```cpp
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

/*****************************************************************************
        Constants
*****************************************************************************/
const int ROWS = 3;
const int COLS = 3;
const int PLAYER_ONE = 1;
const int PLAYER_TWO = 2;
const int BLANK = 0;

/*****************************************************************************
        Function Prototypes
*****************************************************************************/
void displayMenu ();
void initializeBoard (int board[][COLS]);
void drawBoard (int board[][COLS]);
char getBoardChar (int board[][COLS], int row, int col);
void getUserSelection (int board[][COLS], int &row, int &col);
bool determineWinner (int board[][COLS], int &winner);
bool checkTiedGame (int board[][COLS]);
void displayEndingMessage (int winner);

/*****************************************************************************
 Function:     main

 Description:  Displays the menu, then depending on the user's selection,
               either encodes or decodes a message.

 Parameters:   None

 Returned:     Exit Status
*****************************************************************************/
int main ()
{
  int gameBoard[ROWS][COLS];
  int userRow, userCol;
  bool bGameOver = false;
  int winner = 0;
  int currentPlayer;

  initializeBoard (gameBoard);

  currentPlayer = PLAYER_ONE;

  while (false == bGameOver)
  {
    displayMenu ();
    drawBoard (gameBoard);
    cout << endl << "Player " << currentPlayer;
    getUserSelection (gameBoard, userRow, userCol);
    gameBoard[userRow][userCol] = currentPlayer;
    system ("pause");

    displayMenu ();
    drawBoard (gameBoard);
    bGameOver = determineWinner (gameBoard, winner);
    if (false == bGameOver)
    {
      bGameOver = checkTiedGame (gameBoard);
    }

    if (PLAYER_ONE == currentPlayer)
    {
      currentPlayer = PLAYER_TWO;
    }
    else
    {
      currentPlayer = PLAYER_ONE;
    }
  }

  displayEndingMessage (winner);

  return EXIT_SUCCESS;
}

/******************************************************************************
 Function:     displayTitle

 Description:  Displays the title and then calls the function drawBoard to
               display the board

 Parameters:   None

 Returned:     None
 ******************************************************************************/
void displayMenu ()
{
  system ("CLS");
  cout << "********************" << endl
       << "*   Tic-Tac-Toe    *" << endl
       << "********************" << endl << endl;
}

/******************************************************************************
 Function:     initializeBoard

 Description:  Initialize all of the elements in the board to blanks

 Parameters:   board - 2D array that represents the board

 Returned:     None
*****************************************************************************/
void initializeBoard (int board[][COLS])
{
}

/*****************************************************************************
 Function:     drawBoard

 Description:  Displays the board to the screen. This function must call
               the function displayBoardChar on each element in the array

 Parameters:   board - 2D array that represents the board

 Returned:     None
*****************************************************************************/
void drawBoard (int board[][COLS])
{
}

/*****************************************************************************
 Function:     getBoardChar

 Description:  Returns a single board character:
               If value is 0 - returns a blank
               If value is 1 - returns an X
               If value is 2 - returns an O

 Parameters:   board - 2D array that represents the board
               row   - the row index of the element
               col   - the col index of the element

 Returned:     the character reprsenting the player number
*****************************************************************************/
char getBoardChar (int board[][COLS], int row, int col)
{
  return ' ';
}

/*****************************************************************************
 Function:     getUserSelection

 Description:  Asks the user to enter in their selection and makes sure that
               the user selection is valid and that the element in the array
               is blank

 Parameters:   board - 2D array that represents the board
               row   - the row index of the element the user selected
               col   - the col index of the element the user selected

 Returned:     none
*****************************************************************************/
void getUserSelection (int board[][COLS], int &row, int &col)
{
}

/*****************************************************************************
 Function:     determineWinner

 Description:  Determines the winner of the game. A win can happen diagonally,
               vertically, or horizontally.

 Parameters:   board  - 2D array that represents the board
               winner - The winning player. Either PLAYER_ONE or PLAYER_TWO

 Returned:     true if a winner is found, false otherwise
*****************************************************************************/
bool determineWinner (int board[][COLS], int &winner)
{
  return true; // REPLACE THIS
}

/*****************************************************************************
 Function:     checkTiedGame

 Description:  Determines if all of the elements in the board have been filled.
               If they have been filled, then the game is a tie.

 Parameters:   board - 2D array that represents the board

 Returned:     true if there are no empty slots on the board, false otherwise
*****************************************************************************/
bool checkTiedGame (int board[][COLS])
{
  return true; // REPLACE THIS
}

/*****************************************************************************
 Function:     displayEndingMessage

 Description:  Display a message depending on the outcome of the game:
               "Congratulations player one! You have won."
               "Congratulations player two! You have won."
               "Cat's Game. The game was a tie."

 Parameters:   winner - The winning player

 Returned:     none
*****************************************************************************/
void displayEndingMessage (int winner)
{
}
```