

NAMA : IRVAN BUDI ZAELANI

NIM : 2222105099

KELAS : 2TI03

DOSEN : NOVAN ZULKARNAIN, S.T., M.Kom.

MATA KULIAH : PEMOGRAMAN BERORIENTASI OBJEK

Penjelasan Lengkap Aplikasi "Mario: Whac A Mole"

1. Deskripsi Aplikasi

"Mario: Whac A Mole" adalah sebuah game komputer yang terinspirasi dari permainan arcade klasik "Whac-A-Mole". Dalam game ini, pemain harus memukul karakter yang muncul dari beberapa lubang dalam grid. Karakter-karakter ini muncul secara acak dan pemain harus bereaksi dengan cepat untuk mendapatkan skor tinggi.

2. Aturan Permainan

Permainan ini memiliki beberapa aturan dasar yang harus diikuti oleh pemain:

- Tujuan Utama : Mencapai skor tertinggi dengan memukul karakter mole (tikus tanah) yang muncul dari lubang.
- Karakter dalam Permainan :
 - Mole : Karakter yang harus dipukul oleh pemain. Memukul mole akan menambah skor pemain.
 - Piranha Plant : Karakter yang tidak boleh dipukul. Memukul Piranha Plant dapat mengurangi skor atau menyebabkan penalti tertentu dalam permainan.

3. Mekanisme Permainan

Permainan ini memiliki antarmuka pengguna yang sederhana namun fungsional, dan menggunakan mekanisme interaksi yang intuitif.

- Antarmuka Pengguna :

- Grid : Terdapat grid yang terdiri dari beberapa lubang. Karakter mole dan Piranha Plant muncul dari lubang-lubang ini.

- Tampilan Skor : Skor pemain ditampilkan di bagian atas layar. Gambar menunjukkan skor saat ini adalah 50.

- Interaksi Pemain :

- Pemain menggunakan alat input seperti mouse atau layar sentuh untuk memukul karakter mole yang muncul.

- Pemain harus berhati-hati agar tidak memukul Piranha Plant, karena ini dapat mengurangi skor atau memberikan penalti.

4. Sistem Skor

Skor dalam permainan ini diakumulasikan berdasarkan jumlah karakter mole yang berhasil dipukul oleh pemain. Setiap mole yang dipukul menambah sejumlah poin ke skor total. Sebaliknya, memukul Piranha Plant dapat mengurangi skor atau memberikan penalti lainnya, tergantung pada aturan spesifik yang diterapkan dalam permainan.

5. Tantangan dan Keseruan

Permainan ini menantang karena menguji kecepatan reaksi dan ketangkasan pemain. Karakter mole dan Piranha Plant muncul secara acak dan cepat, sehingga pemain harus selalu waspada dan cepat dalam bereaksi. Keseruan permainan ini terletak pada:

- Tantangan Kecepatan : Pemain harus cepat dalam mengenali dan memukul mole yang muncul.
- Tantangan Ketepatan : Pemain harus memastikan hanya memukul mole dan menghindari Piranha Plant.

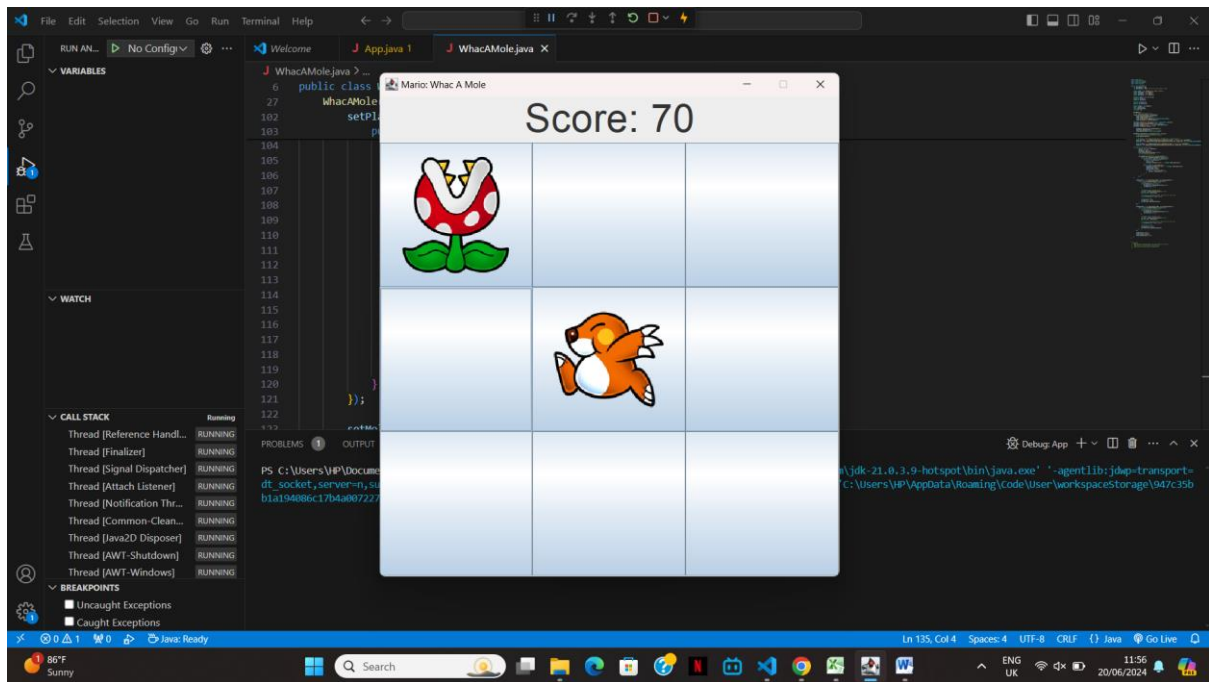
6. Tujuan dan Manfaat

Selain sebagai hiburan, permainan ini juga memiliki beberapa manfaat:

- Meningkatkan Refleks : Membantu pemain meningkatkan refleks dan koordinasi tangan-mata.
- Mengasah Ketangkasan : Meningkatkan kemampuan untuk bereaksi cepat dan tepat.
- Penghilang Stres : Menyediakan hiburan yang menyenangkan dan bisa menjadi penghilang stres.

Kesimpulan

"Mario: Whac A Mole" adalah sebuah permainan sederhana namun menantang yang menguji kecepatan dan ketangkasan pemain. Dengan mekanisme yang mudah dipahami dan antarmuka yang intuitif, permainan ini menawarkan hiburan yang seru dan dapat menjadi alat yang baik untuk meningkatkan refleks dan koordinasi mata-tangan pemain. Skor yang terus diperbarui memberikan motivasi bagi pemain untuk terus mencoba dan memperbaiki hasil mereka, menjadikan permainan ini menyenangkan dan bermanfaat.



App.Java

```
public class App {
    public static void main(String[] args) throws Exception {
        WhacAMole whacAMole = new WhacAMole();
    }
}
```

WhacAMole.Java

```
import java.awt.*;
import java.awt.event.*;
import java.util.Random;
import javax.swing.*;

public class WhacAMole {
    int boardWidth = 600;
    int boardHeight = 650; //50 for the text panel on top

    JFrame frame = new JFrame("Mario: Whac A Mole");
    JLabel textLabel = new JLabel();
    JPanel textPanel = new JPanel();
    JPanel boardPanel = new JPanel();

    JButton[] board = new JButton[9];
    ImageIcon moleIcon;
    ImageIcon plantIcon;

    JButton currMoleTile;
    JButton currPlantTile;
```

```

Random random = new Random();
Timer setMoleTimer;
Timer setPlantTimer;
int score = 0;

WhacAMole() {
    // frame.setVisible(true);
    frame.setSize(boardWidth, boardHeight);
    frame.setLocationRelativeTo(null);
    frame.setResizable(false);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setLayout(new BorderLayout());

    textLabel.setFont(new Font("Arial", Font.PLAIN, 50));
    textLabel.setHorizontalAlignment(JLabel.CENTER);
    textLabel.setText("Score: " + Integer.toString(score));
    textLabel.setOpaque(true);

    textPanel.setLayout(new BorderLayout());
    textPanel.add(textLabel);
    frame.add(textPanel, BorderLayout.NORTH);

    boardPanel.setLayout(new GridLayout(3, 3));
    // boardPanel.setBackground(Color.black);
    frame.add(boardPanel);

    // plantIcon = new ImageIcon(getClass().getResource("./piranha.png"));
    Image plantImg = new
ImageIcon(getClass().getResource("./piranha.png")).getImage();
    plantIcon = new ImageIcon(plantImg.getScaledInstance(150, 150,
java.awt.Image.SCALE_SMOOTH));

    Image moleImg = new
ImageIcon(getClass().getResource("./monty.png")).getImage();
    moleIcon = new ImageIcon(moleImg.getScaledInstance(150, 150,
java.awt.Image.SCALE_SMOOTH));

    for (int i = 0; i < 9; i++) {
        JButton tile = new JButton();
        board[i] = tile;
        boardPanel.add(tile);
        tile.setFocusable(false);
        // tile.setIcon(plantIcon);

        tile.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                JButton tile = (JButton) e.getSource();

```

```

        if (tile == currMoleTile) {
            score += 10;
            textLabel.setText("Score: " +
Integer.toString(score));
        }
        else if (tile == currPlantTile) {
            textLabel.setText("Game Over: " +
Integer.toString(score));
            setMoleTimer.stop();
            setPlantTimer.stop();
            for (int i = 0; i < 9; i++) {
                board[i].setEnabled(false);
            }
        }
    }
});
}

setMoleTimer = new Timer(1000, new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //remove icon from current tile
        if (currMoleTile != null) {
            currMoleTile.setIcon(null);
            currMoleTile = null;
        }

        //randomly select another tile
        int num = random.nextInt(9); //0-8
        JButton tile = board[num];

        //if tile is occupied by plant, skip tile for this turn
        if (currPlantTile == tile) return;

        //set tile to mole
        currMoleTile = tile;
        currMoleTile.setIcon(moleIcon);
    }
});

setPlantTimer = new Timer(1500, new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //remove icon from current tile
        if (currPlantTile != null) {
            currPlantTile.setIcon(null);
            currPlantTile = null;
        }

        //randomly select another tile

```

```

        int num = random.nextInt(9); //0-8
        JButton tile = board[num];

        //if tile is occupied by mole, skip tile for this turn
        if (currMoleTile == tile) return;

        //set tile to mole
        currPlantTile = tile;
        currPlantTile.setIcon(plantIcon);
    }
});

setMoleTimer.start();
setPlantTimer.start();
frame.setVisible(true);
}
}

/*
 * Homework
 * - Add Multiple Piranha plants and store them in an Array
 * - Add Button on the bottom to restart game
 * - Keep track of high score and display it
 */

```