Project Title:

Halma Checkers Gameplay with AI using Minimax Algorithm

Submitted By:

M. Mufrah Jawaid (22k-5046)

Group Members:

- Yousuf Hassan (22k-5051) -> Group Leader
- M. Mufrah Jawaid (22k-5046)
- Abdul Mateen Muddasir (22k-5070)

Instructor:

Aalishba Subhani

Course:

Artificial Intelligence

Submission Date:

10th March 2025

Project Description

This project is a Python implementation of the board game Checkers Gameplay, using the Pygame library for the graphical user interface. Checkers Gameplay is a two-player strategy board game invented in the 19th century. This version features an 8x8 board where a player competes against an AI opponent. The goal of the game is to move all pieces from the starting corner to the opposite corner before the opponent.

The AI opponent utilizes the Minimax algorithm with Alpha-Beta Pruning to make its moves strategically. The game displays valid potential moves for the player, ensuring smooth gameplay.

Motivation

This project was chosen to implement key Artificial Intelligence concepts, such as game search algorithms, recursion, and heuristic evaluations, in a real-world application. It also helps strengthen Object-Oriented Programming skills and deepen understanding of AI decision-making strategies.

Features

- **Single-player Mode:** The player competes against an AI opponent.
- **AI with Minimax Algorithm & Alpha-Beta Pruning:** The AI makes optimal decisions by simulating future moves.
- **Graphical User Interface (GUI):** The game is visually interactive using Pygame.
- **Move Validation System:** Ensures only valid moves are allowed.
- **Game Board Evaluation:** AI evaluates board states using:
  - Distance between pieces and the goal zone.
  - Number of pieces in the opponent’s starting zone.
  - Penalties for pieces still in the starting zone.

Challenges & Solutions

1. **Move Validation**
   - Players can move to adjacent spaces or jump over other pieces.
   - **Solution:** Implemented recursive functions to track all valid moves.
2. **AI Decision-Making**
   - The AI needs to evaluate board states accurately.
   - **Solution:** Used a heuristic evaluation function to assess board control, opponent blocking, and piece positioning.

Technical Details

Tools & Technologies Used:

- **Programming Language:** Python
- **Libraries:** Pygame
- **AI Algorithm:** Minimax with Alpha-Beta Pruning

Requirements:

- Python 3.x

- Pygame Library (pip install pygame)

Installation Steps:

1. Clone the repository to your local machine.
2. Install Pygame using the command:

```
pip install pygame
```

3. Run the game:

```
python main.py
```

Expected Outcomes

- A fully functional AI-powered Checkers Gameplay.
- A well-implemented Minimax algorithm with Alpha-Beta Pruning.
- Enhanced problem-solving skills in AI-based game development.
- A refined graphical interface that provides a seamless gaming experience.

Conclusion

This project will demonstrate the practical application of Artificial Intelligence in strategic decision-making. By implementing Minimax and Alpha-Beta Pruning, we will build an AI opponent capable of making optimal moves in Checkers Gameplay. Additionally, this project will provide valuable hands-on experience in game development, AI algorithms, and Python programming.

References

- Pygame Documentation
- Minimax Algorithm
- Alpha-Beta Pruning