

Tic-Tac-Toe

By

**Sanket K. Gupta (CE-045) (22CEUOG043)**  
B.Tech. CE Semester I

Subject: PPS-I

Under the guidance of  
**Prof. Hariom Pandya**  
**Prof. Chintan Chatterjee**



**Dharmsinh Desai University, Nadiad**  
**Faculty of Technology**  
**Department of Computer Engineering**

## **CERTIFICATE**

This is to certify that **Programming For Problem Solving- I** Project entitled “**Tic-Tac-Toe**” is the bonafide report of work carried out by,

**Sanket K. Gupta (22CEUOG043)**

Of Department of Computer Engineering, Semester-I, academic year 2022-2023, under the supervision and guidance of:

**Prof. Hariom Pandya**

**Prof. Chintan Chatterjee**

# Contents

<b>1</b>	<b>Abstract</b>	<b>3</b>
1.1	Purpose . . . . .	3
<b>2</b>	<b>Introduction</b>	<b>4</b>
2.1	Introduction to Tic-Tac-Toe . . . . .	4
2.2	Technologies Used . . . . .	4
<b>3</b>	<b>The UX</b>	<b>5</b>
<b>4</b>	<b>Requirements taken care of</b>	<b>6</b>
<b>5</b>	<b>Data Structure and Algorithm</b>	<b>7</b>
5.1	Data Structure . . . . .	7
5.2	Algorithm . . . . .	7
<b>6</b>	<b>Bibliography</b>	<b>8</b>

# 1 Abstract

Tic-Tac-Toe is the digitalization of the famous game of children by the same name.

## 1.1 Purpose

The main purpose behind the creation of this C Language Project was to strengthen my understanding about the concepts of C Language, including, but not limited to:

- Conditional Statements
- Looping

This Project has helped me also to understand how programming applies in real life scenarios, and also that before solving a problem, one must have a clear and deep understanding of the algorithm that he/she would be using.

## 2 Introduction

### 2.1 Introduction to Tic-Tac-Toe

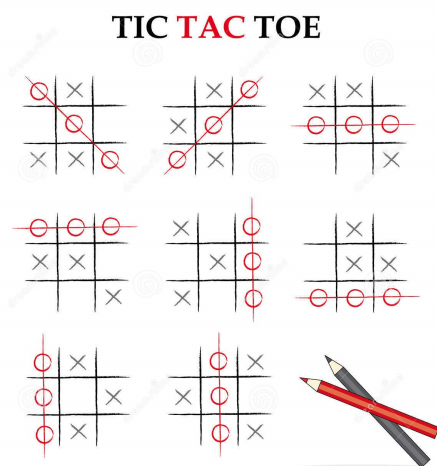
The Game consists of a 3 x 3 grid with two players. One of the players', say Player 1, can start the game by marking a **x** (cross) in one of the nine boxes. Player 2, then, proceeds by marking a **o** (circle). Then, the turns pass, until any one of the following situation occurs:

**Player 1 wins:** If *three crosses come in a line*, either in a row, or in a column, or in any of the two diagonals of the layout, then Player 1 wins.

**Player 2 wins:** If *three circles come in a line*, either in a row, or in a column, or in any of the two diagonals of the layout, then Player 2 wins.

**A tie occurs:** If *all the nine boxes are filled without anyone winning*, then it is declared a tie.

The Graphical Representation of winning conditions for Player 2 is given below, similar conditions exist for Player 1 also.



### 2.2 Technologies Used

- C Language (For Programming)
- Online GDB (For Debugging)
- VS Code (As an IDE)

### 3 The UX

After the execution of the program, it displays the following constraints imposed on the two players:

Player No.	Symbol
1	o (circle)
2	x (cross)

These constraints have been imposed so as to simplify the implementation, and in no way, do they interfere with the playing experience.

Next, the program prompts for the names of the two players.

Game starts with Player 2 specifying a position(**integer between 0 to 9**) for **x** (cross). Then, the current situation of game is displayed.

Next comes the turn of Player 1, and they too specify the position for **o** (circle).

The game so proceeds until a decision is reached, i.e., either a player wins, or a tie occurs.

A sample game is displayed as follows:

```
PS D:\CodingC> gcc .\tictactoe.c
PS D:\CodingC> .\a.exe

TIC-TAC-TOE!

Rules:
1. Player 1 can put only 'o'.
2. Player 2 can put only 'x'.
Both players need to specify positions where to put their respective character.

Enter Player 1's name:Sanket
Enter Player 2's name:Harshpal

Harshpal, enter position number where you want to put 'x':1
The current situation of game is this:
x

Sanket, enter position number where you want to put 'o':2
The current situation of game is this:
x o

Harshpal, enter position number where you want to put 'x':4
The current situation of game is this:
x o
x

Sanket, enter position number where you want to put 'o':3
The current situation of game is this:
x o o
x

Harshpal, enter position number where you want to put 'x':7
The current situation of game is this:
x o o
x
x
x
Harshpal wins!

PS D:\CodingC> []
```

## 4 Requirements taken care of

**Invalid position is entered:** Whenever a player tries to enter a position that is *already filled, or is not present in the layout grid*, the program prompts for **try again**.

**Exit message:** If *a player wins or if there is a tie*, message is displayed accordingly.

## 5 Data Structure and Algorithm

### 5.1 Data Structure

A **2D Character Array** of size **3 x 3** is used as a data structure for **Layout Grid**.

The major challenge faced here was **to convert the position specified in integer by the players to the indices of row and column of Layout Grid**.

If **p** indicates integer variable for **position**, **i** indicates integer variable for **row index**, **j** indicates integer variable for **column index**, **GRID** is the **macro for 3**, and **arithmetic rules of C Language** are followed, then mathematical formulas are applicable:

$$i = \left( \frac{p - 1}{GRID} \right)$$
$$j = (p - 1) - \left( \left( \frac{p - 1}{GRID} \right) \times GRID \right)$$

### 5.2 Algorithm

The major part of the implementation consists of a **while loop** which prompts the appropriate player for their position, based on the value of a variable **turn**.

Value of turn variable	Player Prompted
Even	Player 1
Odd	Player 2

After each successful entry of position, **condition** is checked for **row match**, **column match**, **principal diagonal match**, and **secondary match**. If there is any match, the appropriate player is **declared winner**, and the program terminates with the use of **exit(0)**.

If there is **no match even after all nine turns are exhausted**, it is declared a **tie**.



## 6 Bibliography

### References taken from:

1. **Programming in ANSI C**, book by E. Balagurusamy.
2. **W3Schools**, website.