AI Assignment 2

Assignment: Implementing the Minimax Algorithm with Alpha-Beta Pruning

Deadline: 06/11/2024

Objective:
The goal of this assignment is to implement the minimax algorithm along with alpha-beta pruning for a simple two-player game (like Tic-Tac-Toe) and analyze the performance of the algorithm.

Part 1: Minimax Algorithm Implementation

1. **Game Selection:**
   Implement a two-player deterministic game. You can choose a simple game like **Tic-Tac-Toe** or **Connect 4** (a simplified version can be used if needed).
2. **Game Logic:**
   - Define the game states, including player turns and terminal conditions (win, lose, or draw).
   - Create a game tree representation showing possible moves for both players (maximizer and minimizer).
3. **Minimax Algorithm:**
   Write a function that applies the minimax algorithm to calculate the best possible move for the player. The function should:
   - Evaluate game states using a utility function (e.g., +1 for a win, -1 for a loss, 0 for a draw).
   - Recursively simulate moves for both players to explore future game states.

Part 2: Alpha-Beta Pruning

4. **Optimizing with Alpha-Beta Pruning:**
   Extend the minimax algorithm to include alpha-beta pruning. The modified algorithm should:
   - Prune branches of the game tree that do not affect the final decision, improving the efficiency of the algorithm.

5. **Performance Comparison:**
    - Compare the time complexity and performance (in terms of nodes explored) between the basic minimax algorithm and the alpha-beta pruning implementation.

Part 3: Coding and Testing

6. **Code Implementation:**
    - Implement both the minimax and alpha-beta pruning algorithms in **Python** or another language of your choice.
    - Provide functions to simulate the game and play against an AI using both algorithms.
7. **Test Cases:**
    - Create several test cases to validate your implementation.
    - Compare the performance of the algorithms by running the same test cases and measuring the time taken to compute moves.

Part 4: Written Report

8. **Write-Up:**
   In a report, summarize:
    - How the minimax algorithm works.
    - The benefits of alpha-beta pruning.
    - The performance differences observed during testing.

Bonus: Advanced Exploration (Optional)

9. **Heuristic Evaluation:**
   Modify the evaluation function to use heuristic evaluation for non-terminal states (for example, for intermediate positions in Tic-Tac-Toe, favor configurations that could lead to a win).

Submission:
Submit your source code, test cases, and a report discussing your findings.

Evaluation Criteria:

- Correct implementation of the minimax and alpha-beta pruning algorithms.

- Clarity and readability of the code.
- Performance optimization using pruning.
- Quality of the test cases and written report.

Note

Submissions will be **rigorously checked** for plagiarism using automated tools. If any part of your code is found to be copied from another student, an online source, or generated by AI tools (such as ChatGPT), you will **immediately receive a score of zero** for the entire assignment.