The step-by-step procedure you've outlined demonstrates the key logic behind the Alpha-Beta pruning algorithm in adversarial search. Here are the important points and logics witnessed throughout the process:

1. **Initialization of Alpha and Beta**: The algorithm starts by initializing α as negative infinity and β as positive infinity, representing the best achievable scores for Max and Min, respectively.
2. **Performing the Search**: Starting from the root of the game tree, the algorithm considers each possible move for the current player (Max or Min) and evaluates the resulting game state using a heuristic or utility function.
3. **Pruning Unnecessary Branches**: During the search, the algorithm updates α and β values. In the Max player's turn, α is updated to the maximum of α and the evaluated score, while in the Min player's turn, β is updated to the minimum of β and the evaluated score. If β becomes less than or equal to α at any node, the algorithm prunes the rest of the branches below that node, as they won't affect the final decision.
4. **Backpropagation**: After evaluating or pruning all child nodes, the algorithm propagates the best possible score back up the tree. If it's a Max player's turn, it returns the maximum score from the child nodes, and if it's a Min player's turn, it returns the minimum score from the child nodes.
5. **Deciding the Move**: At the root node, the algorithm chooses the move that corresponds to the highest score for Max or the lowest score for Min, representing the best decision based on the current search depth and heuristic evaluation.

The key logic behind Alpha-Beta pruning lies in its ability to eliminate unnecessary branches from the search tree, thereby reducing the number of nodes that need to be evaluated. By maintaining bounds (α and β) on the possible values of each node, the algorithm can efficiently determine which branches can be pruned without affecting the final decision.

Through the step-by-step procedure, we observe how Alpha-Beta pruning optimizes the search process by avoiding the exploration of irrelevant branches, ultimately leading to a more efficient and effective decision-making process in adversarial search scenarios.