```
using System;
using System.Collections.Generic;
using System.Text;
using static System.Console;

namespace TICTACTOEINFO350PROJUPD
{
   class Program
   {
      public static void BoardReset() //This is used to set a reset to the game(method).
      {
         char[] AssignArrBoard =
         {
            '1', '2', '3','4', '5', '6','7', '8', '9'
         };

         ArrayBoard = AssignArrBoard;
         Board();
         turns = 0;
      }

      public static char playerTurn = ' ';

      static int turns = 0; //This will count the amount of turns that are taken in each individual game.
      //Once the game reaches the max amount of 10 turns the game will automatically result in a draw.


      static char[] ArrayBoard =
      {
         '1', '2', '3','4', '5', '6','7', '8', '9' //player input value
      };

      public static void Board()
      {
         Console.Clear();
         Console.WriteLine("  -------------------------");
         Console.WriteLine("  |      |      |      |");
         Console.WriteLine("  |   {0}  |   {1}  |   {2}  |", ArrayBoard[0], ArrayBoard[1], ArrayBoard[2]);
         Console.WriteLine("  |      |      |      |");
         Console.WriteLine("  -------------------------");
         Console.WriteLine("  |      |      |      |");
```

```
        Console.WriteLine("  |   {0}   |   {1}   |   {2}   |", ArrayBoard[3], ArrayBoard[4], ArrayBoard[5]);
        Console.WriteLine("  |       |       |       |");
        Console.WriteLine("  -------------------------");
        Console.WriteLine("  |       |       |       |");
        Console.WriteLine("  |   {0}   |   {1}   |   {2}   |", ArrayBoard[6], ArrayBoard[7], ArrayBoard[8]);
        Console.WriteLine("  |       |       |       |");
        Console.WriteLine("  -------------------------");
    } //This is used to show a representation of the board and the numbers that correspond with each individual box by using writeline.

    public static void Introduction()
    {
        Console.ForegroundColor = ConsoleColor.Blue;
        Console.WriteLine("\nLets play tic-tac-toe! Press any key to start");
        Console.ResetColor();
        Console.ReadKey();
        Console.Clear();
        Console.ForegroundColor = ConsoleColor.Blue;
        Console.WriteLine("How to play");
        Console.ResetColor();
        Console.WriteLine("Each player selects a number in the square of their choice" +
                "\nPlayer 1 = X.  Player 2 = O");
        Console.WriteLine("\nThe first player who gets either three X's or O's in a row wins" +
                "\nReady? \npress any key to start...");
        Console.ReadKey();
    } //This method introduces the game and how it's played. It also is made to provide a stucture to the game to make it easier to play.

    public static void XorO(int player, int input)
    {

        if (player == 1) playerTurn = 'X';
        else if (player == 2) playerTurn = 'O';

        switch (input)
        {
            case 1: ArrayBoard[0] = playerTurn; break;
            case 2: ArrayBoard[1] = playerTurn; break;
            case 3: ArrayBoard[2] = playerTurn; break;
            case 4: ArrayBoard[3] = playerTurn; break;
            case 5: ArrayBoard[4] = playerTurn; break;
```

```
            case 6: ArrayBoard[5] = playerTurn; break;
            case 7: ArrayBoard[6] = playerTurn; break;
            case 8: ArrayBoard[7] = playerTurn; break;
            case 9: ArrayBoard[8] = playerTurn; break;
        }

    } //Controls if the player is X or O.0 counts as 9 spots, put notes where u think it is warranted.

    static void Main(string[] args)
    {
        int player = 2;
        int input = 0;
        bool inputCorrect = true;

        Introduction();


        do //This represents that each player with alternate in picking the number that corresponds to the box they would like to input their character into.
        {
            if (player == 2)
            {
                player = 1;
                XorO(player, input);
            }
            else if (player == 1)
            {
                player = 2;
                XorO(player, input);
            }

            Board();
            turns++;



            VerticalWin();
            DiagonalWin();
            HorizontalWin();

            if (turns == 10)
            {
```

```
            Draw();
        }

        do
        {
            Console.WriteLine("\nIt is now your turn Player {0}!", player);
            try
            {
                input = Convert.ToInt32(Console.ReadLine());
            }
            catch
            {
                Console.WriteLine("Please enter a number that appears on the board!");
            }

            if ((input == 1) && (ArrayBoard[0] == '1'))
                inputCorrect = true;
            else if ((input == 2) && (ArrayBoard[1] == '2'))
                inputCorrect = true;
            else if ((input == 3) && (ArrayBoard[2] == '3'))
                inputCorrect = true;
            else if ((input == 4) && (ArrayBoard[3] == '4'))
                inputCorrect = true;
            else if ((input == 5) && (ArrayBoard[4] == '5'))
                inputCorrect = true;
            else if ((input == 6) && (ArrayBoard[5] == '6'))
                inputCorrect = true;
            else if ((input == 7) && (ArrayBoard[6] == '7'))
                inputCorrect = true;
            else if ((input == 8) && (ArrayBoard[7] == '8'))
                inputCorrect = true;
            else if ((input == 9) && (ArrayBoard[8] == '9'))
                inputCorrect = true;
            else
            {
                Console.WriteLine("Sorry that number has already been entered. \nPlease enter another number
that appears on the board...");
                inputCorrect = false;
            }
```

```
        } while (!inputCorrect);
      } while (true);

    } //Do while loop used to keep turns organized and ensures correct use of values.

    public static void VerticalWin()
    {
      char[] playerTurns = { 'X', 'O' };

      foreach (char playerSymbol in playerTurns)
      {
        if (((ArrayBoard[0] == playerSymbol) && (ArrayBoard[3] == playerSymbol) && (ArrayBoard[6] ==
playerSymbol))
          || ((ArrayBoard[1] == playerSymbol) && (ArrayBoard[4] == playerSymbol) && (ArrayBoard[7] ==
playerSymbol))
          || ((ArrayBoard[2] == playerSymbol) && (ArrayBoard[5] == playerSymbol) && (ArrayBoard[8] ==
playerSymbol)))
        {
          Console.Clear();
          if (playerSymbol == 'X')
          {
            Console.WriteLine("Game Over! Congratulations Player 1, You Won!.\nBy vertical win!\nYou are
the Tic Tac Toe GOAT!!!\n");
          }
          else
          {
            Console.WriteLine("Game Over! Congratulations Player 2, You Won!\nBy vertical win!\nYou are
the Tic Tac Toe GOAT!!!\n");
          }

          ending();
          Console.WriteLine("Want to play again? Press any key to restart game.");
          Console.ReadKey();
          BoardReset();

          break;
        }
      }
    } //Used to determine whether the game was won by a vertical win.

    public static void HorizontalWin()
    {
```

```
        char[] playerTurns = { 'X', 'O' };

        foreach (char playerSymbol in playerTurns)
        {
          if (((ArrayBoard[0] == playerSymbol) && (ArrayBoard[1] == playerSymbol) && (ArrayBoard[2] ==
playerSymbol))
              || ((ArrayBoard[3] == playerSymbol) && (ArrayBoard[4] == playerSymbol) && (ArrayBoard[5] ==
playerSymbol))
              || ((ArrayBoard[6] == playerSymbol) && (ArrayBoard[7] == playerSymbol) && (ArrayBoard[8] ==
playerSymbol)))
          {
            Console.Clear();
            if (playerSymbol == 'X')
            {
              Console.WriteLine("Game Over! Congratulations Player 1. You Won!\nYou achieved a horizontal
W! " +
                        "\nYou are now a Tic Tac Toe legend!!!\n" +
                        "\nTurns taken{0}", turns);
            }
            else if (playerSymbol == 'O')
            {
              Console.WriteLine("Game Over! Congratulations Player 2 You Won.\nYou achieved a horizontal W!
" +
                        "\nYou are the Tic Tac Toe winner!!!\n");
            }

            ending();
            Console.WriteLine("Want to play again? Press any key to restart game.");
            Console.ReadKey();
            BoardReset();

            break;
          }
        }
      } ///Used to determine whether the game was won by a horizontal win.

      public static void DiagonalWin()
      {
        char[] playerTurns = { 'X', 'O' };

        foreach (char playerSymbol in playerTurns)
```

```
        {
            if (((ArrayBoard[0] == playerSymbol) && (ArrayBoard[4] == playerSymbol) && (ArrayBoard[8] ==
playerSymbol))
                || ((ArrayBoard[6] == playerSymbol) && (ArrayBoard[4] == playerSymbol) && (ArrayBoard[2] ==
playerSymbol)))
            {
                Console.Clear();
                if (playerSymbol == 'X')
                {
                    Console.WriteLine("Game Over! Congratulations Player 1, You Won!\nYou won diagonally! " +
                            "\nYou are a Tic Tac Toe GOAT!!!\n");
                }
                else
                {
                    Console.WriteLine("Game Over! Congratulations Player 2, You Won!\nYou won diagonally! " +
                            "\nYou are a Tic Tac Toe GOAT!!!\n");
                }

                ending();
                Console.WriteLine("Want to play again? Press any key to restart game.");
                Console.ReadKey();
                BoardReset();

                break;
            }
        }
    } //Used to determine whether the game was won by a diagonal win

    public static void Draw()
    {

        {
            Console.WriteLine("Tie Game! No winners this round..." +
                    "\nWant to play again? Press any key to restart game.");

            Console.ReadKey();
            BoardReset();

        }
    } //Used to determine whether the game has resulted in a draw.
    public static void ending()
```

```
        {
            Console.ForegroundColor = ConsoleColor.Blue;
            Console.WriteLine("\nYOU WIN!!!! LETS GOO!!! Press any key to continue. ");
            Console.ReadKey();
            Console.Clear();
            Console.ResetColor();
        } //End of code/Game till loop begins again
    }
}
```