Object Oriented Programming  
Tic Tac Toe 5 by 5  
  
  
This TicTacToe application is a JavaFX implementation of the classic TicTacToe game, but with a twist , it uses a 5x5 grid instead of the traditional 3x3 grid. Players take turns clicking on cells to place their marks 'X' or 'O'. The game dynamically checks for horizontal, vertical, and diagonal lines of five marks to determine a winner, and it also detects when the game ends in a draw if all cells are filled without a winner. The outcome of the game is displayed at the bottom of the grid for the players to see.

import javafx.application.Application;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.layout.GridPane;

import javafx.scene.text.Font;

import javafx.scene.text.FontWeight;

import javafx.scene.text.Text;

import javafx.stage.Stage;

public class TicTacTFivebyFive extends Application {

private char currentPlayer = 'X';

private Button[][] buttons = new Button[5][5];

private boolean gameOver = false;

private Text winnerText = new Text();

@Override

public void start(Stage primaryStage) {

GridPane gridPane = new GridPane();

gridPane.setAlignment(Pos.CENTER);

gridPane.setHgap(10);

gridPane.setVgap(10);

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

Button button = new Button("");

button.setPrefSize(80, 80);

button.setFont(Font.font("Arial", FontWeight.BOLD, 24));

int row = i, col = j;

button.setOnAction(e -> handleButtonClick(row, col, button));

buttons[i][j] = button;

gridPane.add(button, j, i);

}

}

winnerText.setFont(Font.font("Arial", FontWeight.BOLD, 20));

gridPane.add(winnerText, 0, 5, 5, 1);

Scene scene = new Scene(gridPane, 500, 600);

primaryStage.setScene(scene);

primaryStage.setTitle("Modified TicTacToe");

primaryStage.show();

}

private void handleButtonClick(int row, int col, Button button) {

if (!gameOver && button.getText().isEmpty()) {

button.setText(Character.toString(currentPlayer));

if (checkWin(row, col)) {

gameOver = true;

winnerText.setText(currentPlayer + " wins!");

} else if (checkDraw()) {

gameOver = true;

winnerText.setText("It's a draw!");

} else {

currentPlayer = (currentPlayer == 'X') ? 'O' : 'X';

}

}

}

private boolean checkWin(int row, int col) {

String mark = Character.toString(currentPlayer);

int count = 0;

for (int j = 0; j < 5; j++) {

if (buttons[row][j].getText().equals(mark)) {

count++;

if (count == 5) return true;

} else {

count = 0;

}

}

count = 0;

for (int i = 0; i < 5; i++) {

if (buttons[i][col].getText().equals(mark)) {

count++;

if (count == 5) return true;

} else {

count = 0;

}

}

count = 0;

for (int i = 0; i < 5; i++) {

if (buttons[i][i].getText().equals(mark)) {

count++;

if (count == 5) return true;

} else {

count = 0;

}

}

count = 0;

for (int i = 0; i < 5; i++) {

if (buttons[i][4 - i].getText().equals(mark)) {

count++;

if (count == 5) return true;

} else {

count = 0;

}

}

return false;

}

private boolean checkDraw() {

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

if (buttons[i][j].getText().isEmpty()) {

return false;

}

}

}

return true;

}

public static void main(String[] args) {

launch(args);

}

}