Vidyavardhini’s College of Engineering & Technology

Department of Artificial Intelligence and Data Science

Experiment No. 10
Topic : Course Project
Name: Saurabh Mane
Roll Number: 33
Date of Performance:
Date of Submission:

Experiment No. 10 Mini Project

Theory:

Tic-tac-toe is a simple two-player game typically played on a 3x3 grid. While it may not seem like a complex game, there are several interesting aspects and theories related to it. Here are a few theories and strategies associated with tic-tac-toe:

Game Complexity: Tic-tac-toe is often used as an introductory example in the study of game theory. It's a solved game, meaning that with perfect play from both players, the outcome is known. The game will either end in a draw (a cat's game) or a win for one of the players. This makes it a good model for exploring concepts like decision trees and strategy in game theory.

Winning Strategies: When both players make perfect moves, the game should end in a draw. This is because it's relatively easy to force a draw by either player. If one player makes a mistake, the other can capitalize on it and win. For instance, if you go first and take the center square, your opponent can always force a draw with perfect play.

The First Move Advantage: In tic-tac-toe, the player who goes first (typically "X") has a slight advantage when both players play perfectly. This is because the first player can always secure a win or a draw with optimal moves, while the second player can only hope for a draw in most cases.

Common Winning Patterns: There are eight possible ways to win in tic-tac-toe (three rows, three columns, and two diagonals). Understanding these winning patterns is crucial to forming a winning strategy. For example, if you have two of your symbols in a row, column, or diagonal, you should aim to complete the sequence to win or block your opponent from completing their sequence.

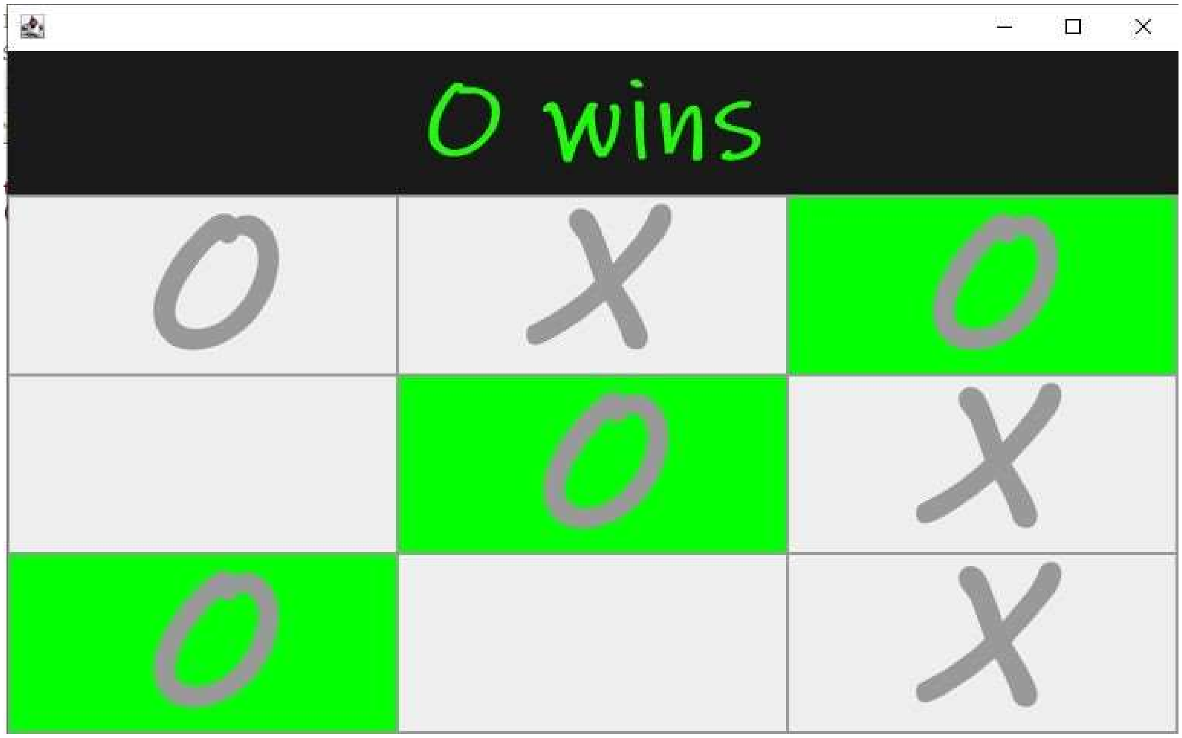
Program:

```
//***************************************
public class Main { public static void
main(String[] args) {

                TicTacToe tictactoe = new TicTacToe();

        }
}
//***************************************
import java.awt.*; import java.awt.event.*; import
java.util.*; import javax.swing.*; public class
TicTacToe implements ActionListener{

        Random random = new Random();
        JFrame frame = new JFrame();
        JPanel title_panel = new JPanel();
        JPanel button_panel = new JPanel();
        JLabel textfield = new JLabel();
        JButton[] buttons = new JButton[9];
        boolean player1_turn;

        TicTacToe(){

                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                frame.setSize(800,800);
                frame.getContentPane().setBackground(new Color(50,50,50));
                frame.setLayout(new BorderLayout()); frame.setVisible(true);

                textfield.setBackground(new Color(25,25,25));
                textfield.setForeground(new Color(25,255,0));
                textfield.setFont(new Font("Ink Free",Font.BOLD,75));
                textfield.setHorizontalAlignment(JLabel.CENTER);
                textfield.setText("Tic-Tac-Toe");
                textfield.setOpaque(true);
                title_panel.setLayout(new BorderLayout());
                title_panel.setBounds(0,0,800,100);

                button_panel.setLayout(new GridLayout(3,3));
                button_panel.setBackground(new Color(150,150,150));

                for(int  i=0;i<9;i++)  {  buttons[i]  =
                        new                   JButton();
                        button_panel.add(buttons[i])
                        ;        buttons[i].setFont(new
                        Font("MV
                        Boli",Font.BOLD,120));
                        buttons[i].setFocusable(false);
                        buttons[i].addActionListener(this);
                }

                title_panel.add(textfield);
                frame.add(title_panel,BorderLayout.NORTH);
                frame.add(button_panel);

                firstTurn();
        }

        @Override
        public void actionPerformed(ActionEvent e) {

                for(int i=0;i<9;i++) { if(e.getSource()==buttons[i]) { if(player1_turn) {
                        if(buttons[i].getText()=="") { buttons[i].setForeground(new
                        Color(255,0,0));
                                        buttons[i].setText("X");
                                        player1_turn=false;
                                        textfield.setText("O turn");
                                        check();
                                }
                        } else { if(buttons[i].getText()=="") {
                        buttons[i].setForeground(new Color(0,0,255));
                                        buttons[i].setText("O");
                                        player1_turn=true;
                                        textfield.setText("X turn");
                                        check();
                                }
                        }
                    }
                }
        } public void firstTurn()

        {
                try {
                        Thread.sleep(2000);
                } catch (InterruptedException e) { // TODO
                Auto-generated catch block
                e.printStackTrace(); }

                if(random.nextInt(2)==0) {
                        player1_turn=true;
                        textfield.setText("X turn");
                } else
                {
                playe
                r1_tu
                rn=fa
                lse;
                        textfield.setText("O turn");
                }
        }

        public void check() { //check X
                win conditions
                if(
                                (buttons[0].getText()=="X") &&
                                (buttons[1].getText()=="X") &&
                                (buttons[2].getText()=="X")
                                ) {
                        xWins(0,1,2);
                }
                if(
                                (buttons[3].getText()=="X") &&
                                (buttons[4].getText()=="X") &&
                                (buttons[5].getText()=="X")
                                ) {
                        xWins(3,4,5);
                }
                if(
                                (buttons[6].getText()=="X") &&
                                (buttons[7].getText()=="X") &&
                                (buttons[8].getText()=="X")
                                ) {
                        xWins(6,7,8);
                }
                if(
                                (buttons[0].getText()=="X") &&
                                (buttons[3].getText()=="X") &&
                                (buttons[6].getText()=="X")
                        ) {
                    xWins(0,3,6);
                }
                if(
                        (buttons[1].getText()=="X") &&
                        (buttons[4].getText()=="X") &&
                        (buttons[7].getText()=="X")
                        ) {
                    xWins(1,4,7);
                }
                if(
                        (buttons[2].getText()=="X") &&
                        (buttons[5].getText()=="X") &&
                        (buttons[8].getText()=="X")
                        ) {
                    xWins(2,5,8);
                }
                if(
                        (buttons[0].getText()=="X") &&
                        (buttons[4].getText()=="X") &&
                        (buttons[8].getText()=="X")
                        ) {
                    xWins(0,4,8);
                }
                if(
                        (buttons[2].getText()=="X") &&
                        (buttons[4].getText()=="X") &&
                        (buttons[6].getText()=="X")
                        ) {
                    xWins(2,4,6);
                }
                //check O win conditions
                if(
                        (buttons[0].getText()=="O") &&
                        (buttons[1].getText()=="O") &&
                        (buttons[2].getText()=="O")
                        ) {
                    oWins(0,1,2);
                }
                if(
                        (buttons[3].getText()=="O") &&
                        (buttons[4].getText()=="O") &&
                        (buttons[5].getText()=="O")
                        ) {
                    oWins(3,4,5);
                }
```

```
    if(
            (buttons[6].getText()=="O") &&
            (buttons[7].getText()=="O") &&
            (buttons[8].getText()=="O")
            ) {
        oWins(6,7,8);
    }
    if(
            (buttons[0].getText()=="O") &&
            (buttons[3].getText()=="O") &&
            (buttons[6].getText()=="O")
            ) {
        oWins(0,3,6);
    }
    if(
            (buttons[1].getText()=="O") &&
            (buttons[4].getText()=="O") &&
            (buttons[7].getText()=="O")
            ) {
        oWins(1,4,7);
    }
    if(
            (buttons[2].getText()=="O") &&
            (buttons[5].getText()=="O") &&
            (buttons[8].getText()=="O")
            ) {
        oWins(2,5,8);
    }
    if(
            (buttons[0].getText()=="O") &&
            (buttons[4].getText()=="O") &&
            (buttons[8].getText()=="O")
            ) {
        oWins(0,4,8);
    }
    if(
            (buttons[2].getText()=="O") &&
            (buttons[4].getText()=="O") &&
            (buttons[6].getText()=="O")
            ) {
        oWins(2,4,6);
    }
}

public void xWins(int a,int b,int c) {
    buttons[a].setBackground(Color.GREEN);
```

```
		buttons[b].setBackground(Color.GREEN);
		buttons[c].setBackground(Color.GREEN);

		for(int i=0;i<9;i++) {
			buttons[i].setEnabled(false);
		}
		textfield.setText("X wins");
	}
	public void oWins(int a,int b,int c) {
		buttons[a].setBackground(Color.GREEN);
		buttons[b].setBackground(Color.GREEN);
		buttons[c].setBackground(Color.GREEN);

		for(int i=0;i<9;i++) {
			buttons[i].setEnabled(false);
		}
		textfield.setText("O wins");
	}
}
//***************************************
```

Output:



O wins

O	X	O
	O	X
O		X

Conclusion

In conclusion, while tic-tac-toe may appear straightforward and elementary, it holds more depth and significance than meets the eye. It serves as a foundational example in the field of game theory, illustrating concepts of perfect play, decision trees, and strategy. The game's solved nature, where a perfect strategy exists for both players, reveals that it will always end in a draw or a win for the first player. Nevertheless, the real-world dynamics of the game, including human error and psychological tactics, make tic-tac-toe an engaging and enjoyable pastime. Ultimately, tic-tac-toe showcases the interplay between simple rules and complex strategies, offering valuable insights into strategic thinking and decision-making.