Royal Crow Company

CrowApp

Miguel da Silva & Vitor Adriano

ESCOLHA DO PROJETO

EXERCITAR HABILIDADES DE PROGRAMAÇÃO

FONTE DE ENTRETERIMENTO

HABILIDADES SOCIAIS

TOOLBOX

Canva
YouTube

Java Swing, Java AWT, Java Collections

Skills

Hard Skills

- Orientação a eventos
- Biblioteca Swing
- Apache Netbeans

Soft Skills

- Divisão de tarefas
- Comunicação
- Planejamento da ideia

Barreiras

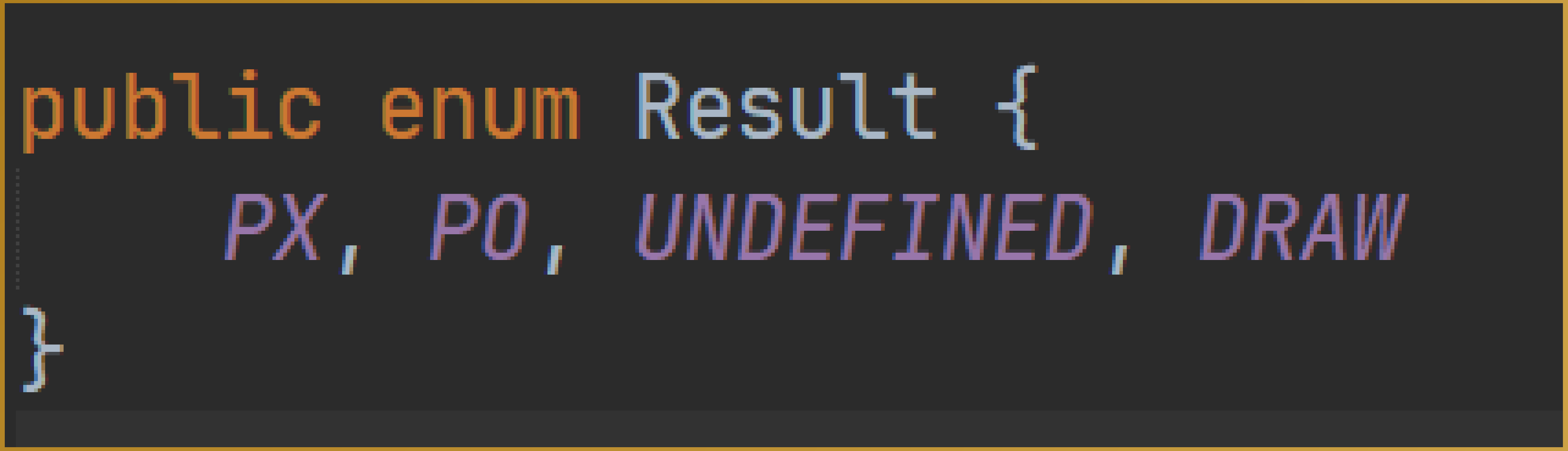
01 Design gráfico

02 Ansiedade

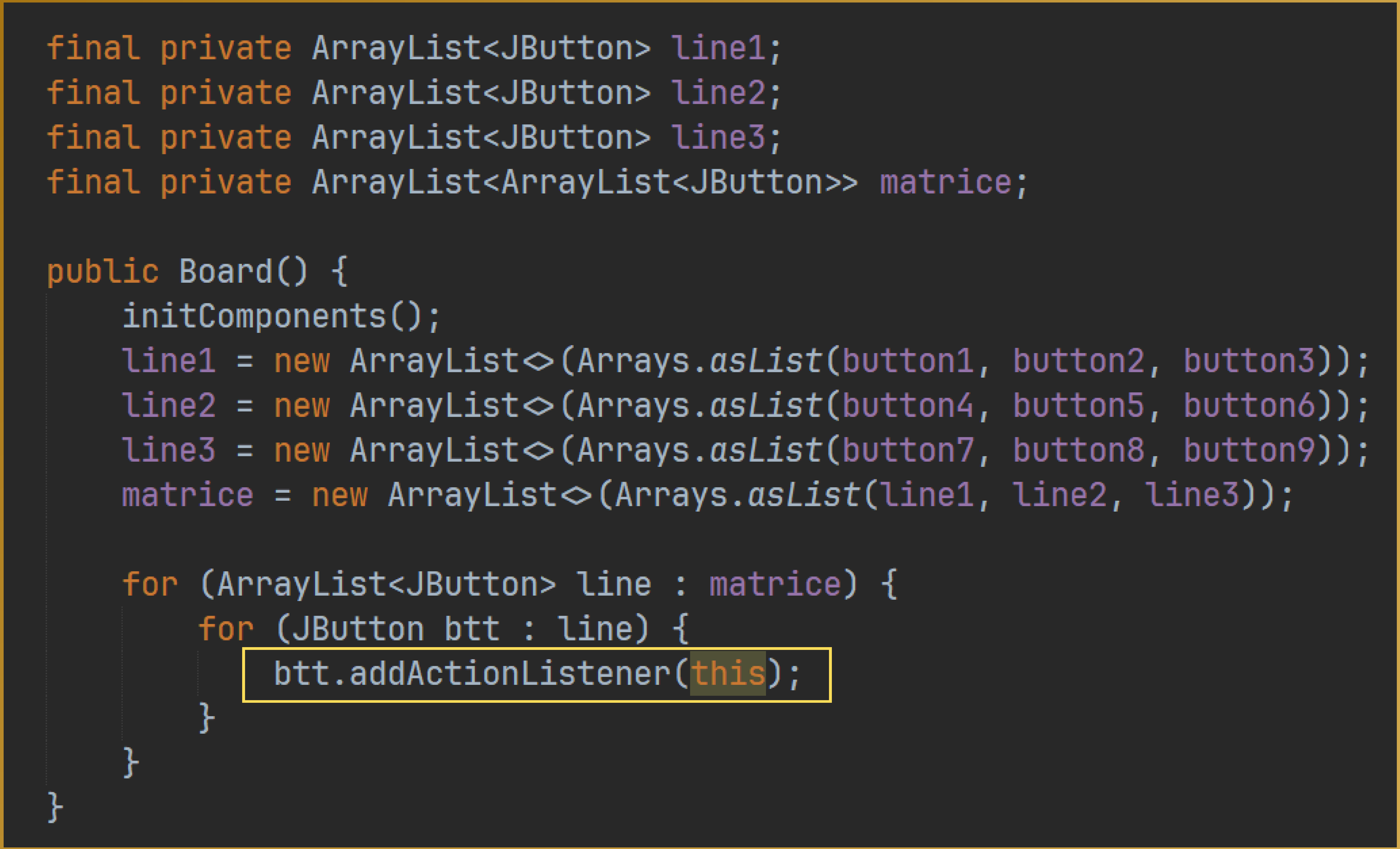
03 Criatividade

Diagrama de classes

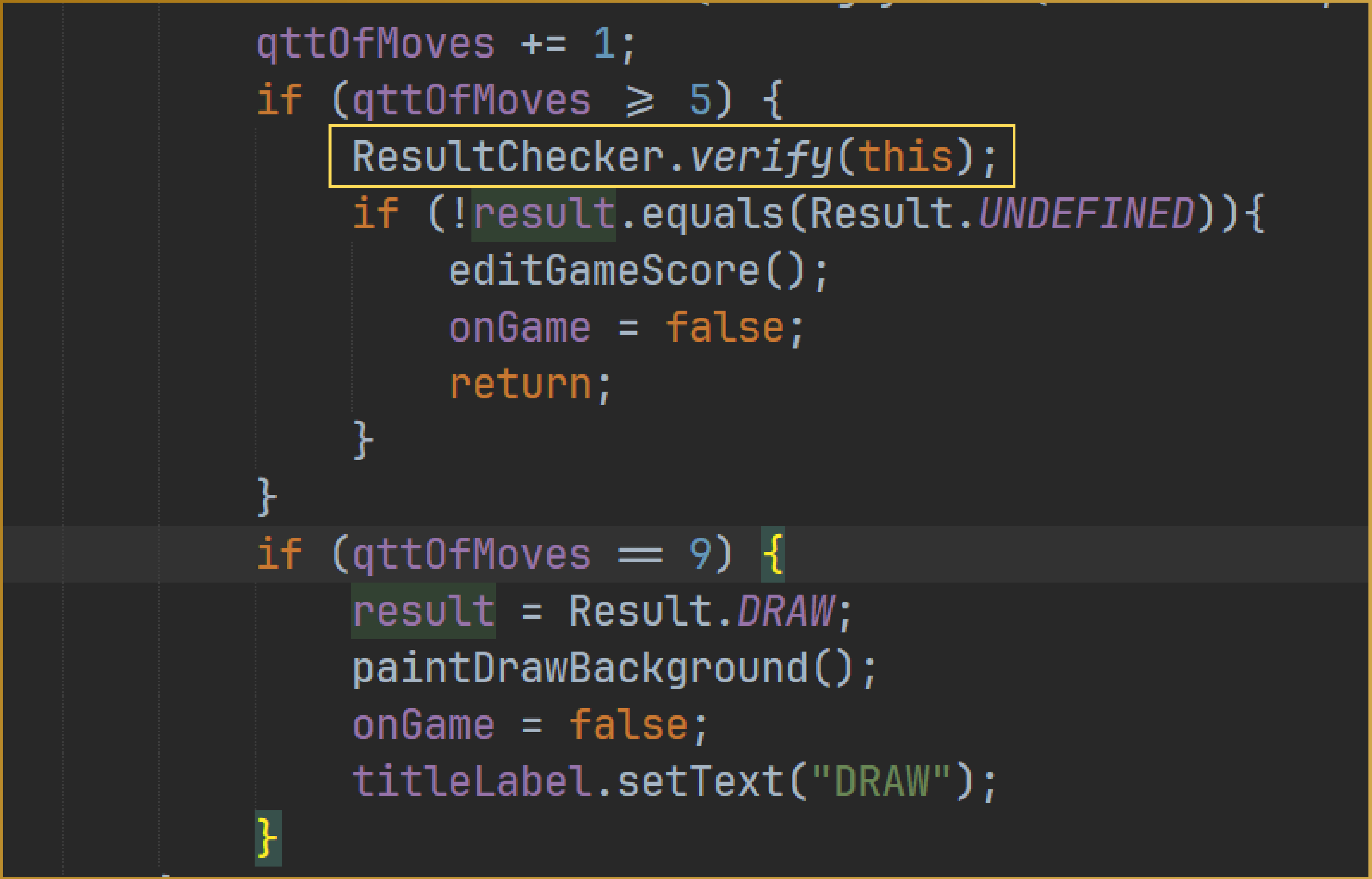
Tic Tac Toe



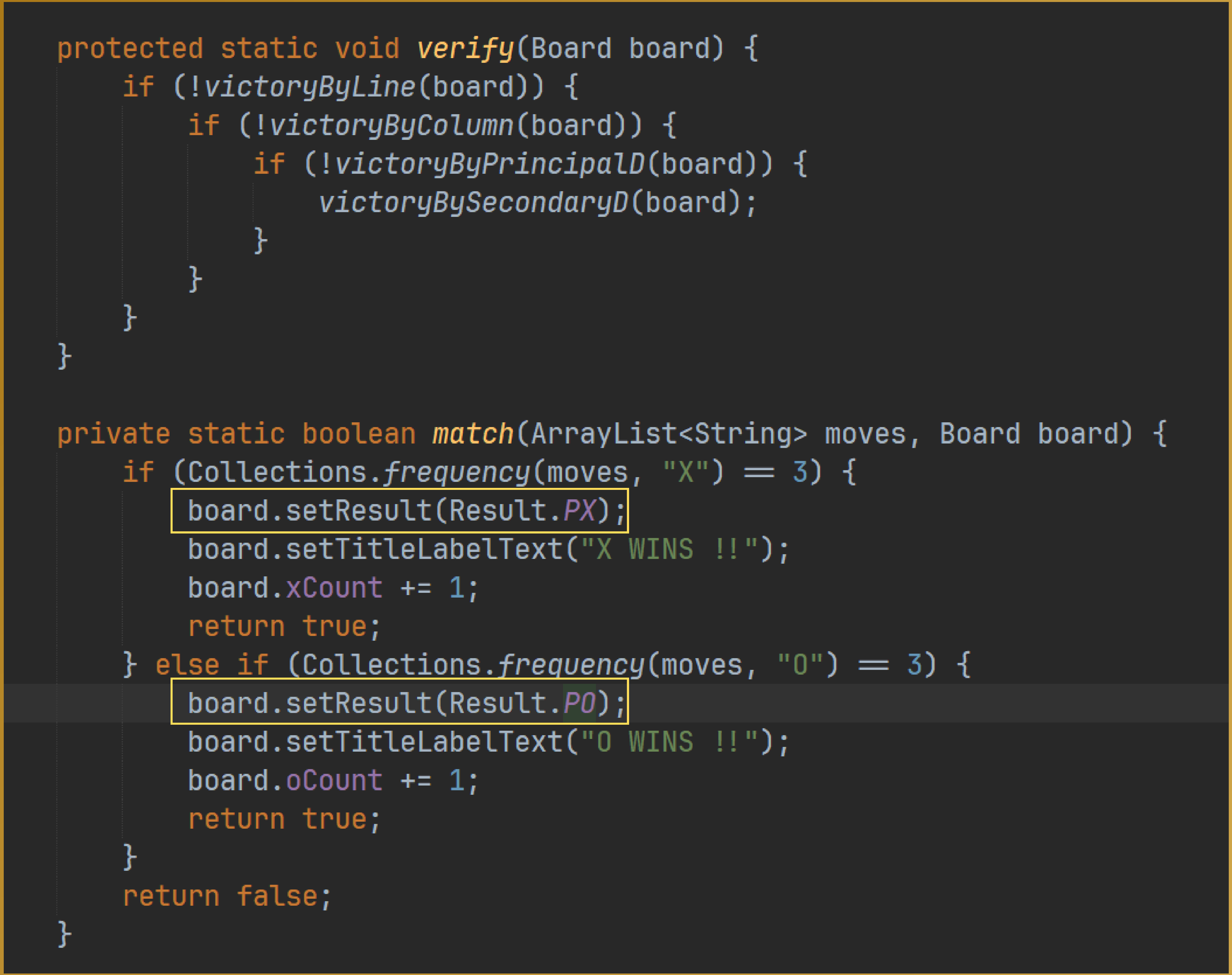
```java
public enum Result {
    PX, PO, UNDEFINED, DRAW
}
```



```java
final private ArrayList<JButton> line1;
final private ArrayList<JButton> line2;
final private ArrayList<JButton> line3;
final private ArrayList<ArrayList<JButton>> matrice;

public Board() {
    initComponents();
    line1 = new ArrayList<>(Arrays.asList(button1, button2, button3));
    line2 = new ArrayList<>(Arrays.asList(button4, button5, button6));
    line3 = new ArrayList<>(Arrays.asList(button7, button8, button9));
    matrice = new ArrayList<>(Arrays.asList(line1, line2, line3));

    for (ArrayList<JButton> line : matrice) {
        for (JButton btt : line) {
            btt.addActionListener(this);
        }
    }
}
```



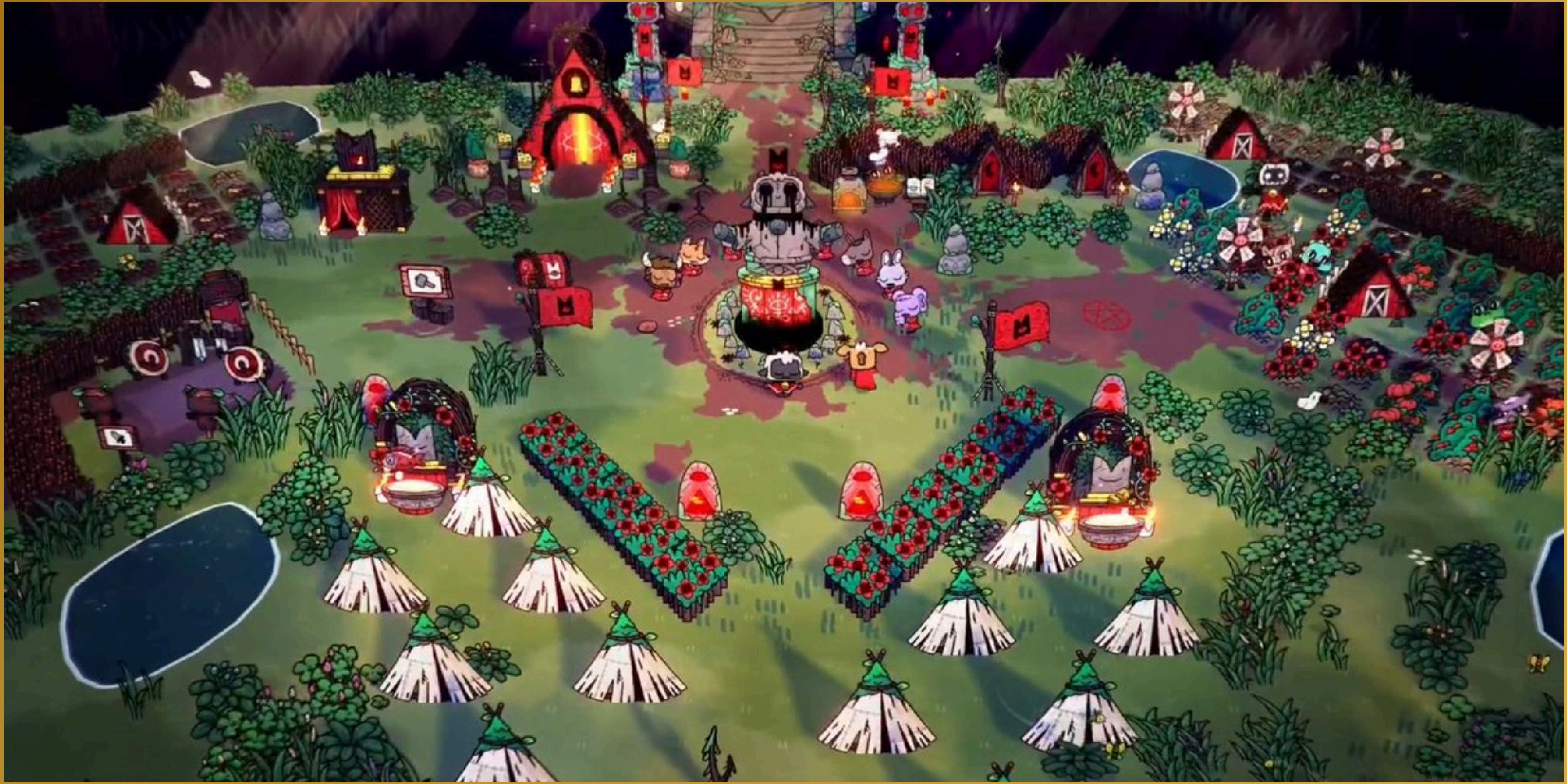
```java
qttOfMoves += 1;
if (qttOfMoves >= 5) {
    ResultChecker.verify(this);
    if (!result.equals(Result.UNDEFINED)){
        editGameScore();
        onGame = false;
        return;
    }
}
if (qttOfMoves == 9) {
    result = Result.DRAW;
    paintDrawBackground();
    onGame = false;
    titleLabel.setText("DRAW");
}
```



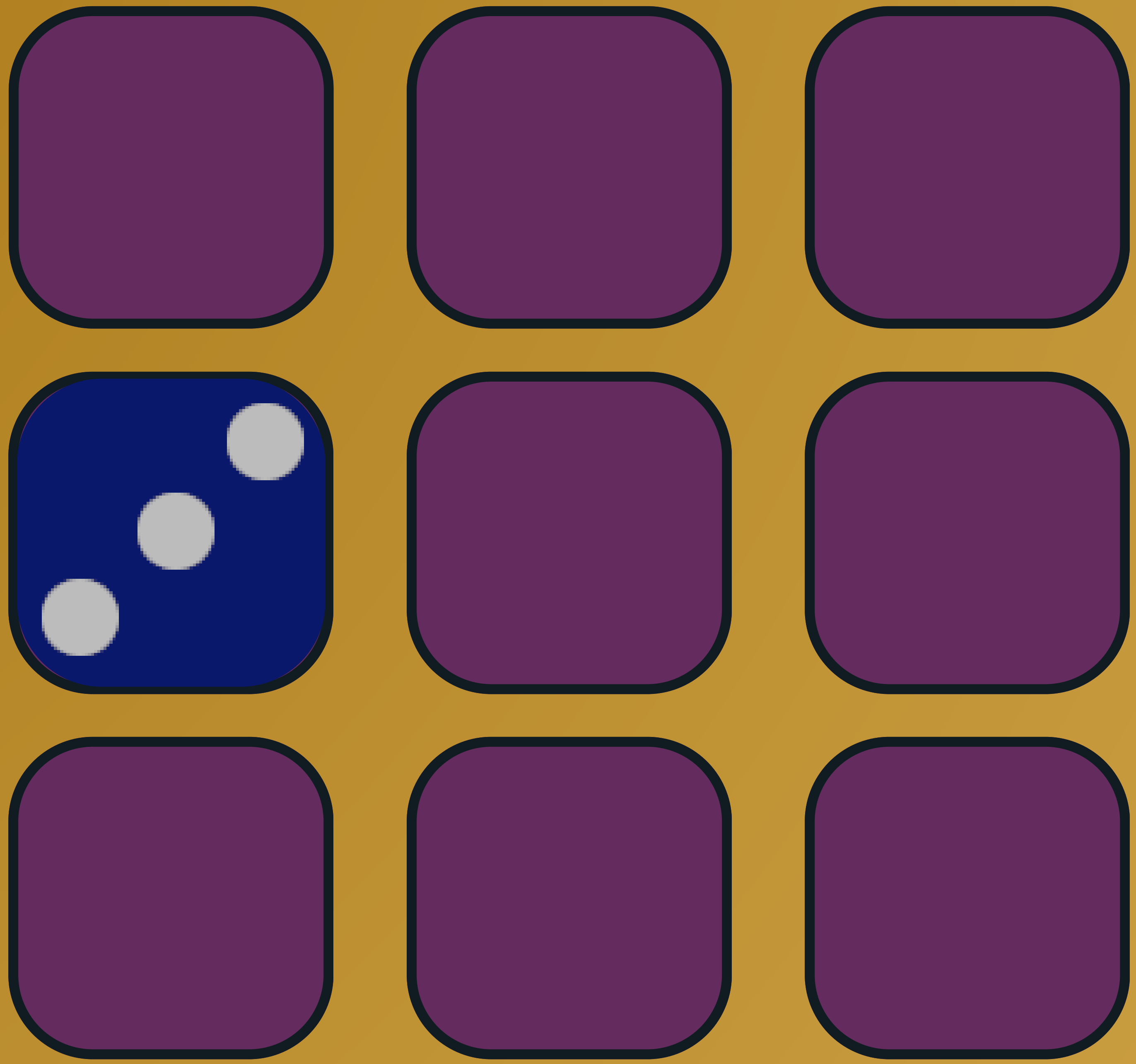
```java
protected static void verify(Board board) {
    if (!victoryByLine(board)) {
        if (!victoryByColumn(board)) {
            if (!victoryByPrincipalD(board)) {
                victoryBySecondaryD(board);
            }
        }
    }
}

private static boolean match(ArrayList<String> moves, Board board) {
    if (Collections.frequency(moves, "X") == 3) {
        board.setResult(Result.PX);
        board.setTitleLabelText("X WINS !!");
        board.xCount += 1;
        return true;
    } else if (Collections.frequency(moves, "O") == 3) {
        board.setResult(Result.PO);
        board.setTitleLabelText("O WINS !!");
        board.oCount += 1;
        return true;
    }
    return false;
}
```

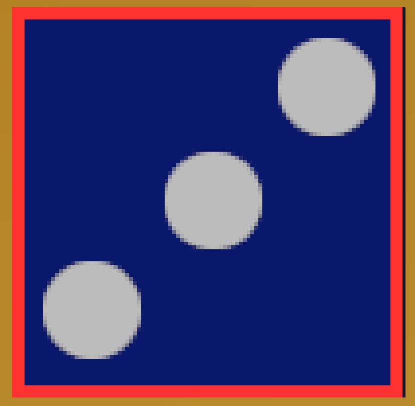
KnuckleBones

Cult of the Lamb

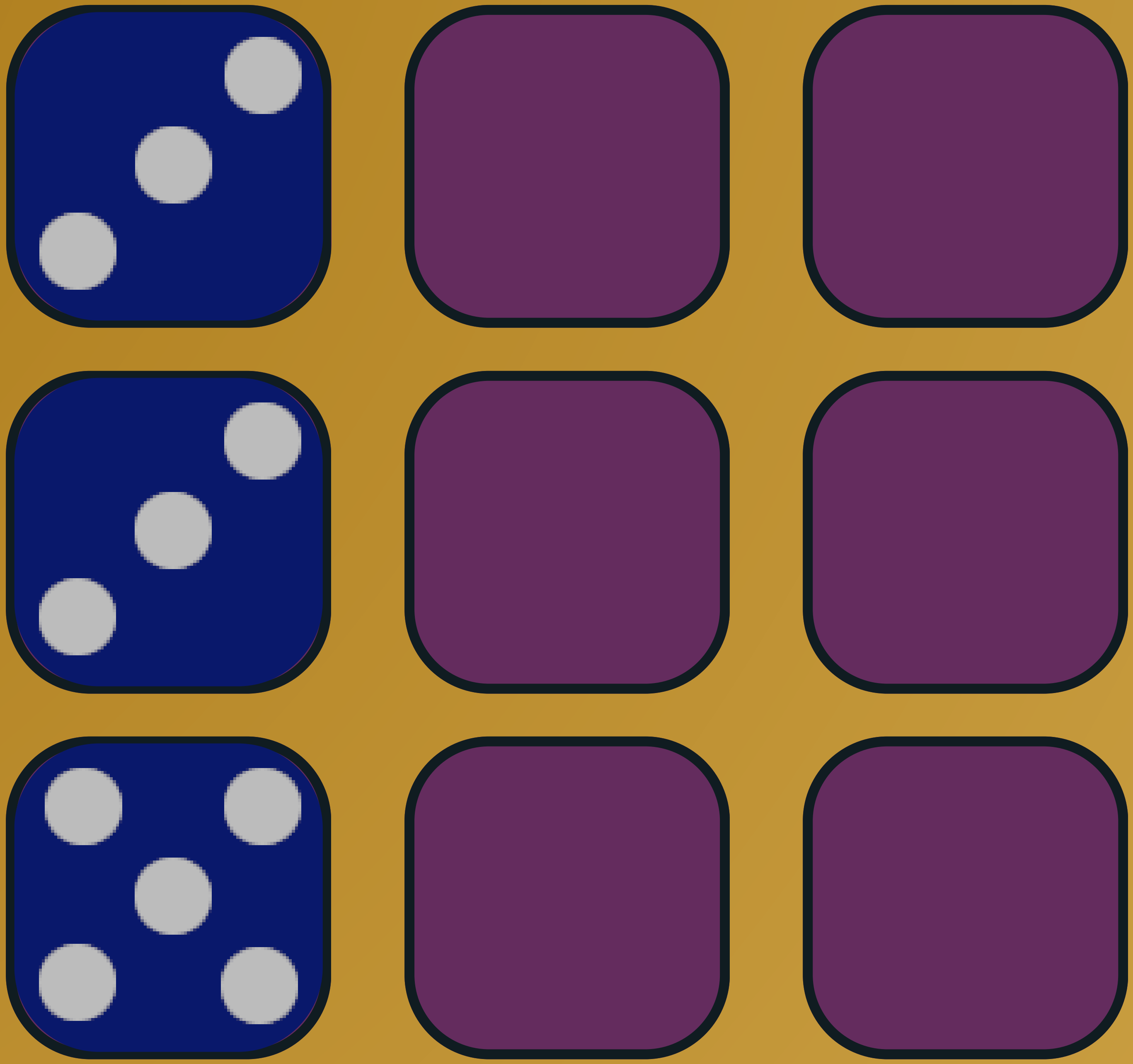


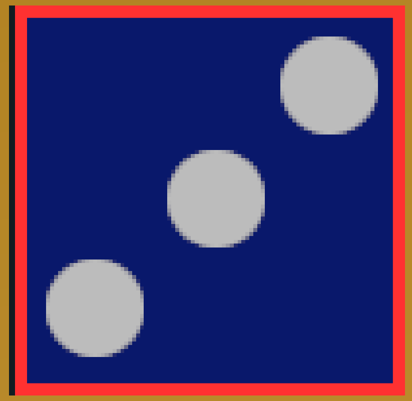
3

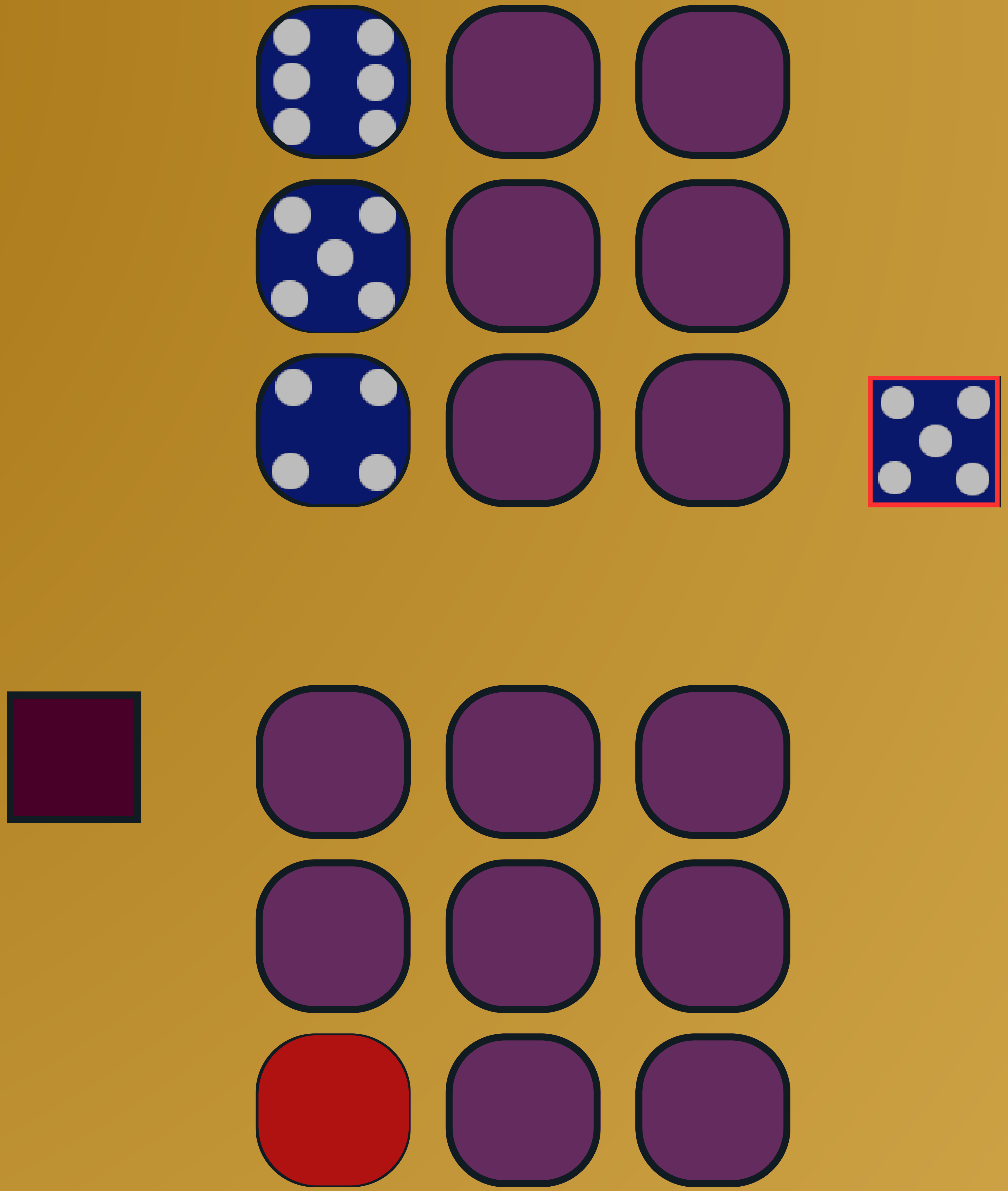


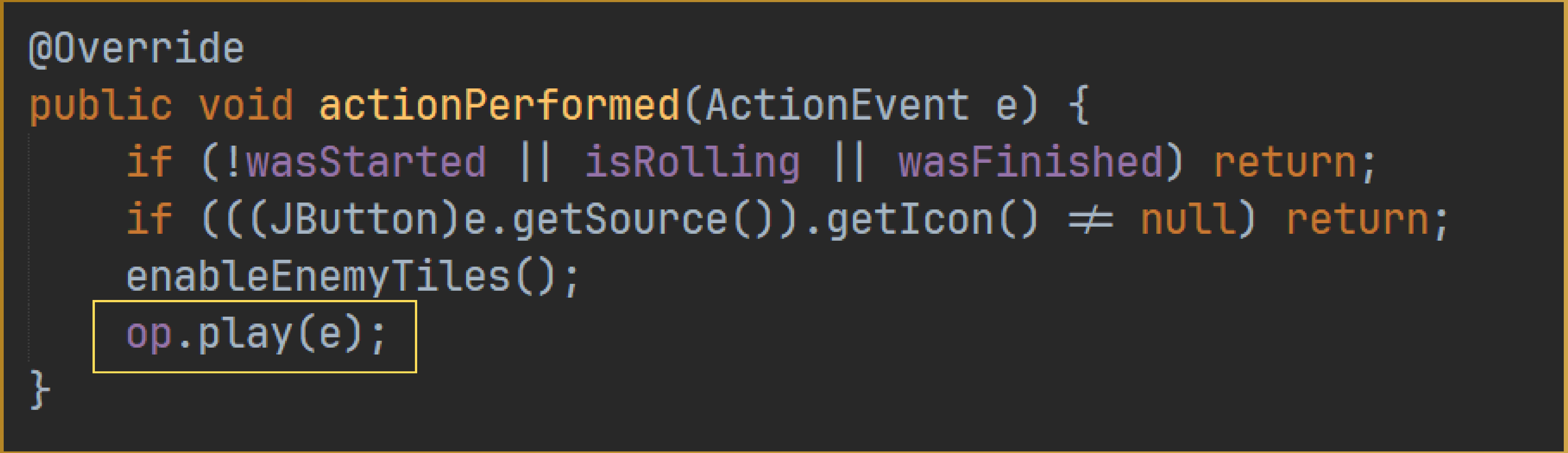


14

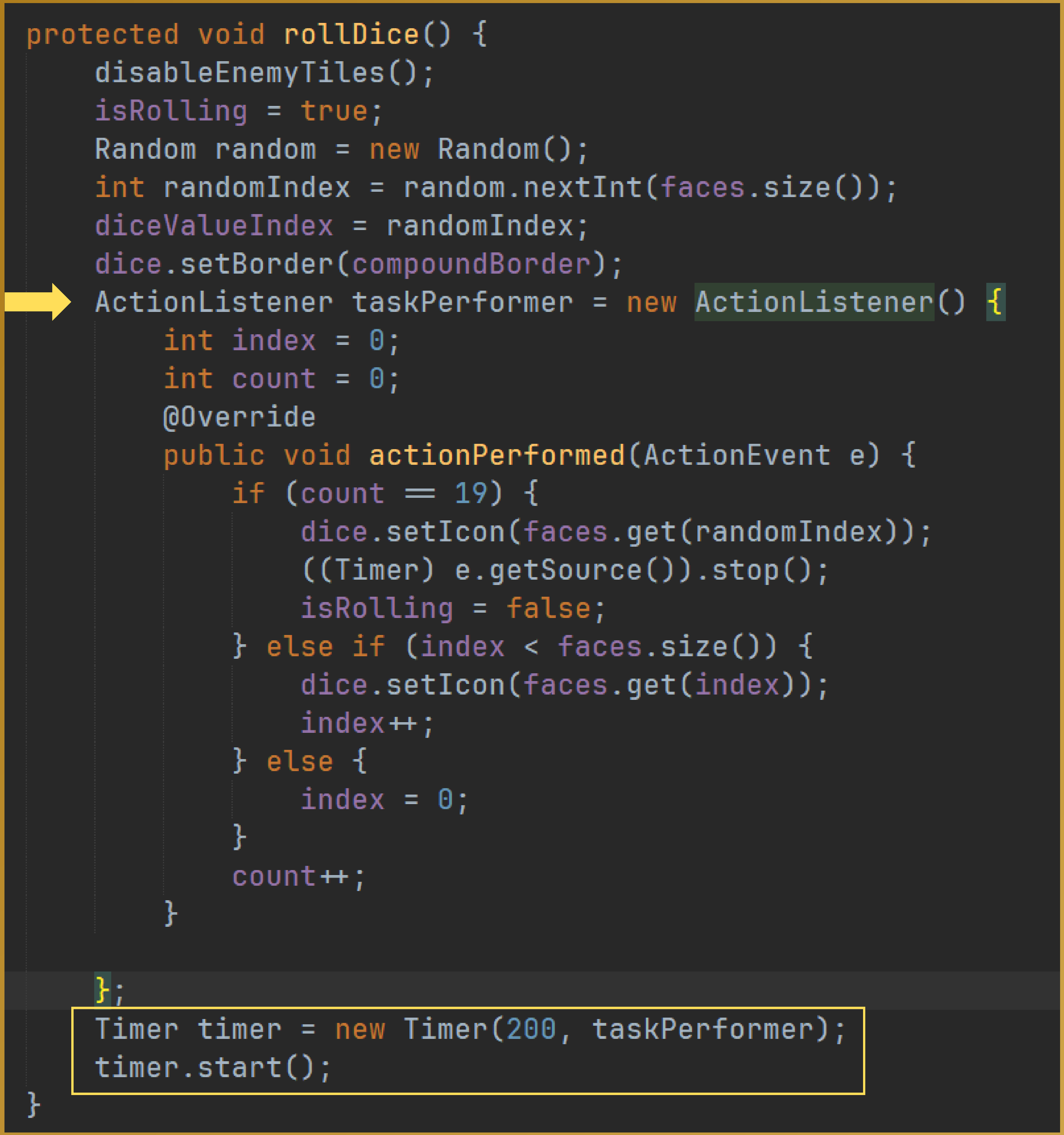




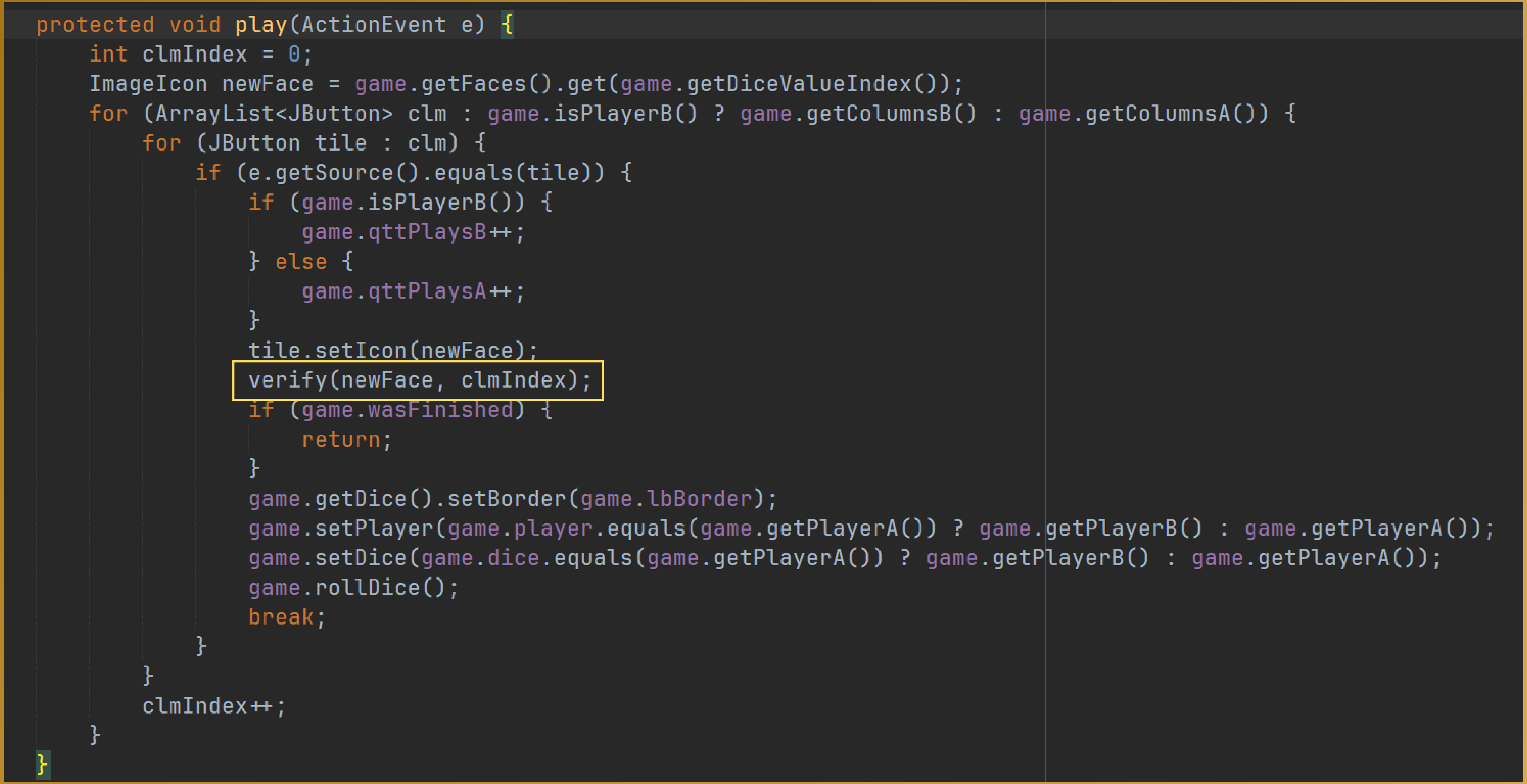




```
@Override
public void actionPerformed(ActionEvent e) {
    if (!wasStarted || isRolling || wasFinished) return;
    if (((JButton)e.getSource()).getIcon() ≠ null) return;
    enableEnemyTiles();
    op.play(e);
}
```



```java
protected void rollDice() {
    disableEnemyTiles();
    isRolling = true;
    Random random = new Random();
    int randomIndex = random.nextInt(faces.size());
    diceValueIndex = randomIndex;
    dice.setBorder(compoundBorder);
    ActionListener taskPerformer = new ActionListener() {
        int index = 0;
        int count = 0;
        @Override
        public void actionPerformed(ActionEvent e) {
            if (count == 19) {
                dice.setIcon(faces.get(randomIndex));
                ((Timer) e.getSource()).stop();
                isRolling = false;
            } else if (index < faces.size()) {
                dice.setIcon(faces.get(index));
                index++;
            } else {
                index = 0;
            }
            count++;
        }

    };
    Timer timer = new Timer(200, taskPerformer);
    timer.start();
}
```



```java
protected void play(ActionEvent e) {
    int clmIndex = 0;
    ImageIcon newFace = game.getFaces().get(game.getDiceValueIndex());
    for (ArrayList<JButton> clm : game.isPlayerB() ? game.getColumnsB() : game.getColumnsA()) {
        for (JButton tile : clm) {
            if (e.getSource().equals(tile)) {
                if (game.isPlayerB()) {
                    game.qttPlaysB++;
                } else {
                    game.qttPlaysA++;
                }
                tile.setIcon(newFace);
                verify(newFace, clmIndex);
                if (game.wasFinished) {
                    return;
                }
                game.getDice().setBorder(game.lbBorder);
                game.setPlayer(game.player.equals(game.getPlayerA()) ? game.getPlayerB() : game.getPlayerA());
                game.setDice(game.dice.equals(game.getPlayerA()) ? game.getPlayerB() : game.getPlayerA());
                game.rollDice();
                break;
            }
        }
        clmIndex++;
    }
}
```

CONCLUSÕES

- Um projeto divertido
- Melhorar a estética do app
- Estudar ferramentas de design gráfico
- Implementar redefinição de dados

Obrigado pela atenção