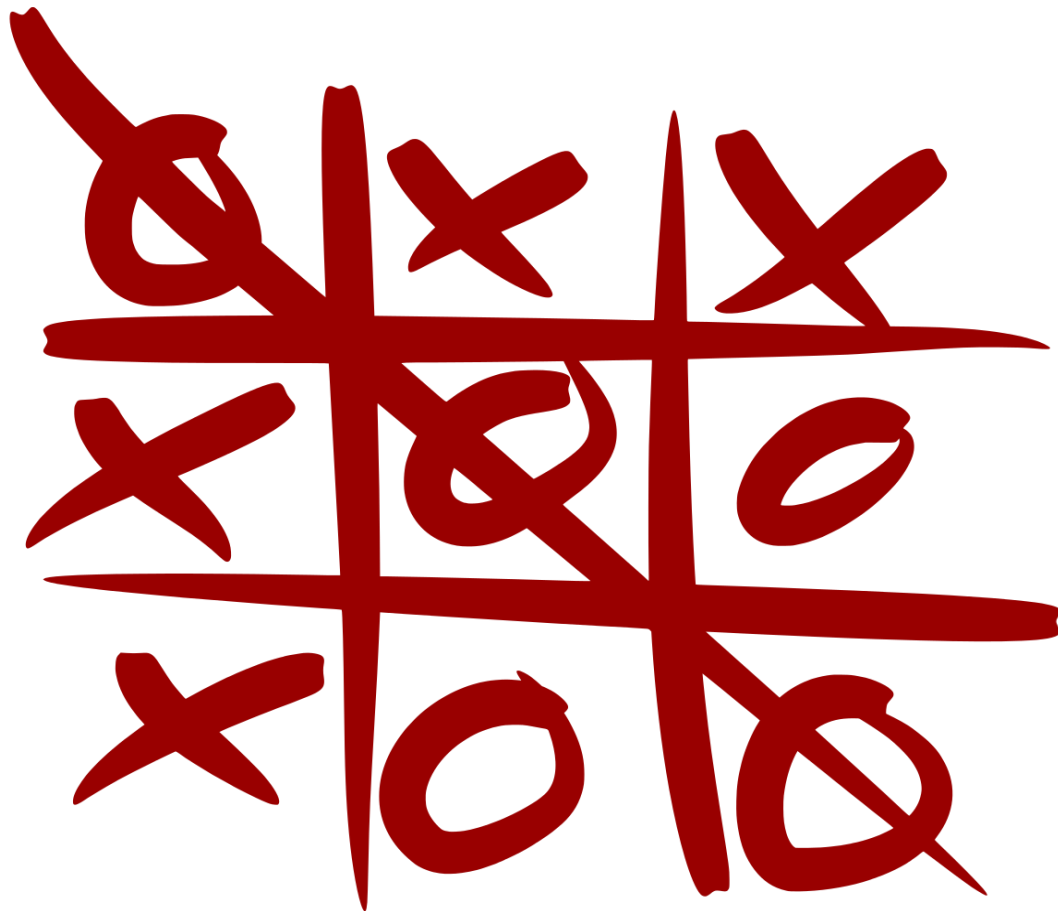


TIC TAC TOE

By: Archit Kashyap (IMT2016064),

Nomula Rohith Yogi (IMT2016072)



1. Introduction

This is a simple game which is played between two players on a 3×3 grid. In this game, each player chooses a symbol which he will use to fill the grids, generally the symbols are either “X” or “O”, each player fills the grids according to his wish when his turn comes.

Let us consider the grid as following:

1	2	3
4	5	6
7	8	9

Now the player whose symbols fills one of the following groups of places wins the game

- a) symbol at 1 = symbol at 2 = symbol at 3
- b) symbol at 4 = symbol at 5 = symbol at 6
- c) symbol at 7 = symbol at 8 = symbol at 9
- d) symbol at 1 = symbol at 4 = symbol at 7
- e) symbol at 2 = symbol at 5 = symbol at 8
- f) symbol at 3 = symbol at 6 = symbol at 9
- g) symbol at 1 = symbol at 5 = symbol at 9
- h) symbol at 3 = symbol at 5 = symbol at 7

2. Game View

We are making the game using the module tkinter, which helps to make the GUI for our game.

2.1 Main

In the start of the game the program asks the player whether he wants to play single player or multiplayer or he wants to exit the game, after the choosing the game version another window opens which contains a 3 * 3 grid where the player can play the game.

CHOOSE THE NUMBER OF PLAYER		
SINGLE	MULTI	EXIT

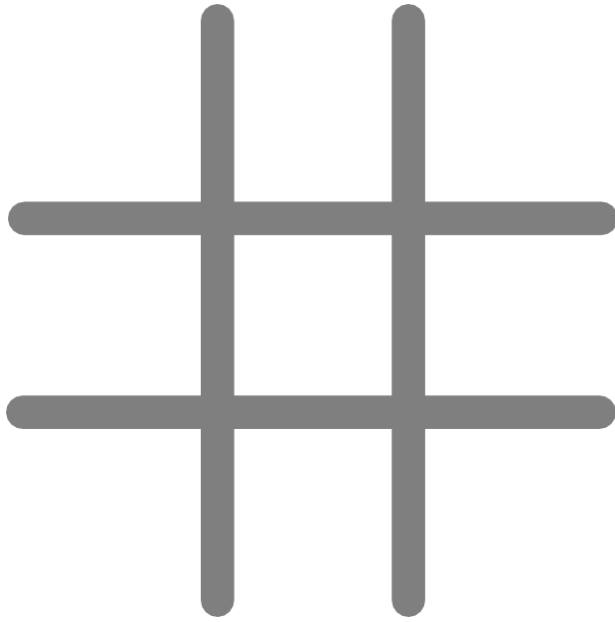
2.1.1. Main class

This class contains the main window of the game. We use widgets such as labels and buttons to make the GUI. The buttons call the different functions for single player and multiplayer respectively. The functions call the respective classes for single player and multiplayer.

2.2. Single Player:

2.2.1 Design:

In this project we have implemented a frame which consists of 11 buttons in total in that 9 buttons are represented in tic tac toe board format and one of the other two buttons represents the RESET and the other EXIT each button has its own importance because the user may want to quit the game at any point of time during the game so the exit button helps them to go to the main frame. The other one is reset button, this button helps the user play the game again and again by setting the board to the initial condition. The nine buttons that were mentioned earlier perform a specified task, when these buttons being clicked by the user "X" appears on the button (as the tic tac toe game itself is played by X and O's). "X" letter on the button indicates that the user has completed his/her turn, then computer places the next move by turning an empty button to "O".



The above is a tic tac toe board and this board is represented with the help of the nine buttons which when clicked represents “X” or “O” according to the turn.

2.2.2 Specification

Here are few specifications of the single player code

2.2.2.1 Class for single player:

The class initiates the list named board and each element in the list is initialized and various functions are defined

2.2.2.1.1 The init function:

This function initializes the eleven buttons that are present on the frame each button is placed in the specified position by using a widget named grid and the nine buttons used for

playing are binded such that when the button is clicked the text “X” or “O” appears on the button. And both the back and the reset buttons functions are called if clicked on them, this is implemented using widget named command.

2.2.2.1.2 View function:

As the button is clicked the text appears this is done by using a widget named config and the font and size are managed by this widget. Every time the button is clicked the program enters into this function and the desired output gets printed.

2.2.2.1.3 task function:

This function contains an event widget and it stores the position at which the button is clicked and calls the player function.

2.2.2.1.4 player function:

This function checks if the position in which the button clicked is empty or it contains “X” or “O” and if contains “X” or “O” it uses a widget named message box to tell user the position he /she had clicked is invalid. And if the move is valid then the position is marked “X” then it enters the view function and then comes the computers turn in which the

player function calls another function called as computers turn and after the computers turn it checks the winning conditions by calling a check win function this function is called after each turn as we never know in which turn who may win it may be the computer or the user

2.2.2.1.5 check win function:

The check win function checks all the winning, loosing and draw conditions as mentioned earlier by if else conditions and a message box is returned after winning or loosing stating the winner.

2.2.2.1.6 computers turn:

This function places “O” randomly where ever it finds an empty button (without any text) enabling the player to win for some conditions. This we have done by using lists that contain the positions where the “O” would be generated. And it is called accordingly.

2.2.2.1.7 end function:

This function is called after clicking the exit button on the frame here we have used the widget destroy to end the program.

2.2.2.1.8 reset function:

This function resets all the text on the buttons to empty

(" ") to enable the user to play the game once again.

2.3 Multi Player

2.3.1 Design

The window contains a grid where the person can play. It contains a status bar below the grid which shows the result of the game. Below the status bar there are two buttons one to "start the game" and another "go back" to the main window. The first person is assigned the default symbol "O" and the next person the symbol "X", when a person clicks a square his respective symbol shows and if he wins the game the status bar displays and the game stops, further by clicking in the start button he can restart the game or he can go back by choosing the back button.



Status

Start

Back

2.3.2 Specification

2.3.2.1 Class for Multiplayer

This class contains various functions which creates the multiplayer game.

2.3.2.1.1 The init function

This function makes the GUI of the game by using widgets such as frame, label, canvas and buttons. Here we make a label which shows the status of the game. There are two buttons one which calls the function to start the game and the other to go back to main window.

2.3.2.1.2 The game function

Here the function calls another function which creates the board or the grid where the game is played, this is achieved by using the canvas widget. After creating the board the fuctions calls another function which checks where the player clicks on the board and keeps account for the player who is clicking and then create the respective symbol in the grid according to the player. After this a function checks whether a player has win or not, if a

player wins then the game stops and the status bar shows the result, else the game continues till a draw is made.

2.3.2.1.3 The display function

This function checks in which region a click has been made by using the co – ordinate system and check which player has made the click and then displays the respective symbol and assigns a numeral value for that box in the grid for further use in the check.

2.3.2.1.4 The check function

This function checks the which player has won the game or there is draw by using the numerals assigned to different boxes in the grid.