**Testing Report for Ultimate Tic-Tac-Toe with Alpha-Beta Pruning**

**Objective:**  
The goal of testing the Ultimate Tic-Tac-Toe game is to ensure that it behaves as expected under a variety of conditions, including different game outcomes (win, tie), valid/invalid moves, and AI performance using the minimax algorithm with alpha-beta pruning.

**Tests:**

1. **Basic Functionality Testing**
   * **Description:** Playing several full games where both players (human and AI) make legal moves.
   * **Expected Outcome:** The game should progress without errors, alternating turns between the human and AI players, with moves being made within the correct small boards and the correct choice of big boards.
   * **Result:** The game progressed smoothly. The board updated correctly after each move.
2. **Win Condition Testing (Small Board)**
   * **Description:** Testing all possible win scenarios on a single small board (rows, columns, diagonals).
   * **Expected Outcome:** Once a small board has a winning line (3 in a row), the entire board should be marked as won by that player, and further moves in that board should not be allowed.
   * **Result:** The small board win condition was detected correctly in all cases, and moves in a won board were prevented on both sides, AI and human.
3. **Win Condition Testing (Big Board)**
   * **Description:** Playing out scenarios where multiple small boards were completed, resulting in an overall win on the large board (rows, columns, diagonals).
   * **Expected Outcome:** Once three small boards are won by the same player in a line (row, column, diagonal), the game should declare a winner
   * **Result:** The big board win condition was correctly identified, and the game declared the correct winner.
4. **Draw (Tie) Testing (Small Board)**
   * **Description:** Filling a small board with no winner (all cells filled with no three in a row).
   * **Expected Outcome:** The small board should be declared a tie, and it should be marked as tied on the large board.
   * **Result:** The tie was detected correctly, and the board was updated to reflect the tie.
5. **Draw (Tie) Testing (Big Board)**
   * **Description:** Playing a full game resulting in all small boards being either won or tied without a winner for the overall big board.
   * **Expected Outcome:** The game should declare the overall result as a tie once all boards are completed.
   * **Result:** The big board tie was correctly identified.
6. **Invalid Move Testing**
   * **Description:** Attempting to make invalid moves such as:
     + Moving into a cell already occupied.
     + Playing on a board that was already won or tied.
     + Inputting invalid cell numbers (outside 1-9).
     + Inputting a string that is not a number.
   * **Expected Outcome:** The game should reject invalid moves and prompt the user to try again.
   * **Result:** Invalid moves were correctly rejected, and the player was prompted to choose again.
7. **Alpha-Beta Pruning Performance Testing**
   * **Description:** Measuring the AI's response time by forcing the AI into different game states where it had to compute a move.
   * **Expected Outcome:** The AI should take no longer than 4 seconds to make a decision, and pruning should reduce the number of nodes evaluated.
   * **Result:** The AI consistently responded within 4 seconds and played optimal moves as expected.
8. **Switching X and O for Human and AI**
   * **Description:** Constantly switched between X and O signs for both the human and AI players during different multiple games to test if the game properly adjusts to the player symbols.
   * **Expected Outcome:** The game should adapt correctly to the new player signs with no issues in the game logic and update the cells accordingly along with the correct output at the end of the game.
   * **Result:** The game handled switching between X and O for both human and AI correctly, and the game continued as expected, with the appropriate player making the correct first move.

**Conclusion:**

The game was thoroughly tested and behaves as expected across a variety of scenarios. All win conditions, draws, and invalid move scenarios were handled correctly. The AI, using alpha-beta pruning, makes optimal decisions and performs within the expected time constraints. Switching between X and O for the human and AI players was also handled correctly. Based on the testing, I believe the program works correctly in all expected contexts.