**Subject: Object Oriented Programming**

**Time: MAY**

**BS (SE)**

**ID: 15815**

**Name: Ahmed Junaid**

**Answer sheet Instructor: M.Ayub Khan**

**There are a total of 1 questions in this paper. Max Marks: 20**

**Question 1**.**Create a Tic Tac Toe game in java (use any java tool for coding ) and explain it**

 **in detail including screenshots?**

**Answer:**

**Prerequisites:**

The tic-tac-toe game I made is in GUI and not CLI / console-based because CLI was too easy for me. The concept and the method used are the same as of CLI, so before I begin I want to explain tic tac toe logic.

TIC TAC TOE is a game based on 3x3 matrix which means

|  |  |  |
| --- | --- | --- |
| 1st element  | 2nd element  | 3rd element |
| 4th element  | 5th element | 6th element |
| 7th element  | 8th element  | 9th element |

So the idea of winning is if any row or Colum or diagonal row has the same player marks that player wins

If I look at this matrix thing I can see the use of multi-dimensional array so we create an array to hold the values and so (I will explain as I start explaining the program)

Now what libraries I used for GUI setup

**GUI Libraries:**

1. **Java.awt** **package:**

Awt short for Abstract Window Toolkit is an API for GUI development of java application this Contains all of the classes for creating user interfaces and for painting graphics and images. A user interface object such as a button or a scrollbar is called, in AWT terminology, a component. The Component class is the root of all AWT components.

This allows us to use event listeners on buttons, pretty much every GUI component is in this package which has the classes for awt API such as buttons, images, layouts, etc

1. **javax.swing:**

Provides a set of "lightweight" (all-Java language) components, to the maximum degree possible, work the same on all platforms

The swing library has modules like JFrame, JButton, JPanel and so on

Now lets go for the code:

**import java.awt.BorderLayout;**

**import java.awt.EventQueue;**

**import javax.swing.\*;**

**import javax.swing.border.EmptyBorder;**

**import java.awt.Color;**

**import java.awt.event.ActionListener;**

**import java.awt.event.ActionEvent;**

**import java.awt.Font;**

**import java.awt.GridLayout;**

**public class TicTacToeJFrame extends JFrame {**

 **private JPanel contentPane;**

 **public String[][] GameBoard = {{"1","2","3"},// initialising the gameboard array with numbers**

 **{"4","5","6"},**

 **{"7","8","9"}};**

 **public boolean turns = false; //if false its X turns and if true its O turns**

 **private int playeronecount = 0; // initialising player one and two count variable**

 **private int playertwocount = 0;// initialising player one and two count variable**

 **JButton first ;// Declaring Jbutton object for firstbtn**

 **JButton secondbtn;// Declaring Jbutton object for secondbtn**

 **JButton third ;// Declaring Jbutton object**

 **JButton fourth ;// Declaring Jbutton object**

 **JButton fifth ;// Declaring Jbutton object**

 **JButton sixth ;// Declaring Jbutton object**

 **JButton seventh ;// Declaring Jbutton object**

 **JButton eight ;// Declaring Jbutton object**

 **JButton nine ;// Declaring Jbutton object**

 **JLabel playerx;// Declaring playerx Jlabel Text object**

 **JLabel Playero; // Declaring playerO Jlabel Text object**

**//--------------------------main function --------------------------------------------**

 **public static void main(String[] args) {**

 **//EventQueue is a platform-independent classthat queues events, both from the underlying peer classesand from trusted application classes.**

**//It encapsulates asynchronous event dispatch machinery whichextracts events from the queue and dispatches them by calling dispatchEvent(AWTEvent) methodon this EventQueue**

 **EventQueue.invokeLater(new Runnable() {**

 **public void run() {**

**// creating a function to run and catching it with try and except/catch block to lauch the Jframe application window**

 **try {**

 **TicTacToeJFrame frame = new TicTacToeJFrame();**

 **frame.setVisible(true);**

 **} catch (Exception e) {**

 **e.printStackTrace();**

 **}**

 **}**

 **});**

 **}**

**//-------------------------------------------------------------------------------------------**

**//=========================Contructor===========================**

 **public TicTacToeJFrame() {**

 **setTitle("Tic Tac Toe");**

 **setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

 **setBounds(100, 100, 831, 552);**

 **contentPane = new JPanel();**

 **contentPane.setBackground(Color.BLACK);**

 **contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));**

 **setContentPane(contentPane);**

 **contentPane.setLayout(new GridLayout(0, 5, 0, 0));**

 **//----------------------------Setup----------------------------------------**

 **first = new JButton("");**

 **first.setBackground(Color.black);**

 **first.setFont(new Font("Rockwell Extra Bold", Font.PLAIN, 40));**

 **contentPane.add(first);**

 **secondbtn = new JButton("");**

 **secondbtn.setBackground(Color.black);**

 **secondbtn.setFont(new Font("Rockwell Extra Bold", Font.PLAIN, 40));**

 **contentPane.add(secondbtn);**

 **third = new JButton("");**

 **third.setBackground(Color.black);**

 **contentPane.add(third);**

 **third.setFont(new Font("Rockwell Extra Bold", Font.PLAIN, 40));**

 **JLabel label\_2 = new JLabel("Made By Ahmed");**

 **label\_2.setFont(new Font("Tahoma", Font.BOLD, 17));**

 **label\_2.setHorizontalAlignment(SwingConstants.CENTER);**

 **label\_2.setForeground(Color.ORANGE);**

 **contentPane.add(label\_2);**

 **JLabel label\_5 = new JLabel("Tic-Tac-Toe");**

 **label\_5.setFont(new Font("Tahoma", Font.BOLD, 18));**

 **label\_5.setForeground(Color.GREEN);**

 **label\_5.setHorizontalAlignment(SwingConstants.CENTER);**

 **contentPane.add(label\_5);**

 **fourth = new JButton("");**

 **fourth.setBackground(Color.black);**

 **contentPane.add(fourth);**

 **fourth.setFont(new Font("Rockwell Extra Bold", Font.PLAIN, 40));**

 **fifth = new JButton("");**

 **fifth.setBackground(Color.black);**

 **contentPane.add(fifth);**

 **fifth.setFont(new Font("Rockwell Extra Bold", Font.PLAIN, 40));**

 **sixth = new JButton("");**

 **sixth.setBackground(Color.black);**

 **contentPane.add(sixth);**

 **sixth.setFont(new Font("Rockwell Extra Bold", Font.PLAIN, 40));**

 **JLabel label = new JLabel("");**

 **contentPane.add(label);**

 **JLabel label\_1 = new JLabel("");**

 **contentPane.add(label\_1);**

 **seventh = new JButton("");**

 **seventh.setBackground(Color.black);**

 **seventh.setFont(new Font("Rockwell Extra Bold", Font.PLAIN, 40));**

 **contentPane.add(seventh);**

 **eight = new JButton("");**

 **eight.setBackground(Color.black);**

 **contentPane.add(eight);**

 **eight.setFont(new Font("Rockwell Extra Bold", Font.PLAIN, 40));**

 **nine = new JButton("");**

 **nine.setBackground(Color.black);**

 **contentPane.add(nine);**

 **nine.setFont(new Font("Rockwell Extra Bold", Font.PLAIN, 40));**

 **JLabel label\_6 = new JLabel("");**

 **contentPane.add(label\_6);**

 **JLabel label\_7 = new JLabel("");**

 **contentPane.add(label\_7);**

 **JLabel player\_2 = new JLabel("Player2");**

 **player\_2.setHorizontalAlignment(SwingConstants.CENTER);**

 **player\_2.setForeground(new Color(0, 102, 204));**

 **player\_2.setFont(new Font("Yu Gothic UI", Font.BOLD, 30));**

 **contentPane.add(player\_2);**

 **Playero = new JLabel("0");**

 **Playero.setFont(new Font("Tahoma", Font.PLAIN, 30));**

 **Playero.setHorizontalAlignment(SwingConstants.CENTER);**

 **Playero.setForeground(Color.WHITE);**

 **contentPane.add(Playero);**

 **JLabel label\_8 = new JLabel("");**

 **contentPane.add(label\_8);**

 **JLabel label\_9 = new JLabel("");**

 **contentPane.add(label\_9);**

 **JLabel label\_10 = new JLabel("");**

 **contentPane.add(label\_10);**

 **JLabel player\_1 = new JLabel("Player1");**

 **player\_1.setHorizontalAlignment(SwingConstants.CENTER);**

 **player\_1.setFont(new Font("Yu Gothic UI", Font.BOLD, 30));**

 **player\_1.setForeground(new Color(153, 0, 0));**

 **contentPane.add(player\_1);**

 **playerx = new JLabel("0");**

 **playerx.setHorizontalAlignment(SwingConstants.CENTER);**

 **playerx.setFont(new Font("Tahoma", Font.PLAIN, 30));**

 **playerx.setForeground(Color.WHITE);**

 **contentPane.add(playerx);**

 **JLabel label\_11 = new JLabel("");**

 **contentPane.add(label\_11);**

 **JButton Reset = new JButton("Reset");**

 **Reset.setForeground(Color.RED);**

 **Reset.setBackground(Color.BLACK);**

 **Reset.setFont(new Font("Tahoma", Font.PLAIN, 17));**

 **contentPane.add(Reset);**

 **JLabel label\_12 = new JLabel("");**

 **contentPane.add(label\_12);**

 **//------------------------------------------------------------------------------------------------------------**

 **//------------------------------EVENT Listeners--------------------------------------------------------**

 **first.addActionListener(new ActionListener() {**

 **public void actionPerformed(ActionEvent e) {**

 **Turn(first,1);**

 **}**

 **});**

 **secondbtn.addActionListener(new ActionListener() {**

 **public void actionPerformed(ActionEvent e) {**

 **Turn(secondbtn,2);**

 **}**

 **});**

 **third.addActionListener(new ActionListener() {**

 **public void actionPerformed(ActionEvent e) {**

 **Turn(third,3);**

 **}**

 **});**

 **fourth.addActionListener(new ActionListener() {**

 **public void actionPerformed(ActionEvent e) {**

 **Turn(fourth,4);**

 **}**

 **});**

 **fifth.addActionListener(new ActionListener() {**

 **public void actionPerformed(ActionEvent e) {**

 **Turn(fifth,5);**

 **}**

 **});**

 **sixth.addActionListener(new ActionListener() {**

 **public void actionPerformed(ActionEvent e) {**

 **Turn(sixth,6);**

 **}**

 **});**

 **seventh.addActionListener(new ActionListener() {**

 **public void actionPerformed(ActionEvent e) {**

 **Turn(seventh,7);**

 **}**

 **});**

 **eight.addActionListener(new ActionListener() {**

 **public void actionPerformed(ActionEvent e) {**

 **Turn(eight,8);**

 **}**

 **});**

 **nine.addActionListener(new ActionListener() {**

 **public void actionPerformed(ActionEvent e) {**

 **Turn(nine,9);**

 **}**

 **});**

 **Reset.addActionListener(new ActionListener() {**

 **public void actionPerformed(ActionEvent e) {**

 **ResetTheBoard();**

 **}**

 **});**

 **//---------------------------------------------------------------------------------------------------------**

 **}//End of Constructor ===============================================**

 **void ResetTheBoard() {**

 **first.setText("");**

 **secondbtn.setText("");**

 **third.setText("");**

 **fourth.setText("");**

 **fifth.setText("");**

 **sixth.setText("");**

 **seventh.setText("");**

 **eight.setText("");**

 **nine.setText("");**

 **int x = 1;**

 **for(int r = 0; r<3;r++) {**

 **// 0 = 1 1= 2 2= 3**

 **for(int c = 0;c<3;c++) {**

 **GameBoard[r][c] = Integer.toString(x);**

 **x++;**

 **}**

 **}//End of For Loop**

 **}**

 **public void Turn(JButton x,int j) {**

 **String i = x.getText();**

 **if(i.isEmpty()) {**

 **if(turns) {**

 **x.setText("O");**

 **x.setForeground(Color.BLUE);**

 **Mark(j,"O");**

 **GameCheck("O");**

 **}else {**

 **x.setText("X");**

 **x.setForeground(Color.red);**

 **Mark(j,"X");**

 **GameCheck("X");**

 **}**

 **turns = !turns;**

 **}**

 **}**

 **public void Mark(int x,String currentturn) {**

 **switch (x) {**

 **case 1:**

 **if (GameBoard[0][0] != "X" && GameBoard[0][0] != "O" ) {**

 **GameBoard[0][0] =currentturn;**

 **}else{**

 **ShowDialog("already used choose different!");**

 **}**

 **break;**

 **case 2:**

 **if (GameBoard[0][1] != "X" && GameBoard[0][1] != "O" ) {**

 **GameBoard[0][1] = currentturn;**

 **}else{**

 **ShowDialog("already used choose different!");**

 **}**

 **break;**

 **case 3:**

 **if (GameBoard[0][2] != "X" && GameBoard[0][2] != "O" ) {**

 **GameBoard[0][2] = currentturn;**

 **}else{**

 **ShowDialog("already used choose different!");**

 **}**

 **break;**

 **case 4:**

 **if (GameBoard[1][0] != "X" && GameBoard[1][0] != "O" ) {**

 **GameBoard[1][0] = currentturn;**

 **}else{**

 **ShowDialog("already used choose different!");**

 **}**

 **break;**

 **case 5:**

 **if (GameBoard[1][1] != "X" && GameBoard[1][1] != "O" ) {**

 **GameBoard[1][1] = currentturn;**

 **}else{**

 **ShowDialog("already used choose different!");**

 **}**

 **break;**

 **case 6:**

 **if (GameBoard[1][2] != "X" && GameBoard[1][2] != "O'") {**

 **GameBoard[1][2] = currentturn;**

 **}else{**

 **ShowDialog("already used choose different!");**

 **}**

 **break;**

 **case 7:**

 **if (GameBoard[2][0] != "X" && GameBoard[2][0] != "O" ) {**

 **GameBoard[2][0] = currentturn;**

 **}else{**

 **ShowDialog("already used choose different!");**

 **}**

 **break;**

 **case 8:**

 **if (GameBoard[2][1] != "X" && GameBoard[2][1] != "O" ) {**

 **GameBoard[2][1] = currentturn;**

 **}else{**

 **ShowDialog("already used choose different!");**

 **}**

 **break;**

 **case 9:**

 **if (GameBoard[2][2] != "X" && GameBoard[2][2] != "O") {**

 **GameBoard[2][2] = currentturn;**

 **}else{**

 **ShowDialog("already used choose different!");**

 **}**

 **break;**

 **default:**

 **break;**

 **}**

 **}**

 **public void GameCheck(String turn) {**

 **if((GameBoard[0][0] == GameBoard[0][1] && GameBoard[0][1] == GameBoard[0][2]) ||**

 **(GameBoard[0][0] == GameBoard[1][0] && GameBoard[0][0] == GameBoard[2][0]) ||**

 **(GameBoard[0][2] == GameBoard[1][2] && GameBoard[0][2] == GameBoard[2][2]) ||**

 **(GameBoard[1][0] == GameBoard[1][1] && GameBoard[1][1] == GameBoard[1][2]) ||**

 **(GameBoard[2][0] == GameBoard[2][1] && GameBoard[2][1] == GameBoard[2][2])||**

 **(GameBoard[0][0] == GameBoard[1][0] && GameBoard[1][0] == GameBoard[2][0])||**

 **(GameBoard[0][1] == GameBoard[1][1] && GameBoard[1][1] == GameBoard[2][1])||**

 **(GameBoard[0][2] == GameBoard[1][2] && GameBoard[1][2] == GameBoard[2][2])||**

 **(GameBoard[0][0] == GameBoard[1][1] && GameBoard[1][1] == GameBoard[2][2])||**

 **(GameBoard[0][2] == GameBoard[1][1] && GameBoard[1][1] == GameBoard[2][0])){**

 **/\*Logic mapping**

 **\* 0|0 0|1 0|2**

 **\* 1|0 1|1 1|2**

 **\* 2|0 2|1 2|2**

 **\* \*/**

 **if(turn == "X"){**

 **ShowDialog("Player 1 wins!");**

 **playeronecount++;**

 **playerx.setText(Integer.toString(playeronecount));**

 **ResetTheBoard();**

 **}else if(turn == "O") {**

 **ShowDialog("Player 2 wins!");**

 **playertwocount++;**

 **Playero.setText(Integer.toString(playertwocount));**

 **ResetTheBoard();**

 **}**

 **}**

 **}**

 **public void ShowDialog(String error) {**

 **JFrame dialog = new JFrame();**

 **JOptionPane.showConfirmDialog(dialog, error);**

 **}**

**}**

**Explanation:**

1. I started by importing the required library which I already explained earlier javax.swing.\* means all the modules/classes from swing package
2. Then I created my main public class named TicTacToeJFrame which is inherited from JFrame class so what extends keyword means is we are extending the superclass of JFrame to child class TicTacToe we did this because I am trying to achieve a GUI
3. Now inside the main class, I created few variables such as Jpanel which holds the Content together like a pane so I am declaring JPanel object as contentPane
4. Then im initializing an array which I explained earlier named GameBoard 3x3

After this, I initialize the logic of player switches and I achieved that by setting the **game protocols** if turns (which is a Boolean) is false that means its playerx turn and vice versa

Then I initialized 2 private int to hold the wins of both player

After that, I declared JButton Objects all of them 3x3= total 9

And also Jlabel of playerx and playero count display text

(note: why did I declare these ones outside the constructor is because the objects created inside a constructor or any method is local and can not be called from outside the method by doing this, I created the instance of these object so that I could access them later.

1. Now I created the main function inside it I called the EventQueue object method named invokeLater which accepts a function Runnable() which is present in java.lang.Runnable class inside it I created a function run() which try to create my main class object(TicTacToeJFrame) and set it to visible which is a method inside JFrame class which my class extends from I made it in try and except/catch block to be on safe site if for any reason it doesn’t create the object. To be safe from runtime errors that’s all we did in the main function
2. Now moving on to over Constructor where we set up the layout, event listeners and much more

 **setTitle("Tic Tac Toe");**

 **setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

 **setBounds(100, 100, 831, 552);**

 **contentPane = new JPanel();**

 **contentPane.setBackground(Color.BLACK);**

 **contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));**

 **setContentPane(contentPane);**

 **contentPane.setLayout(new GridLayout(0, 5, 0, 0));**

I am using the inherited methods like setTitle etc to set up the layout of my application. Initialize the content pane object and set the background color to black and set the border to a new Emptyborder object. Setborder method is from the jpanel object as I declared the contentpane as jpanel object

1. Then comes the button setup by looks part where I initialized my already declared JButton object and setbackground color and font size etc I also made them empty text as default by passing an empty string to the constructor

 **first = new JButton("");**

 **first.setBackground(Color.black);**

 **first.setFont(new Font("Rockwell Extra Bold", Font.PLAIN, 40));**

 **contentPane.add(first);**

and did this with all the 9 buttons this is just for the looks and layout of the button how they look on the application window

1. Now I attached the event listener to all of my buttons to listen for the event such as pressed etc I did this by using a method inside the JButton class which is used to add the listen on the object in an optimized way as I said earlier javax.swing optimizes and makes it uniform the method is named addActionListener which accepts an ActionListener() which is present inside the java.Awt toolset, inside it we have actionPerformed function which takes ActionEvent as parameter I made this listener for each button

**first.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

 **Turn(first,1);**

**}**

**});**

And I also have a custom function called Turn which I will create at the end of class which accept the JButton Object and the number for each button as index as I explained earlier that how tictactoe works

1. Lastly I added the Event listener for reset button which I also initialized in contructor setup and called a custom function upon pressed named ResetTheBoard()
2. Now outside the constructor I created few function that holds the logic behind the game
3. Void ResetTheBoard:

**void ResetTheBoard() {**

 **first.setText("");**

 **secondbtn.setText("");**

 **third.setText("");**

 **fourth.setText("");**

 **fifth.setText("");**

 **sixth.setText("");**

 **seventh.setText("");**

 **eight.setText("");**

 **nine.setText("");**

 **int x = 1;**

 **for(int r = 0; r<3;r++) {**

 **// 0 = 1 1= 2 2= 3**

 **for(int c = 0;c<3;c++) {**

 **GameBoard[r][c] = Integer.toString(x);**

 **x++;**

 **}**

 **}//End of For Loop**

 **}**

This function set the text of each button to an empty string and reset the GameBoard array to 1,2,3,4,5,6,7,8,9 as default values

I used a nested for loop to go through each Colum of each row of the GameBoard array here is a general view of the GameBoard array

When we declare the array by [3][3] This means we have we have 2 space to hold data but because array data storage starts with 0 we get 3 spaces

I mean if I create an array of int[4] x = {1,2,3,4}

We cannot access the 3 by using x[3] that will return 4

|  |  |  |
| --- | --- | --- |
|  Colum1  | Colum2 | Colum3 |
| Row1 | 1 | 2 | 3 |
| Row2 | 4 | 5 | 6 |
| Row3 | 7 | 8 | 9 |

1. Then I created The Turn function which takes JButton and int as parameter inside this function I initialized a String variable I, and initialized it with the JButton which was passed by text by using a method of JButton getText();
2. Then I checked if the I variable( the button text is empty which means there hasn’t been assigned by anything) is empty by using if statement and inside this if statement block I again used an if-else statement to check if turns are true as I said earlier if the turns = true that means its player o turns and false = player x. inside each of these if condition block I set appropriately the symbol x/o according to the player turn and set each of the text color different I choose red for x and blue for o. then I called 2 functions which I will make after this. called Mark which accept the index and the current player symbol

And A GameCheck function which accepts current Player turn if we think why I'm using this GameCheck on current user well because this function is called every button press so if the player takes his turn we check if he wins or not

Finally I make the turns Boolean reaverse of what it was already like to false to true and true to false after every turn by using turns = !turns

The “!” is used for is not whatever like if we say x = !true which will assign the value of x by false because is not true = false

 **public void Turn(JButton x,int j) {**

 **String i = x.getText();**

 **if(i.isEmpty()) {**

 **if(turns) {**

 **x.setText("O");**

 **x.setForeground(Color.BLUE);**

 **Mark(j,"O");**

 **GameCheck("O");**

 **}else {**

 **x.setText("X");**

 **x.setForeground(Color.red);**

 **Mark(j,"X");**

 **GameCheck("X");**

 **}**

 **turns = !turns;**

 **}**

 **}**

1. Now after that function I created the Mark Function it does exactly what it sounds it marks the GameBoard with the player in array form this function takes 2 parameters int x and String currentturn , now inside this function, I used a very handy conditional statement SWITCH statement it takes a value and matches the case if the case meets the value that was passed the block of that case is invoked and breaks the flow after the case is executed I checked for the case for index and performed accordingly to what index represent like if the index that was passed was less than 4 it will be on row 0 which in real space is 1

So I checked inside each case block that if any of the symbols are assigned to that index (what I mean by symbols are x and o) if neither of them is assigned we assign the current user to it to own that place in else block I used yet another custom function which I made to help out to print dialog box to the screen this function is called ShowDialog and has String as a parameter

**public void Mark(int x,String currentturn) {**

 **switch (x) {**

 **case 1:**

 **if (GameBoard[0][0] != "X" && GameBoard[0][0] != "O" ) {**

 **GameBoard[0][0] =currentturn;**

 **}else{**

 **ShowDialog("already used choose different!");**

 **}**

 **break;**

 **case 2:**

 **if (GameBoard[0][1] != "X" && GameBoard[0][1] != "O" ) {**

 **GameBoard[0][1] = currentturn;**

 **}else{**

 **ShowDialog("already used choose different!");**

 **}**

 **break;**

 **case 3:**

 **if (GameBoard[0][2] != "X" && GameBoard[0][2] != "O" ) {**

 **GameBoard[0][2] = currentturn;**

 **}else{**

 **ShowDialog("already used choose different!");**

 **}**

 **break;**

 **case 4:**

 **if (GameBoard[1][0] != "X" && GameBoard[1][0] != "O" ) {**

 **GameBoard[1][0] = currentturn;**

 **}else{**

 **ShowDialog("already used choose different!");**

 **}**

 **break;**

 **case 5:**

 **if (GameBoard[1][1] != "X" && GameBoard[1][1] != "O" ) {**

 **GameBoard[1][1] = currentturn;**

 **}else{**

 **ShowDialog("already used choose different!");**

 **}**

 **break;**

 **case 6:**

 **if (GameBoard[1][2] != "X" && GameBoard[1][2] != "O'") {**

 **GameBoard[1][2] = currentturn;**

 **}else{**

 **ShowDialog("already used choose different!");**

 **}**

 **break;**

 **case 7:**

 **if (GameBoard[2][0] != "X" && GameBoard[2][0] != "O" ) {**

 **GameBoard[2][0] = currentturn;**

 **}else{**

 **ShowDialog("already used choose different!");**

 **}**

 **break;**

 **case 8:**

 **if (GameBoard[2][1] != "X" && GameBoard[2][1] != "O" ) {**

 **GameBoard[2][1] = currentturn;**

 **}else{**

 **ShowDialog("already used choose different!");**

 **}**

 **break;**

 **case 9:**

 **if (GameBoard[2][2] != "X" && GameBoard[2][2] != "O") {**

 **GameBoard[2][2] = currentturn;**

 **}else{**

 **ShowDialog("already used choose different!");**

 **}**

 **break;**

 **default:**

 **break;**

 **} }**

Then I created GameCheck function which is the function used to check if the gameboard is won or not.inside this function i check for each combination /\*Logic mapping

 \* 0|0 0|1 0|2

 \* 1|0 1|1 1|2

 \* 2|0 2|1 2|2

 \* \*/

 So as I know the combination which I already told earlier in TicTacToe game introduction. I used if statement to check for if 1st element and 2nd element are same if yes we used and (&&) operator to further check if 2nd element and 3rd element of 1 row are same if yes that player won I checked for all the possible combination (total of 8 different ways of winning this game) if we win then the if statement block execute and it has another if statement to check for whose turn was it and increment the score of that by using ++ short for 1+1 or 1 += 1

And also use the ShowDialog function with a message saying that player won and also calling ResetThBoard function to reset the game

1. Last function I made is kinda helper ShowDialog which takes a String As parameter message and uses that msg as to display it inside this function I initialized the JFrame object called it dialog. then using the JOptionPane object and using the showConfrimDialog method to display a message. This method takes 2 arguments first one is t JFrame object and the other one is the message itself which we passed to the function upon calling

**OUTPUT:**



 Note: The Jar file is included in the assignment zip file ☺

The Server is not accepting zip files or any other compression here’s link valid for 30 days upload time 6/5/2020

https://ufile.io/vzkwwjjm