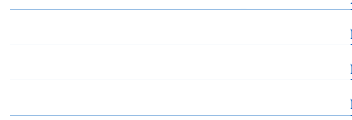


# Tic- Tac-Toe



## I. INTRODUCTION

Our project is to recreate the most famous Tic-Tac-Toe game using the python programming language. We made use of pygame module in this project to create a user-friendly interface.

Basically, our Tic-Tac-Toe game is a traditional player Vs player (PVP) game that displays the required output accordingly as the game ends. Tic-Tac-Toe is a classic 3x3 board game and players are assigned one of the symbols from 'O' and 'X'. The player who succeeds in placing 3 of their assigned symbols in a horizontal, vertical, or diagonal row first is the winner and the game will be drawn if no player could win before they fill the entire board

## II. OBJECTIVE

Our objective is to recreate our childhood memories on a computer using python and it could be even used by children to create their own memories right now. Actually, these games are just created to cure our boredom.

The Tic-Tac-Toe game actually contributes to children's developmental growth in various ways like understanding predictability, problem-solving, hand-eye coordination, turn-taking, and making strategies.

## III. THE GAME

In our game, we've designated the symbol 'O' to the player-1 and the symbol 'X' to the player 2. By default, we give the first chance to the player-1 for every new match.

### A. Procedure:

- Player-1 must select a box out of 9 boxes and his respective 'O' symbol inside the selected box.
- Now, player-2 gets a chance and the symbol 'X' is drawn in the box he/she selected.
- Both the players must play in a way to match their symbols along any row, column, or diagonal fashion.
- The player who manages to match his symbols first is considered to be the winner of that game.
- A screen will be shown up accordingly depending upon who won the game.
- If no player won the game i.e., all the squares are filled then it will be considered as the "Draw".
- Accordingly, a suitable screen will be shown up. So, the game gets over when a player won or game is drawn.
- The player will be asked to press any key on the keyboard to restart the game.

### B. Pseudo code:

- Create a user-friendly interface according to the requirements. (we used pygame module to get this job done).
- Create an empty game board and store the user's move in some form of input.
- Make sure the move is valid and check whether the player won the game or not.
- Change the player's turn and allow the next player to make his move.
- Draw the symbols('O' & 'X') in the boxes according to the player who tapped in them.
- Repeat this until the game gets over (a player won or game is drawn).
- Show up a screen showing how the game got over.

### C. Prerequisite: The user must have installed pygame module in order to play this game.

#### IV. CONCLUSIONS

We could conclude that the code that we've written is working fine by testing the code in several test cases. The test cases for our code are,

*A. A player winning the game:*

- 1) Player-1 winning the game:* We've tested this case by matching player-1's symbol in horizontal, vertical, and diagonal fashion.
- 2) Player-2 winning the game:* We've tested this case same as in the way that we've tested for player-1.

*B. Game getting drawn:* We've tested this case by filling the board in a way that no player wins the game.

#### REFERENCES

[1] [pygame.org](https://www.pygame.org)