**INNOVATIVE FOA : TIC TAC TOE GAME IN JAVA**

This is a tic tac toe game which is made using java and it is a console game , where we have to play against a computer and to win the game , here I have made use of list ,collections and other many others along with a random library which is use to give a random values and it follows an algorithm which I have explained in working.

**Roles :**

User : ‘X’

Computer : ‘Y

**Working :**

So there are 2 players one is a user who will place ‘X’ in the 2d array which I stored by taking an input from user who will be entering the position of that place and that character ‘X’ will be place over there , and the other is a computer , which will generate a random number between 1 to 9 to place over the game board and then the ‘0’ will be place over there , also at the same time it will check whether the place is already occupied by the user or not , if its already occupied then user and computer will store at another value by asking for user and in case of computer , it will generate again a random position , then it will check whether the values stored in the array are in some form of winning position , which I have stored it ina a array and the positions will be check every time the user will enter the values and in case , when user or the computers values math with the winning positions then it will print the message and break from the loop , also this will happen with the draw condition where it will check the length of user and computers input are more than 9 , in that case also it will print the message and break out of the loop , and then the game ends.

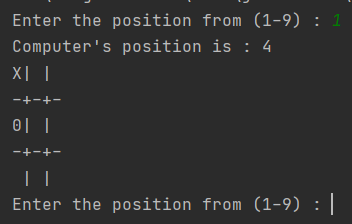
**Java Code : TicTacToe.java**

import java.util.\*;  
  
public class TicTacToe {  
  
 // These are to store the positions  
 static ArrayList<Integer> *playerPositions* =new ArrayList<>();  
 static ArrayList<Integer> *computerPositions* =new ArrayList<>();  
  
 public static void main(String[] args) {  
 char [][] gameBoard = {{' ','|',' ','|',' '},  
 {'-','+','-','+','-'},  
 {' ','|',' ','|',' '},  
 {'-','+','-','+','-'},  
 {' ','|',' ','|',' '} };  
  
 while (true) {  
  
 // This will ask the user for entering the position  
 System.*out*.print("Enter the position from (1-9) : ");  
 Scanner scan = new Scanner(System.*in*);  
 int UserPos = scan.nextInt();  
  
 // This will check if , input by user is present in the user positions and computer positions and then will ask to input again if present  
 while (*playerPositions*.contains(UserPos) || *computerPositions*.contains(UserPos)){ // or  
 System.*out*.println("Position taken , Please choose another position !!");  
 UserPos = scan.nextInt();  
 }  
  
 *choiceGameBoard*(gameBoard, UserPos, "User");  
  
 // checks for winner after entering val for user  
 String result = *checkWinner*();  
 if (result.length()>0){  
 System.*out*.println(result);  
 break;  
 }  
  
 // This is to generate random number for computer  
 Random rand = new Random();  
 int cpuPos = rand.nextInt(9) + 1;  
 System.*out*.println("Computer's position is : " + cpuPos);  
  
 // This will check if , input by computer is present in the user positions and computer positions array and then will again generate another one  
 while (*computerPositions*.contains(cpuPos) || *playerPositions*.contains(cpuPos)){  
 cpuPos = rand.nextInt(9)+1;  
 }  
  
 *choiceGameBoard*(gameBoard, cpuPos, "computer");  
  
 // This will print the values of X and 0 in board  
 *printGameBoard*(gameBoard);  
  
 // checks for winner after entering val for computer  
 result = *checkWinner*();  
 if(result.length()>0){  
 System.*out*.println(result);  
 break;  
 }  
  
 }  
  
 }  
  
 // This for printing the values on board  
 public static void choiceGameBoard(char[][] gameBoard , int pos, String role){  
  
 char symbol=' ';  
 if(role.equals("User"))  
 {  
 symbol='X';  
 *playerPositions*.add(pos);  
 }  
 else if (role.equals("computer")){  
 symbol='0';  
 *computerPositions*.add(pos);  
 }  
  
 switch (pos) {  
 case 1:  
 gameBoard[0][0]=symbol;  
 break;  
 case 2:  
 gameBoard[0][2]=symbol;  
 break;  
 case 3:  
 gameBoard[0][4]=symbol;  
 break;  
 case 4:  
 gameBoard[2][0]=symbol;  
 break;  
 case 5:  
 gameBoard[2][2]=symbol;  
 break;  
 case 6:  
 gameBoard[2][4]=symbol;  
 break;  
 case 7:  
 gameBoard[4][0]=symbol;  
 break;  
 case 8:  
 gameBoard[4][2]=symbol;  
 break;  
 case 9:  
 gameBoard[4][4]=symbol;  
 break;  
 default:  
 break;  
 }  
  
 }  
  
 //This for printing the board  
 public static void printGameBoard(char[][] gameBoard){  
 for (char[] row : gameBoard) {  
 for (char c : row) {  
 System.*out*.print(c);  
 }  
 System.*out*.println();  
 }  
 }  
  
 // This is to check the winners  
 public static String checkWinner() {  
  
 // These are all winner positions  
 List topRow= Arrays.*asList*(1,2,3);  
 List midRow= Arrays.*asList*(4,5,6);  
 List botRow= Arrays.*asList*(7,8,9);  
 List leftCol= Arrays.*asList*(1,4,7);  
 List midCol= Arrays.*asList*(2,5,8);  
 List rightCol= Arrays.*asList*(3,6,9);  
 List cross1= Arrays.*asList*(1,5,9);  
 List cross2= Arrays.*asList*(3,5,7);  
  
 List<List> winningPos = new ArrayList<List>();  
 winningPos.add(topRow);  
 winningPos.add(midRow);  
 winningPos.add(botRow);  
 winningPos.add(leftCol);  
 winningPos.add(midCol);  
 winningPos.add(rightCol);  
 winningPos.add(cross1);  
 winningPos.add(cross2);  
  
 // this will traverse through winningPos and check  
 for (List l :winningPos) {  
 if (*playerPositions*.containsAll(l)){  
 return "Congratulations , you won the game !!";  
 }  
 else if(*computerPositions*.containsAll(l)){  
 return "Sorry , you lose by Computer :( , thanks for playing !!";  
 }  
 else if(*playerPositions*.size()+*computerPositions*.size()==9){  
 return "This game is a Draw , please try again !!";  
 }  
 }  
 return "";  
 }  
}

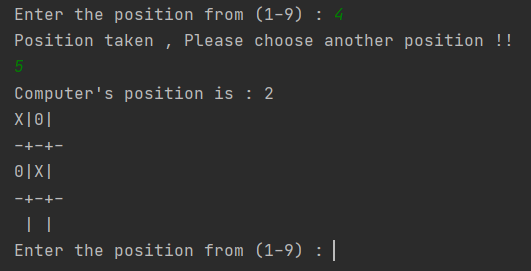
**Output :**

1) First the user will enter the position number and X will be placed over there

2) Then a computer has inserted the 0 in a random position



3) Here , in case of position already occupied , it will ask us again



4) Then we will follow these above steps unless we reach a conclusion

