Assignment: Implementing a Chess-Playing Agent Using Minimax and Alpha-Beta Pruning

Objective:

Develop a chess-playing agent in Python using a chess library. The agent should use the Minimax algorithm enhanced with Alpha-Beta pruning to evaluate game states and make optimal moves.

Requirements:

- Use a Python chess library (python-chess or any other similar library).
- Implement the Minimax algorithm.
- Incorporate Alpha-Beta pruning into the Minimax algorithm to optimize performance.
- The agent should be capable of playing a complete game of chess against a human or another agent.
- The agent should handle different game states and make decisions based on a specified depth of search.
- Ensure your code is modular, with clear functions and documentation.

Steps and Guidelines:

1. Setup and Library Installation
    - Install the chosen chess library. For python-chess, use the following command:
      pip install python-chess
2. Basic Chess Game Setup]
    - Create a basic setup to initialize a chessboard using the library.
    - Implement functions to display the board and make moves.
3. Implementing Minimax Algorithm
    - Develop the Minimax algorithm for decision making.
    - Define a utility function to evaluate board states.
    - Implement recursive functions to simulate moves and evaluate outcomes.
4. Incorporating Alpha-Beta Pruning
    - Enhance your Minimax algorithm by adding Alpha-Beta pruning to reduce the number of nodes evaluated.
    - Ensure that the pruning correctly eliminates non-promising branches of the search tree.
5. Depth Control
    - Implement a depth control mechanism to limit how far ahead the agent looks when evaluating moves. This will make the agent more efficient and responsive.
6. Testing and Debugging
    - Test your agent against a human player and another basic agent.
    - Debug any issues related to move generation, evaluation, or pruning.
7. Documentation and Code Quality
    - Comment your code to explain the purpose and functionality of each function.

Ensure code is modular, separating the game engine, agent logic, and utility functions.

Deliverables:

- Source Code: Complete Python code for the chess-playing agent.
- Readme File: Instructions on how to run the agent, including any dependencies.
- Documentation: A brief explanation of how your Minimax and Alpha-Beta pruning work.

Bonus (Optional):

- Advanced Evaluation: Implement more sophisticated evaluation functions considering positional advantages.
- User Interface: Develop a simple graphical user interface (GUI) to play against the agent.
- Different Difficulty Levels: Allow the user to set different depths for the agent, thus adjusting the difficulty level.