



# **TIC-TAC-TOE GAME**

**(X-O GAME)**

## PROJECT TEAM :

Name	Sitting number	Section
■ الاء عماد عبد الحميد وهبة	4084	4
■ ايمان محمد عبد المحسن احمد	4109	4
■ جهاد علاء الدين ابراهيم محمد	4133	5
■ رؤى ايمن رمزي محمد جاويش	4166	6
■ رؤى مرسى عطية عبد الحميد	4167	6
■ مريم رافت رمزي عبدالملك	4456	14
■ ناردين صبحي شاكر عبدالملاك	4499	15
■ نرمين غيث علي حسين	4500	15
■ نهي احمد محمد محمد	4502	15

## Main Idea Of The Project :

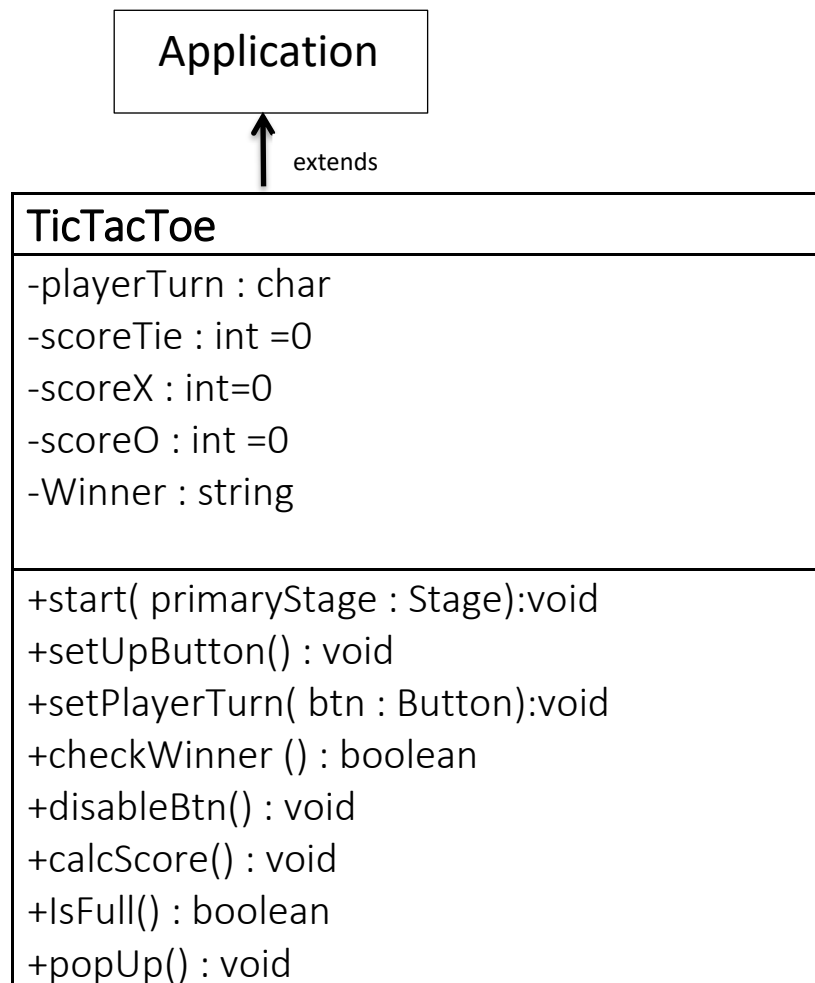
Tic-tac-toe is a game in which there are only two players playing together ,when you run the program ,it appears in the first stage a question that makes you choose to start with **X** or **O**.

The game is set to be on a screen divided into 9 buttons three above each others , one of the players start the game say " player **X** " by clicking on one a button of the 9 ones ( he can't change his selected one after choosing it unless he click on Reset button that reset all scores to zero and clear all characters ) , then it will be player's **O** turn then it will back player's **X** turn and so on , till one of them win by achieving three same characters in the same row or in the same column or in a diagonal shape .

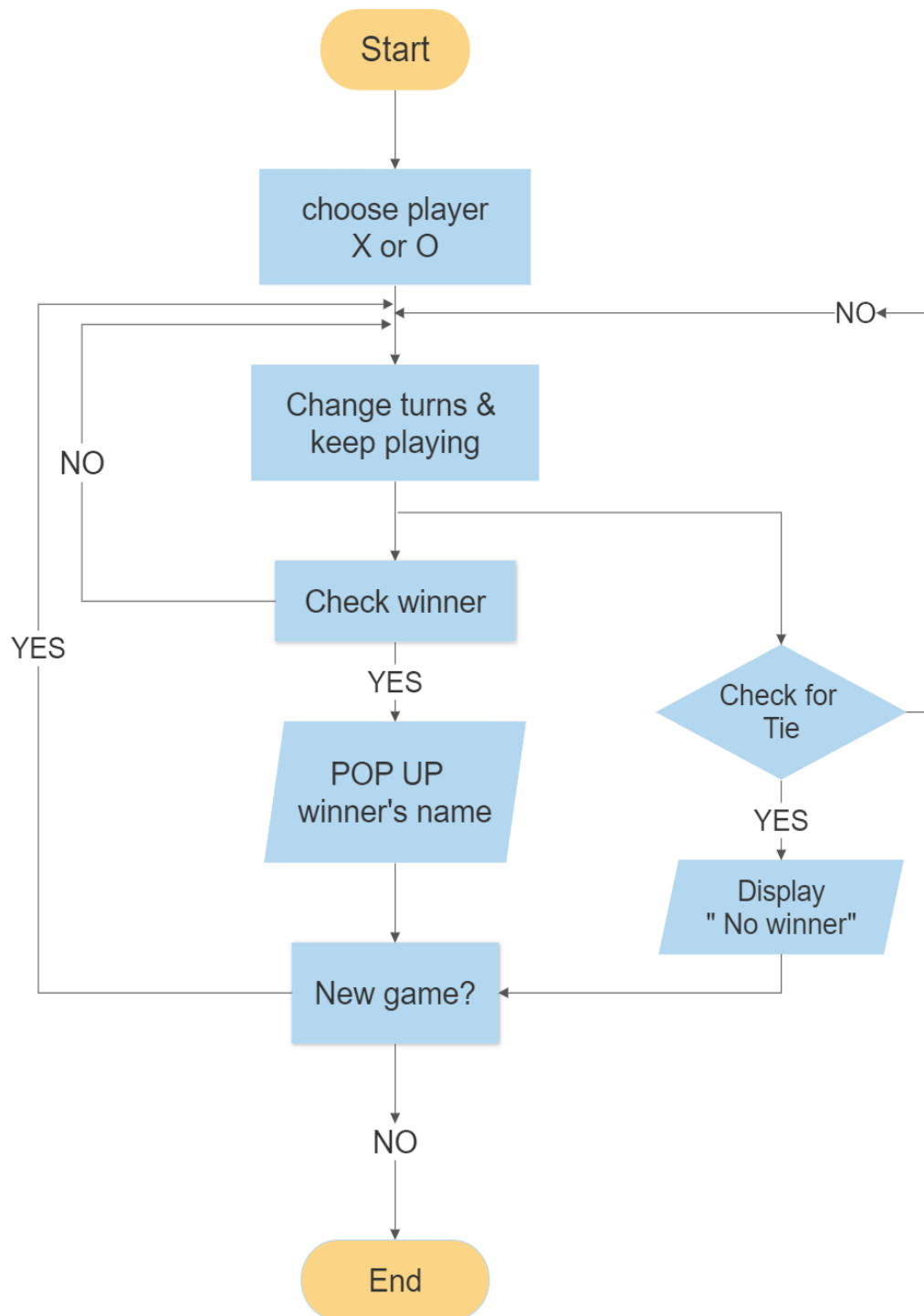
The game will be ended if one player win or if the game is tied ( if all the 9 buttons have been chosen and no player won ) ,the winner score will be increased by one .

There will be new stage showing the winner one if it is player **X** or **O** or the game is tied or there is no winner , then asking the player if he wants to play new game , if he clicked yes the marked x and o characters will be deleted and the winner of the last game will start the new game , if he choose NO the game will be closed .

## UML :



## FLOW CHART:



# Main Design Of The Game :

## The first stage:

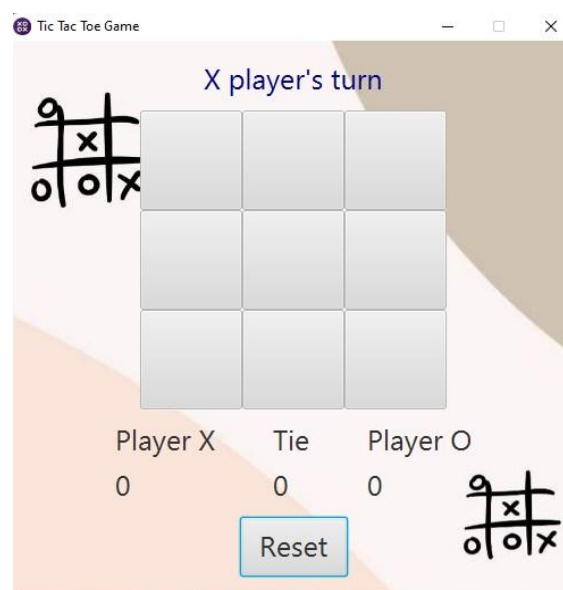
It contains two scenes

First scene asks the user to choose player to begin with

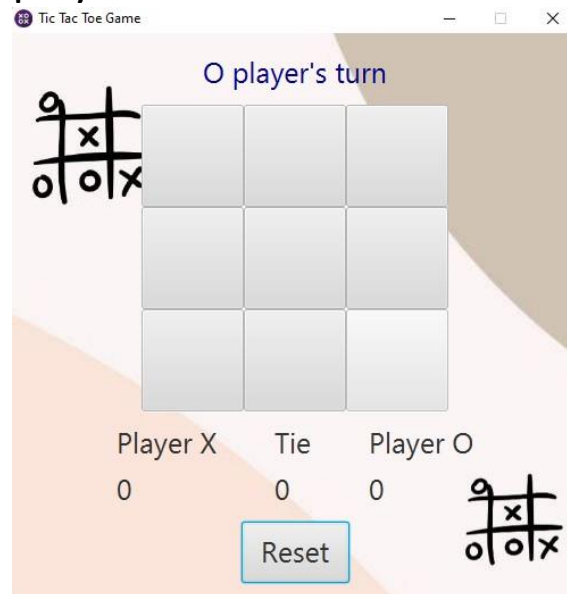


Main scene:

If you choose player X

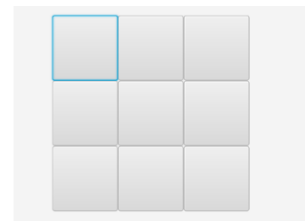


If you choose player O



First Grid pane :

- has the 9 buttons ( game board)



Second Grid pane : has

- Labels of player x , player O and tie.
- Labels that show the score of the players or if it is tied .
- Button reset that clear the texts on the all 9 buttons

Player X	Tie	Player O
0	0	0

Vertical box that has :

- title label
- pane
- pane2
- reset button

with defined gabs and insets .

## The second stage :

As shown in figure , appears in case of the game has ended even if there is a winner ( player X or player O ) or there is no winner ( game is tied ).

It shows which player wins , it has :

- label that displays the winner's name
- label new game
- two buttons that enable the user to play new game or close the game .

If player X wins



If player O wins



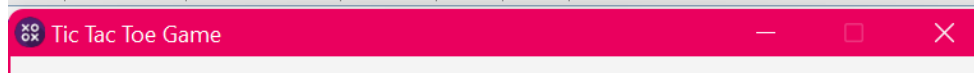
If there is no winner





## Adding icon to the Stage :

This indicates the image icon of the game in the title bar .



```
Image icon = new Image("unnamed.png");  
primaryStage.getIcons().add(icon);
```



## Used Imports:

```
import javafx.application.Application;  
import javafx.application.Platform;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.GridPane;  
import javafx.scene.layout.HBox;  
import javafx.scene.layout.StackPane;  
import javafx.scene.layout.VBox;  
import javafx.scene.paint.Color;  
import javafx.scene.text.Font;  
import javafx.stage.Stage;
```

## Used Methods :

```
@Override
public void start(Stage primaryStage) {...42 lines }

// Method to set up the buttons to receive X or O according to the player turn
public void setUpButton() {...50 lines }

public void setPlayerTurn(Button btn) {...30 lines }

public boolean checkWinner() {...52 lines }

public void disableBtn() {...11 lines }

public void calcScore() {...22 lines }

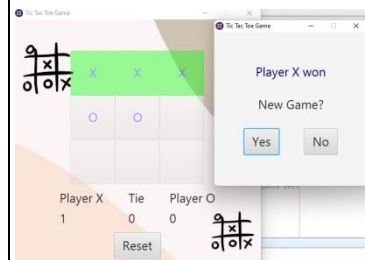
public boolean isFull() {...10 lines }

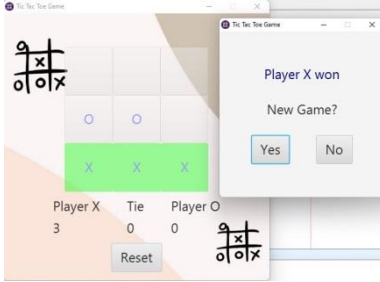
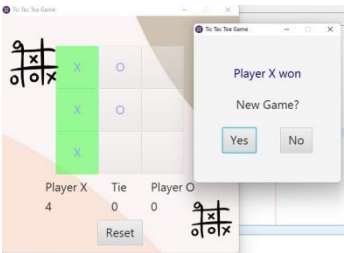
public void popUp() {...23 lines }
/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    launch(args);
}
```

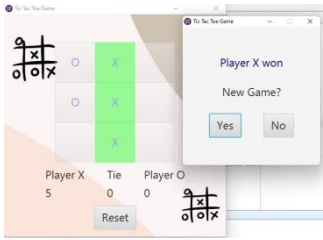
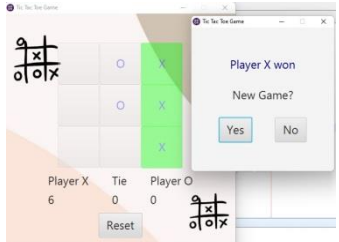
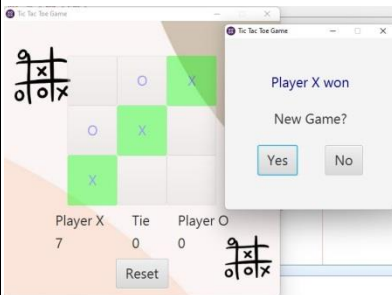
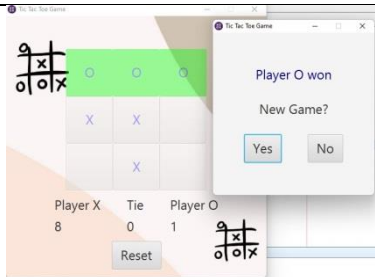
## Cases Of Output :

There are 17 cases in which game could be proceeded which are :

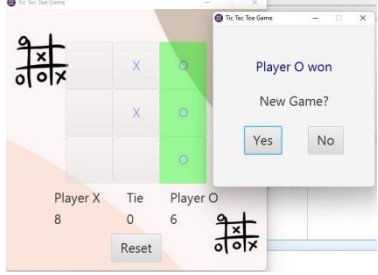
- 1 player x wins by completing first row indicating that by coloring the buttons of same sequenced characters by green color.



2	Player x wins by completing second row with x	
3	Player x wins by completing third row with x	
4	Player x wins by completing the left diagonal	
5	Player x wins by completing first column	

6	Player x wins by completing second column	
7	Player x wins by completing third columns	
8	Player x wins by completing the right diagonal	
1	player o wins by completing first row indicating that by coloring the buttons of same sequenced characters by green color.	

2	Player o wins by completing second row with o	
3	Player o wins by completing third row with o	
4	Player o wins by completing the left diagonal	
5	Player o wins by completing first column	
6	Player o wins by completing second column	

7	Player o wins by completing third columns	
8	Player o wins by completing the right diagonal	
	No winner case , all cell are clicked without getting winner	

## Some recommendations about ideas to extend this project in the future :

We can make this project more advanced by some additional options such as :

- Enabling the user to choose if he wants to play with the computer itself or with another player .
- Enabling the user to choose font style and size from selected options made by the developer.
- Enabling the user to select the design of the board or choose the default one .
- Enabling the user to choose the background he liked from selected options made by the developer.

## Problems Faced

- Some errors:
  - Local variables referenced from a lambda expression must be final
- In checkWinner method it was always end the game and show the second stage even if the user chooses only one button until we added ( `!= ""` ) in the if conditions to make check the buttons only with characters and not the empty ones .