1.

Function to create and initialize the game board.

Function to displays the current state of the board.

Function to determines if there's a winner/looser or if the game is a draw.

Function to update the board with the player's move.

Function to implement the backtracking algorithm to evaluate the best move for the AI.

Function that uses the Minimax algorithm to find the optimal move for the AI.

2.

Initializes a 3x3 grid.

Requires the current board state.

Requires the current board state and returns the winner or a draw.

Requires the current board, player symbol, and the row/column for the move.

Requires the current board state and a boolean indicating if it's maximizing or minimizing.

Requires the current board state to return the best move.

3.

A logical data structure for the Tic Tac Toe board is a 2D array (or vector in C++). This allows easy access to each cell using row and column indices:

4.

Makes the code easier to manage and understand.

If you want to expand the game, having a class structure makes this easier.

5.

The AI will use the Minimax algorithm to decide its move. It cannot use a random number because that wouldn’t guarantee intelligent play. Instead, the algorithm evaluates all possible moves and selects the one that maximizes the AI's chances of winning while minimizing the player's chances.