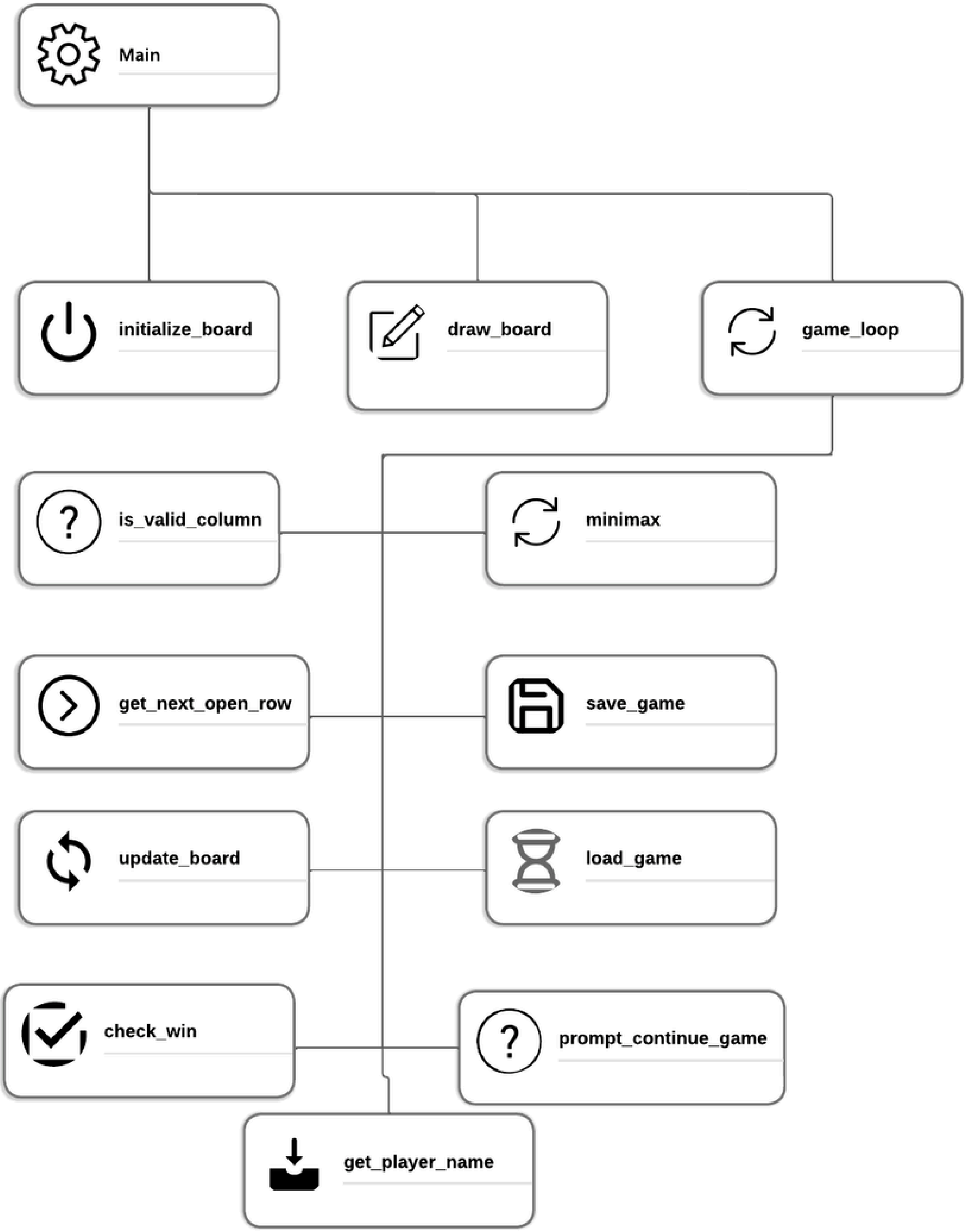
7)minimax:

Input: Game board, depth, maximizing_player flag
Return: Best column, score
Task: Implements the minimax algorithm for AI decision-making.

8)game_loop:

Input: Pygame screen object, game board, player name, turn
Return: None
Task: Handles the main game loop, player moves, and AI moves.

Hierarchy Chart



Description

Explanation of the Theory Behind the Code:

1)Game Board Representation:

The game board is represented as a 2D list where rows and columns correspond to the 7x6 grid of the "Connect 4" game. Each cell can be empty, occupied by the player, or occupied by the AI.

2)AI Decision-Making:

The minimax algorithm evaluates possible moves by simulating the game tree to a certain depth. It maximizes the AI's chances of winning while minimizing the player's chances. The evaluation function assigns high scores for winning moves and low scores for losing scenarios.

3)Graphical User Interface: Pygame is used to create a visually appealing interface with colorful graphics. Players interact with the game through mouse clicks, while the AI's moves are automatically displayed.

Work Distribution

Team Members:

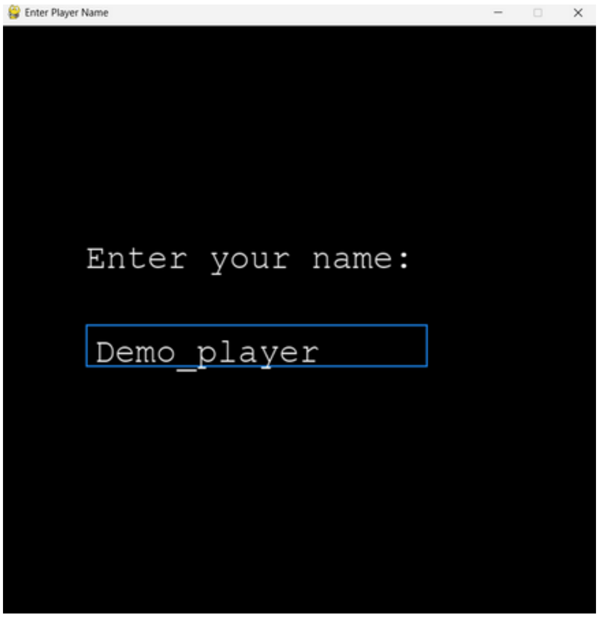
1. **Yassin Mohamed:**
   - **Designed the minimax algorithm for AI decision-making.**
   - **Developed I/O file functionalities**
   - **Developed Pause menu**

2. **Hana Ahmed:**
   - **Developed the Board design and its functionalities using pygame library**

3. **Yehia Ghanem:**
   - **Developed the Win detection using check_win function**

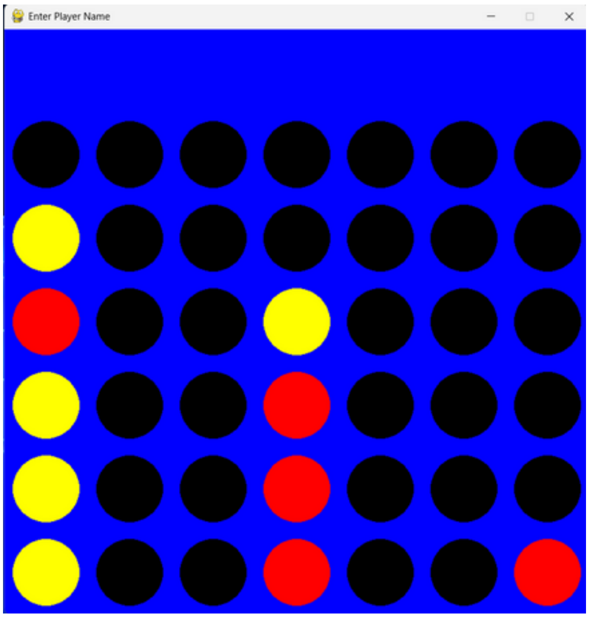
4. **Sedra Othman:**
   - **Developed Game Loop and User Interaction using game_loop and pygame**

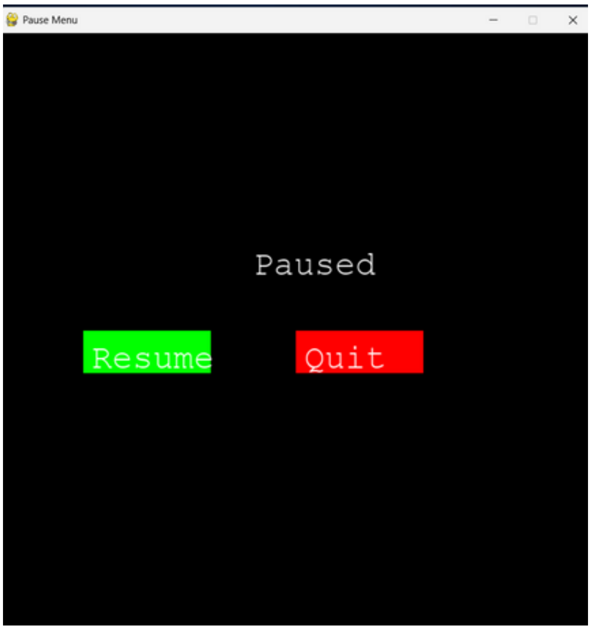
Test Cases & Sample screens



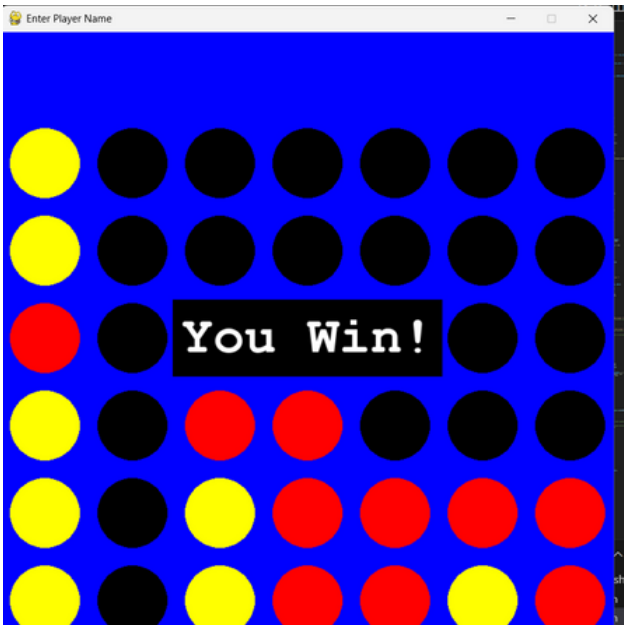
Enter your name:

Demo_player





Paused



You Win!

Challenges and Problems

AI Optimization:

Initial versions of the minimax algorithm were slow due to evaluating the entire game tree. Limiting the depth improved performance.

Graphical Glitches:

Some graphical elements did not update correctly during rapid moves. This was fixed by carefully managing screen refreshes.

File I/O:

Ensuring compatibility across different operating systems for saving and loading files required additional testing.

Conclusion and Discussion/ suggestions of improvements/ future work

The “Connect 4” Game was a fruitful task and helped in gaining alot of knowledge and information on how to incorporate team work in accomplishing such a tough milestone in programming.

Future improvements:

- Enhancing the AI with more advanced algorithms like alpha-beta pruning.

- Adding multiplayer support.

- Improving the graphical interface with animations and Leaderboards

List of references

1.Pygame Documentation: https://www.pygame.org/docs/

2.Keith Galli, an MIT graduate YouTuber : https://www.youtube.com/@KeithGalli

3. Stack Overflow: Pygame Optimization Tips: https://en.wikipedia.org/wiki/Connect_Four