

Statement:

Design a tic tac toe game using java graphics.

Implementation:

Based on the description, it appears that i've created a simple Tic-Tac-Toe game using Java and object-oriented programming principles. The game consists of four classes: Game, Player, GameBoard, and TicTacToeGui. The TicTacToeGui class extends JFrame and implements ActionListener to provide a visual interface for the game using JFrame and JButton components. The constructor of TicTacToeGui class sets up the buttons with action event listeners to handle button presses. The Game class contains two arrays to store the button press records as 0 or 1, which likely represent the moves made by the two players in the game. The class also has two methods: pXwin() and pOwin(). The pXwin() method checks if player X has won the game by examining the stored button press records, and returns true if X has won, otherwise false. Similarly, the pOwin() method checks if player O has won the game and returns true if O has won, otherwise false. The Player class contains a nowPlayer variable that is initialized with 'X' and likely represents the current player in the game. The class also has a resetPlayer() method that can be used to reset the current player field. The GameBoard class contains an array btnUsed[] which likely stores information about whether a button has been pressed or not, using 1 for pressed and 0 for not pressed. This array can be used to keep track of the state of the game board during the game. Overall, it seems like you have implemented a basic Tic-Tac-Toe game with a graphical user interface using Java and object-oriented programming concepts. Players can make moves by clicking on buttons in the GUI, and the game logic is implemented in the Game and Player classes to determine if a player has won or not. The GameBoard class likely keeps track of the state of the game board during the game.

Source Code:

```
public class Game {
    int[] pXset = {0,0,0,0,0,0,0,0,0};
    int[] pOset = {0,0,0,0,0,0,0,0,0};

    public boolean pXWin() {
        // Horizontal checks
        if ((pXset[0] == 1 && pXset[1] == 1 && pXset[2] == 1)) {
            return true;
        }
        else if (pXset[3] == 1 && pXset[4] == 1 && pXset[5] == 1) {
            return true;
        }
        else if (pXset[6] == 1 && pXset[7] == 1 && pXset[8] == 1) {
            return true;
        }
        // Vertical checks
        else if (pXset[0] == 1 && pXset[3] == 1 && pXset[6] == 1) {
            return true;
        }
        else if (pXset[1] == 1 && pXset[4] == 1 && pXset[7] == 1) {
```

```

        return true;
    }
    else if (pXset[2] == 1 && pXset[5] == 1 && pXset[8] == 1) {
        return true;
    }
    // Diagonal checks
    else if (pXset[0] == 1 && pXset[4] == 1 && pXset[8] == 1) {
        return true;
    }
    else if (pXset[2] == 1 && pXset[4] == 1 && pXset[6] == 1) {
        return true;
    }
    return false;
}

```

```

public boolean pOWin() {
    // Horizontal checks
    if ((pOset[0] == 1 && pOset[1] == 1 && pOset[2] == 1)) {
        return true;
    }
    else if (pOset[3] == 1 && pOset[4] == 1 && pOset[5] == 1) {
        return true;
    }
    else if (pOset[6] == 1 && pOset[7] == 1 && pOset[8] == 1) {
        return true;
    }
    // Vertical checks
    else if (pOset[0] == 1 && pOset[3] == 1 && pOset[6] == 1) {
        return true;
    }
    else if (pOset[1] == 1 && pOset[4] == 1 && pOset[7] == 1) {
        return true;
    }
    else if (pOset[2] == 1 && pOset[5] == 1 && pOset[8] == 1) {
        return true;
    }
    // Diagonal checks
    else if (pOset[0] == 1 && pOset[4] == 1 && pOset[8] == 1) {
        return true;
    }
    else if (pOset[2] == 1 && pOset[4] == 1 && pOset[6] == 1) {
        return true;
    }
    return false;
}

```

```

    }

    void resetGame() {
        for(int i=0; i<9; i++){
            pXset[i]=0;
            pOset[i]=0;
        }
    }
}

public class GameBoard {
    int moves = 0;

    int[] btnUsed = {0,0,0,0,0,0,0,0,0};

    void resetGameBoard() {
        moves=0;
        for(int i=0; i<9; i++){
            btnUsed[i]=0;
        }
    }
}

import java.util.Arrays;

public class Player {

    public char nowPlayer = 'X';

    void resetPlayer(){
        nowPlayer = 'X';
    }
}

import java.awt.Dimension;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JOptionPane;

```

```

public class TicTacToeGUI extends JFrame implements ActionListener {

    Player player;
    GameBoard gameboard;
    public Game game;

    JButton b0, b1, b2, b3, b4, b5, b6, b7, b8;

    public TicTacToeGUI(Game game, GameBoard gameboard, Player player) {
        this.game = game;
        this.gameboard = gameboard;
        this.player = player;

        b0 = new JButton();
        b1 = new JButton();
        b2 = new JButton();
        b3 = new JButton();
        b4 = new JButton();
        b5 = new JButton();
        b6 = new JButton();
        b7 = new JButton();
        b8 = new JButton();

        setTitle("Tic Tac Toe");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 300);
        setLayout(new GridLayout(3, 3));

        b0.setPreferredSize(new Dimension(100, 100));
        b0.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                if (gameboard.btnUsed[0] == 0) {
                    if (player.nowPlayer == 'X') {
                        b0.setText("X");
                        gameboard.moves++;
                        game.pXset[0] = 1;
                        gameboard.btnUsed[0] = 1;
                        if (gameboard.moves >= 5 && game.pXWin()) {
                            playAgainConfirm('X');
                        }
                        player.nowPlayer = 'O';
                    } else {
                        b0.setText("O");
                        gameboard.moves++;
                    }
                }
            }
        });
    }
}

```

```

        game.pOset[0] = 1;
        gameboard.btnUsed[0] = 1;
        if (gameboard.moves >= 5 && game.pOWin()) {
            playAgainConfirm('O');
        }
        player.nowPlayer = 'X';
    }
} else {
    JOptionPane.showMessageDialog(b0, "Button is already used");
}
if (gameboard.moves == 9) {
    playAgainConfirm('D');
}
}
});
add(b0);

b1.setPreferredSize(new Dimension(100, 100));
b1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (gameboard.btnUsed[1] == 0) {
            if (player.nowPlayer == 'X') {
                b1.setText("X");
                gameboard.moves++;
                game.pXset[1] = 1;
                gameboard.btnUsed[1] = 1;
                if (gameboard.moves >= 5 && game.pXWin()) {
                    playAgainConfirm('X');
                }
                player.nowPlayer = 'O';
            } else {
                b1.setText("O");
                gameboard.moves++;
                game.pOset[1] = 1;
                gameboard.btnUsed[1] = 1;
                if (gameboard.moves >= 5 && game.pOWin()) {
                    playAgainConfirm('O');
                }
                player.nowPlayer = 'X';
            }
        } else {
            JOptionPane.showMessageDialog(b1, "Button is already used");
        }
    }
});
if (gameboard.moves == 9) {

```

```

        playAgainConfirm('D');
    }
}
});
add(b1);

b2.setPreferredSize(new Dimension(100, 100));
b2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (gameboard.btnUsed[2] == 0) {
            if (player.nowPlayer == 'X') {
                b2.setText("X");
                gameboard.moves++;
                game.pXset[2] = 1;
                gameboard.btnUsed[2] = 1;
                if (gameboard.moves >= 5 && game.pXWin()) {
                    playAgainConfirm('X');
                }
                player.nowPlayer = 'O';
            } else {
                b2.setText("O");
                gameboard.moves++;
                game.pOset[2] = 1;
                gameboard.btnUsed[2] = 1;
                if (gameboard.moves >= 5 && game.pOWin()) {
                    playAgainConfirm('O');
                }
                player.nowPlayer = 'X';
            }
        } else {
            JOptionPane.showMessageDialog(b2, "Button is already used");
        }
        if (gameboard.moves == 9) {
            playAgainConfirm('D');
        }
    }
});
add(b2);

b3.setPreferredSize(new Dimension(100, 100));
b3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (gameboard.btnUsed[3] == 0) {
            if (player.nowPlayer == 'X') {

```

```

        b3.setText("X");
        gameboard.moves++;
        game.pXset[3] = 1;
        gameboard.btnUsed[3] = 1;
        if (gameboard.moves >= 5 && game.pXWin()) {
            playAgainConfirm('X');
        }
        player.nowPlayer = 'O';
    } else {
        b3.setText("O");
        gameboard.moves++;
        game.pOset[3] = 1;
        gameboard.btnUsed[3] = 1;
        if (gameboard.moves >= 5 && game.pOWin()) {
            playAgainConfirm('O');
        }
        player.nowPlayer = 'X';
    }
} else {
    JOptionPane.showMessageDialog(b3, "Button is already used");
}
if (gameboard.moves == 9) {
    playAgainConfirm('D');
}
}
});
add(b3);

```

```

b4.setPreferredSize(new Dimension(100, 100));
b4.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (gameboard.btnUsed[4] == 0) {
            if (player.nowPlayer == 'X') {
                b4.setText("X");
                gameboard.moves++;
                game.pXset[4] = 1;
                gameboard.btnUsed[4] = 1;
                if (gameboard.moves >= 5 && game.pXWin()) {
                    playAgainConfirm('X');
                }
                player.nowPlayer = 'O';
            } else {
                b4.setText("O");
                gameboard.moves++;
            }
        }
    }
});

```

```

        game.pOset[4] = 1;
        gameboard.btnUsed[4] = 1;
        if (gameboard.moves >= 5 && game.pOWin()) {
            playAgainConfirm('O');
        }
        player.nowPlayer = 'X';
    }
} else {
    JOptionPane.showMessageDialog(b4, "Button is already used");
}
if (gameboard.moves == 9) {
    playAgainConfirm('D');
}
}
});
add(b4);

```

```

b5.setPreferredSize(new Dimension(100, 100));
b5.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (gameboard.btnUsed[5] == 0) {
            if (player.nowPlayer == 'X') {
                b5.setText("X");
                gameboard.moves++;
                game.pXset[5] = 1;
                gameboard.btnUsed[5] = 1;
                if (gameboard.moves >= 5 && game.pXWin()) {
                    playAgainConfirm('X');
                }
                player.nowPlayer = 'O';
            } else {
                b5.setText("O");
                gameboard.moves++;
                game.pOset[5] = 1;
                gameboard.btnUsed[5] = 1;
                if (gameboard.moves >= 5 && game.pOWin()) {
                    playAgainConfirm('O');
                }
                player.nowPlayer = 'X';
            }
        } else {
            JOptionPane.showMessageDialog(b5, "Button is already used");
        }
    }
});
if (gameboard.moves == 9) {

```



```

        playAgainConfirm('D');
    }
}
});
add(b5);

b6.setPreferredSize(new Dimension(100, 100));
b6.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (gameboard.btnUsed[6] == 0) {
            if (player.nowPlayer == 'X') {
                b6.setText("X");
                gameboard.moves++;
                game.pXset[6] = 1;
                gameboard.btnUsed[6] = 1;
                if (gameboard.moves >= 5 && game.pXWin()) {
                    playAgainConfirm('X');
                }
                player.nowPlayer = 'O';
            } else {
                b6.setText("O");
                gameboard.moves++;
                game.pOset[6] = 1;
                gameboard.btnUsed[6] = 1;
                if (gameboard.moves >= 5 && game.pOWin()) {
                    playAgainConfirm('O');
                }
                player.nowPlayer = 'X';
            }
        } else {
            JOptionPane.showMessageDialog(b6, "Button is already used");
        }
        if (gameboard.moves == 9) {
            playAgainConfirm('D');
        }
    }
});
add(b6);

b7.setPreferredSize(new Dimension(100, 100));
b7.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (gameboard.btnUsed[7] == 0) {
            if (player.nowPlayer == 'X') {

```

```

        b7.setText("X");
        gameboard.moves++;
        game.pXset[7] = 1;
        gameboard.btnUsed[7] = 1;
        if (gameboard.moves >= 5 && game.pXWin()) {
            playAgainConfirm('X');
        }
        player.nowPlayer = 'O';
    } else {
        b7.setText("O");
        gameboard.moves++;
        game.pOset[7] = 1;
        gameboard.btnUsed[7] = 1;
        if (gameboard.moves >= 5 && game.pOWin()) {
            playAgainConfirm('O');
        }
        player.nowPlayer = 'X';
    }
} else {
    JOptionPane.showMessageDialog(b7, "Button is already used");
}
if (gameboard.moves == 9) {
    playAgainConfirm('D');
}
}
});
add(b7);

```

```

b8.setPreferredSize(new Dimension(100, 100));
b8.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (gameboard.btnUsed[8] == 0) {
            if (player.nowPlayer == 'X') {
                b8.setText("X");
                gameboard.moves++;
                game.pXset[8] = 1;
                gameboard.btnUsed[8] = 1;
                if (gameboard.moves >= 5 && game.pXWin()) {
                    playAgainConfirm('X');
                }
                player.nowPlayer = 'O';
            } else {
                b8.setText("O");
                gameboard.moves++;
            }
        }
    }
});

```

```

        game.pOset[8] = 1;
        gameboard.btnUsed[8] = 1;
        if (gameboard.moves >= 5 && game.pOWin()) {
            playAgainConfirm('O');
        }
        player.nowPlayer = 'X';
    }
} else {
    JOptionPane.showMessageDialog(b8, "Button is already used");
}
if (gameboard.moves == 9) {
    playAgainConfirm('D');
}
}
});
add(b8);

setVisible(true);
}

public void playAgainConfirm(char x) {
    if (x == 'X') {
        String winMessage = "Player X wins the game!\n";
        int dialogResult = JOptionPane.showOptionDialog(null, winMessage+"Do you want to
Play Again?", "Confirmation", JOptionPane.YES_NO_OPTION,
JOptionPane.QUESTION_MESSAGE, null, null, null);
        if (dialogResult == JOptionPane.YES_OPTION) {
            resetGame();
        } else {
            System.exit(0);
        }
    } else if (x == 'O') {
        String winMessage = "Player O wins the game!\n";
        int dialogResult = JOptionPane.showOptionDialog(null, winMessage+"Do you want to
Play Again?", "Confirmation", JOptionPane.YES_NO_OPTION,
JOptionPane.QUESTION_MESSAGE, null, null, null);
        if (dialogResult == JOptionPane.YES_OPTION) {
            resetGame();
        } else {
            System.exit(0);
        }
    } else if (x == 'D') {
        String winMessage = "It's a DRAW !!\n";
    }
}

```

```

        int dialogResult = JOptionPane.showOptionDialog(null, winMessage+"Do you want to
Play Again?", "Confirmation", JOptionPane.YES_NO_OPTION,
JOptionPane.QUESTION_MESSAGE, null, null, null);
        if (dialogResult == JOptionPane.YES_OPTION) {
            resetGame();
        } else {
            System.exit(0);
        }
    }
}

```

```

public void resetGame() {
    gameboard.resetGameBoard();
    game.resetGame();
    player.resetPlayer();
    blankBtns();
}

```

```

public void blankBtns() {
    b0.setText("");
    b1.setText("");
    b2.setText("");
    b3.setText("");
    b4.setText("");
    b5.setText("");
    b6.setText("");
    b7.setText("");
    b8.setText("");
}

```

```

@Override
public void actionPerformed(ActionEvent e) {
    throw new UnsupportedOperationException("Not supported yet.");
}

```

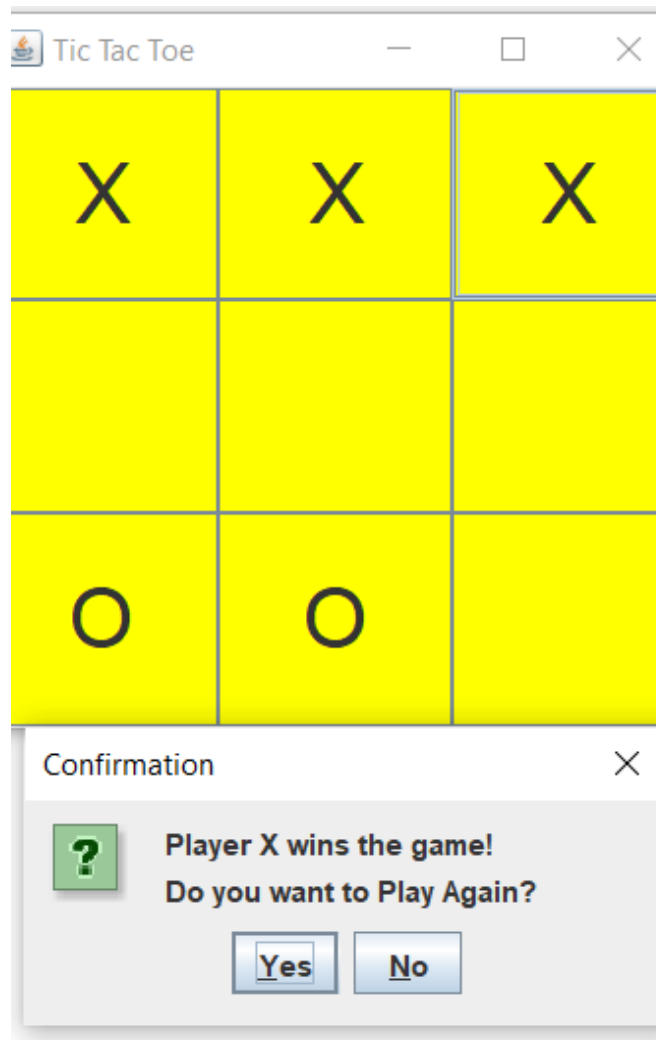
```

public static void main(String[] args) {
    Game game = new Game();
    GameBoard gameboard = new GameBoard();
    Player player = new Player();
    new TicTacToeGUI(game, gameboard, player);
}
}

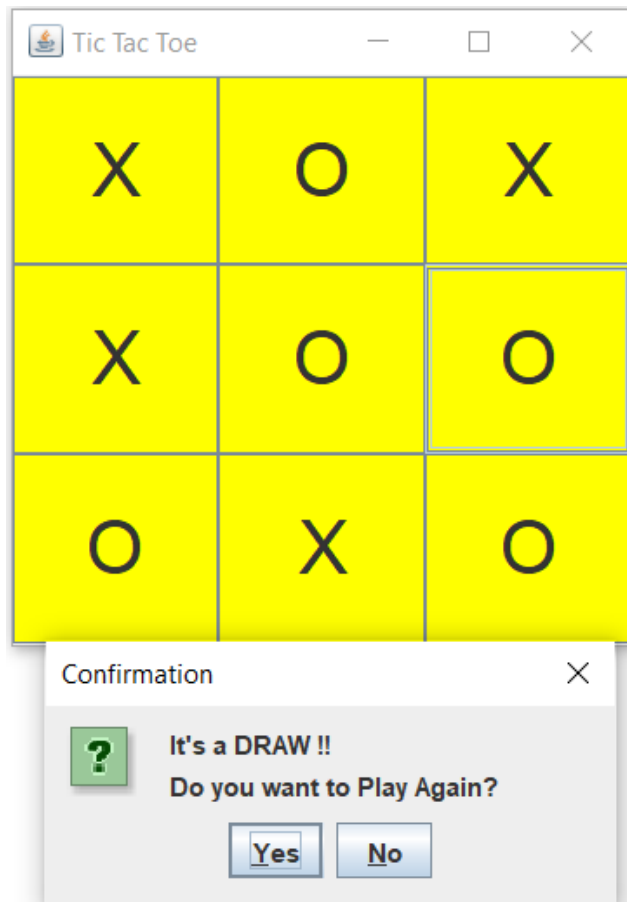
```

Output:

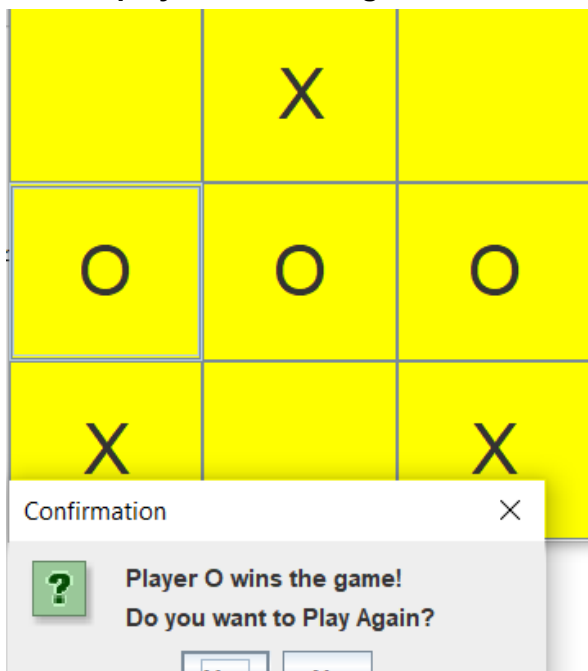
->When player x won the game:



->When its a draw:



->When player O won the game::



Discussion:

From the code, it appears that i've used Java and the Swing library to create a graphical user interface (GUI) for the game. The game consists of four classes: Game, Player, GameBoard, and TicTacToeGui. The TicTacToeGui class extends JFrame and implements ActionListener, which means it inherits from the JFrame class (a window container) and also implements the ActionListener interface to handle button click events. The buttons are created using JButton components, and each button has an ActionListener registered in the constructor, which allows them to respond to user clicks. The Game class contains two arrays to store the button press records as 0 or 1, which presumably represent the moves made by Player X and Player O on the game board. The pXwin() and pOwin() methods are used to determine if either player X or player O has won the game based on the current button press records. These methods likely implement the game logic and check for winning conditions, such as three buttons in a row, column, or diagonal having the same value (0 or 1). The Player class contains a nowPlayer variable, which is initialized with 'X', presumably representing the current player who is making a move. The resetPlayer() method is used to reset the field, which likely involves resetting the nowPlayer variable to its initial value. The GameBoard class contains an array btnUsed[], which stores 1 if a button is pressed, and 0 otherwise. This array likely keeps track of which buttons on the game board have been used (pressed) during the game. Overall, your game seems to have a good structure with separate classes for different functionalities, such as GUI, game logic, and player management. It's important to ensure that the classes are well-designed and follow object-oriented programming principles, such as encapsulation, inheritance, and polymorphism, to make the code maintainable and extensible. Additionally, make sure to handle different edge cases, such as invalid moves and game over conditions, to create a robust and enjoyable game experience for the players.