

[407a] Tic-Tac-Toe Solution

```
using System;
```

```
namespace LWTech.ChipAnderson.TicTacToe
```

```
{  
    class Program  
    {  
        static Random rng = new Random();  
  
        static void Main(string[] args)  
        {  
            Console.WriteLine("Tic-Tac-Toe \t\t Chip Anderson");  
            Console.WriteLine("=====");  
            Console.WriteLine();  
  
            int[] grid = new int[9];    // using a linear array instead of 3x3 matrix  
  
            int winner = 0;  
            int numTurns = 0;  
  
            // Initialize the grid  
            InitializeGrid(grid);  
  
            int currentPlayer = 1;  
  
            // As long as we don't have a winner...  
            bool gameOver = false;  
            while (!gameOver)  
            {  
                numTurns++;  
                Console.WriteLine("Player #" + currentPlayer + "'s turn!");  
  
                // Have the current player fill in an empty square (randomly).  
                int squareSelected = SelectEmptySquare(grid);  
                grid[squareSelected] = currentPlayer;  
  
                // Show everyone the grid  
                DisplayGrid(grid);  
            }  
        }  
    }  
}
```

```
// Did they get three in a row?
if (FoundThreeInARow(grid))
{
    gameOver = true;
    winner = currentPlayer;
}
else if (numTurns == 9)
{
    // All square are full and no winner. Tie game!
    gameOver = true;
    winner = 0;
}
else
{
    // No, change players
    currentPlayer = (currentPlayer == 2) ? 1 : 2;
}

}

// Tell everyone who won the game
if (winner == 0)
{
    Console.WriteLine("It's a tie!");
}
else
{
    Console.WriteLine($"Player #{winner} won!");
}
}

static void InitializeGrid(int[] grid)
{
    for (int i = 0; i < 9; i++)
    {
        grid[i] = 0;
    }
}

static void DisplayGrid(int[] grid)
{
    for (int j = 0; j < 3; j++)
```

```
{  
    for (int i = 0; i < 3; i++)  
    {  
        char c = " XO"[grid[i * 3 + j]];  
        Console.Write(" " + c);  
        if (i < 2)  
            Console.Write(" |");  
    }  
    Console.WriteLine();  
    if (j < 2)  
        Console.WriteLine("----+----+----");  
}  
Console.WriteLine();  
}
```

```
static int SelectEmptySquare(int[] grid)
```

```
{  
    int i;  
  
    do  
    {  
        i = rng.Next(9);  
    } while (grid[i] != 0);  
  
    return i;  
}
```

```
static bool FoundThreeInARow(int[] grid)
```

```
{  
    // Horizontals  
    if (grid[0] != 0 && grid[0] == grid[1] && grid[1] == grid[2])  
        return true;  
    if (grid[3] != 0 && grid[3] == grid[4] && grid[4] == grid[5])  
        return true;  
    if (grid[6] != 0 && grid[6] == grid[7] && grid[7] == grid[8])  
        return true;  
  
    // Verticals  
    if (grid[0] != 0 && grid[0] == grid[3] && grid[3] == grid[6])  
        return true;  
    if (grid[1] != 0 && grid[1] == grid[4] && grid[4] == grid[7])  
        return true;
```

```
if (grid[2] != 0 && grid[2] == grid[5] && grid[5] == grid[8])
    return true;

// Diagonals
if (grid[0] != 0 && grid[0] == grid[4] && grid[4] == grid[8])
    return true;
if (grid[2] != 0 && grid[2] == grid[4] && grid[4] == grid[6])
    return true;

return false;
}
}
}
```