Code Documentation

Tic Tac Toe Game designed in C Language

Sadat Hossain Adol

Contents

Data Types Used

INT: int is a fundamental data type in C that represents integer numbers. It can hold both positive and negative whole numbers, typically within a certain range determined by the system architecture (commonly from -2,147,483,648 to 2,147,483,647 for 32-bit systems).

CHAR: char is a fundamental data type in C that represents a single character, such as a letter, digit, or special symbol. It is stored as an integer value corresponding to its ASCII or Unicode representation.

POINTERS: A pointer is a variable that stores the memory address of another variable. It enables direct access and manipulation of memory locations, allowing efficient management of memory and facilitating dynamic memory allocation.

BOOLEAN: bool is a data type introduced in C99 (and provided by <stdbool.h>) that represents boolean values, which can be either true or false. Used for logical operations and conditions in programming, including control flow decisions (if, while, for statements)

Functions

- Void initializeBoard:

Purpose:

Initializes the Tic-Tac-Toe board with empty spaces (' ') at the start of a new game.

Parameters:

board[3][3]: A 2-dimensional array of characters representing the Tic-Tac-Toe board.

Explanation:

This function iterates through each element of the 3x3 board array using nested loops (for loops). It assigns each element the value ' ', indicating an empty space where players can place their marks ('X' or 'O').

- Void printBoard:

Purpose:

Prints the current state of the Tic-Tac-Toe board to the console for players to visualize.

Parameters:

board[3][3]: A 2-dimensional array of characters representing the Tic-Tac-Toe board.

Explanation:

This function formats and prints the 3x3 board array to the console. It uses nested loops to iterate through each row and column of the board, printing each cell's value ('X', 'O', or ' ') and formatting the board with lines and separators to enhance readability.

- Void printInstructions

Purpose:

Prints game instructions to the console, explaining how to play Tic-Tac-Toe.

Parameters:

None.

Explanation:

This function outputs text to the console, providing players with instructions on how to play the Tic-Tac-Toe game. It covers basic rules, gameplay mechanics, and how to input moves.

- Void printMoveHistory

Purpose:

Prints the history of moves made during the game, showing each move's row and column.

Parameters:

moveHistory[][2]: A 2-dimensional array of integers storing the history of moves, where each row contains the row and column of a move.

moveCount: Integer representing the number of moves made during the game.

Explanation:

This function iterates through the moveHistory array and prints each move's details (row and column) to the console. It uses moveCount to determine how many moves to print, ensuring only valid moves made during the game are displayed.

- Bool makeMove

Purpose:

Attempts to make a move (place a mark) on the Tic-Tac-Toe board at the specified row and column.

Parameters:

board[3][3]: A 2-dimensional array of characters representing the Tic-Tac-Toe board.

row: Integer representing the row (0 to 2) where the player wants to place their mark.

col: Integer representing the column (0 to 2) where the player wants to place their mark.

mark: Character representing the mark ('X' or 'O') to be placed on the board.

Return Value:

Returns true if the move was successful (the specified cell was empty and the mark was placed), otherwise returns false.

Explanation:

This function checks if the specified row and col are within valid bounds (0 to 2) and if the corresponding cell in the board is empty (' '). If both conditions are met, it assigns the mark to the board at the specified position and returns true. Otherwise, it returns false, indicating an invalid move.

- Bool checkWin

Purpose:

Checks if the specified player (mark) has won the game by having three marks in a row, column, or diagonal.

Parameters:

board[3][3]: A 2-dimensional array of characters representing the Tic-Tac-Toe board.

mark: Character representing the mark ('X' or 'O') to check for a win.

Return Value:

Returns true if the player with the specified mark has won the game, otherwise returns false.

Explanation:

This function checks all possible winning conditions on the board for the player represented by mark. It examines rows, columns, and both diagonals to see if all three cells contain the same mark. If any winning condition is found, it returns true; otherwise, it returns false.

- Bool checkDraw

Purpose:

Checks if the game has ended in a draw (all cells on the board are filled without any player winning).

Parameters:

board[3][3]: A 2-dimensional array of characters representing the Tic-Tac-Toe board.

Return Value:

Returns true if the game is a draw (all cells are filled without a winner), otherwise returns false.

Explanation:

This function iterates through the board array to check if there are any empty cells (' '). If all cells are filled and no winning condition (checkWin function) is met, it returns true, indicating a draw. Otherwise, it returns false.

- Int main

Purpose:

The main function that controls the flow of the Tic-Tac-Toe game, manages player turns, and interacts with the user through input and output.

Parameters:

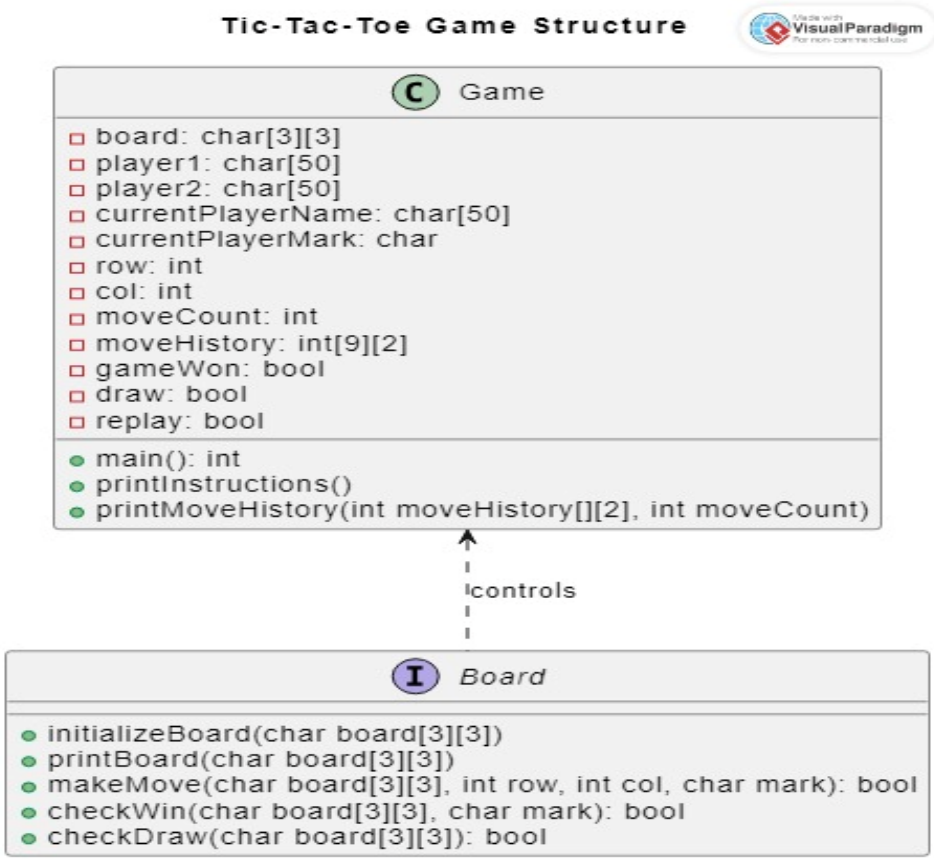
None.

Return Value:

Returns an integer (0) to indicate successful execution of the program.

Explanation:

The main function initializes game variables (board, player names, etc.), enters a loop to allow players to make moves (makeMove), checks for game-ending conditions (win or draw), switches players after each move, and prompts players if they want to play again (replay). It also calls other functions (printBoard, checkWin, checkDraw, printMoveHistory, etc.) to handle specific aspects of the game's functionality.



Github: https://github.com/Sadatadol/Tic-Tac-Toe-in-C-language/tree/main