CODE MANUAL

1. **Include Header Files**: The C++ code for Tic-Tac-Toe game starts by including several header files:
    - **<iostream>**: This header provides input and output functionalities for displaying messages on the console.
    - **<list>**: It's included to use the list data structure for storing coordinates of empty cells.
    - **<utility>**: It's included to use the **pair** data structure for representing coordinate pairs.
    - **<limits>**: Included to handle input validation and clear the input buffer.

2. **Enumeration PieceType**: An enumeration named **PieceType** is introduced to represent the different types of playing pieces. It includes three values: **EMPTY**, **X**, and **O**. These values represent an empty cell, 'X' piece, and 'O' piece, respectively.

3. **PlayingPiece Class**: The **PlayingPiece** class is defined to represent a playing piece on the game board. It has two constructors:
    - A default constructor, which creates an empty playing piece.
    - A constructor that takes a **PieceType** as an argument to set the piece type.

4. **Board Class**: The **Board** class is introduced to represent the game board. It includes the following functionalities:
    - A constructor that creates a game board of a specified size and initializes it with a default piece type (typically **EMPTY**).
    - A destructor responsible for freeing the memory used by the board.
    - The **placePiece** method, which is used to place a piece on the board at a specified row and column.
    - The **getEmptyCells** method, which returns a list of coordinates representing empty cells on the board.
    - The **printBoard** method, responsible for displaying the current state of the game board.

5. **Player Class**: The **Player** class is introduced to represent a player in the game. Each player has a name and a playing piece, either 'X' or 'O'.

6. **Game Class**: The **Game** class is defined to manage the game. It includes the following methods:
    - The **initializeGame** method initializes the game by creating players and the game board. It also sets up the initial player pieces and names.
    - The **startGame** method manages the game loop, allowing players to take turns, place their pieces on the board, and check for a winner after each move.
    - The **checkForWinner** method checks for a winner by examining the rows, columns, diagonals, and anti-diagonals on the board.

7. **Main Function**: The **main** function is the entry point of the program. It creates an instance of the **Game** class, initializes the game, and starts the game loop.

8. **Game Manual**: At the end of the **main** function, a game manual is provided. It explains how to play the game, the rules, and instructions for players.