Government Polytechnic, Pune

An Autonomous Institute
of Government of
Maharashtra

Report on

Micro Project

Puzzle Game

Submitted By

Minal Chhatre(1906016)
Priyanka Balivada(1906008)
Aishwarya Auti(1906003)

Under the guidance of

smt. Saraswati Panchakshari

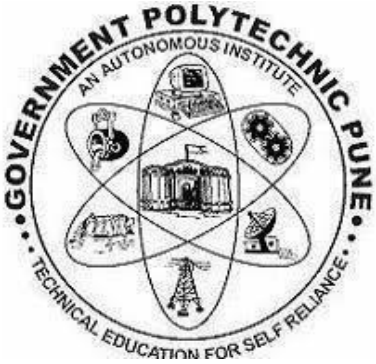
G3 Division

Department of Computer Engineering

Government Polytechnic, Pune

Government Polytechnic, Pune -16

(An autonomous institute of Government of Maharashtra)



Certificate

This is to certify that **Minal Chhatre, Priyanka Balivada, Aishwarya Auti** of class Third Year (2021-2022) have successfully completed project demonstration on **Snake Game** under the guidance of **Mrs. Saraswati Panchakshari**Mam in partial fulfillment of the requirement for the award of diploma in computer engineering from Government Polytechnic, Pune.

Mrs. Saraswati	**Prof. S.B.Nikam**	**Dr. V.S. Bandal**
(Course Guide)	(Head Of Department)	(Principal

Abstract

Java applets are small applications written in the Java programming language, or another programming language that compiles to Java bytecode, and delivered to users in the form of Java bytecode. The user launched the Java applet from a web page, and the applet was then executed within a Java virtual machine (JVM) in a process separate from the web browser itself. A Java applet could appear in a frame of the web page, a new application window, Sun's AppletViewer, or a stand-alone tool for testing applets. Java applets were usually written in Java, but other languages such as Jython, JRuby, Pascal,[11] Scala, NetRexx, or Eiffel (via SmartEiffel) could be used as well.

Acknowledgement

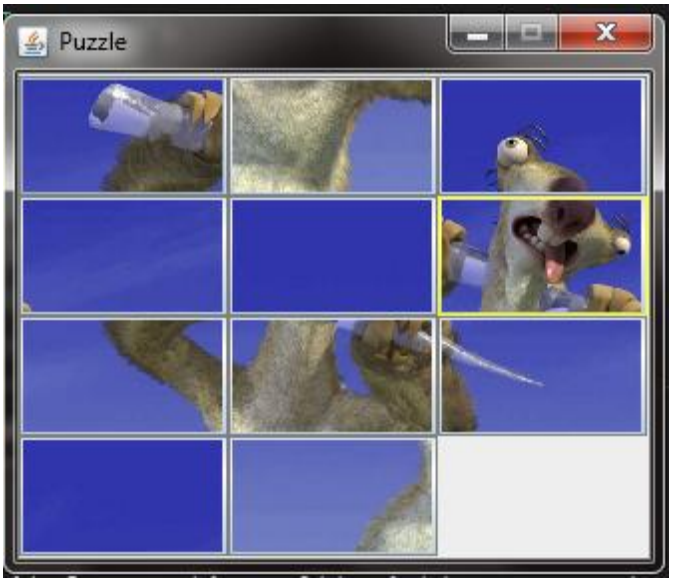
It is my privilege to acknowledge with deep sense of gratitude to our guide Mrs. Saraswati Panchakshari for her valuable suggestions and guidance throughout my course of study and timely help given to me in completion of our Project. I express my gratitude to Dr.V.S.Bandal (Principal) for their kind cooperation. I am highly obliged to the entire staff of the Computer Engineering Dept. for their kind cooperation and help. I would also like to thank my parents who patiently helped me through our work. Their support and faith is one of the reasons for the fulfillment of this Seminar. I'd also like to extend special thanks to all our colleagues and friends for helping us whenever needed. Also, a warm thanks to Library staff members for the availability of the library facilities needed for preparing this report.

Last but clearly not the least I would thank the Almighty for giving us the strength to complete my report on time.

Contents

Introduction

When you think of puzzle games, what do you think of? When most people think of puzzle games, they only have one or two images in their head. Probably a simple game with minimalistic theming and a basic tile-matching mechanic. However, puzzle games have birthed games that look and play in radically different ways from each other while still falling fully into the puzzle genre. And this diversification happened very quickly. In the space of ten years the core sub-genres were created, and then over the next 25 years puzzle games have innovated heavily, devising mechanics that push the human mind to its limits as puzzle game players have bent time and space, built amazing machines, and navigated fantastical locations. This is only a short list of the amazing diversity that has been hidden in the puzzle genre over its young life.



Case Study:

Goals:

Unlike traditional word or number puzzles, picture puzzles challenge you to discover differences or similarities between pictures or pieces of a picture. By following a few strategies (and they're not just mental!), you can solid solve a puzzle and boost your brain power at the same time.

Picture Puzzle is a relaxing puzzle game to put all the pieces together to create correct images. Picture Puzzles includes attractive image. This game helps to improve the capability of their identification skills, improve concentration, and visual brain skills.

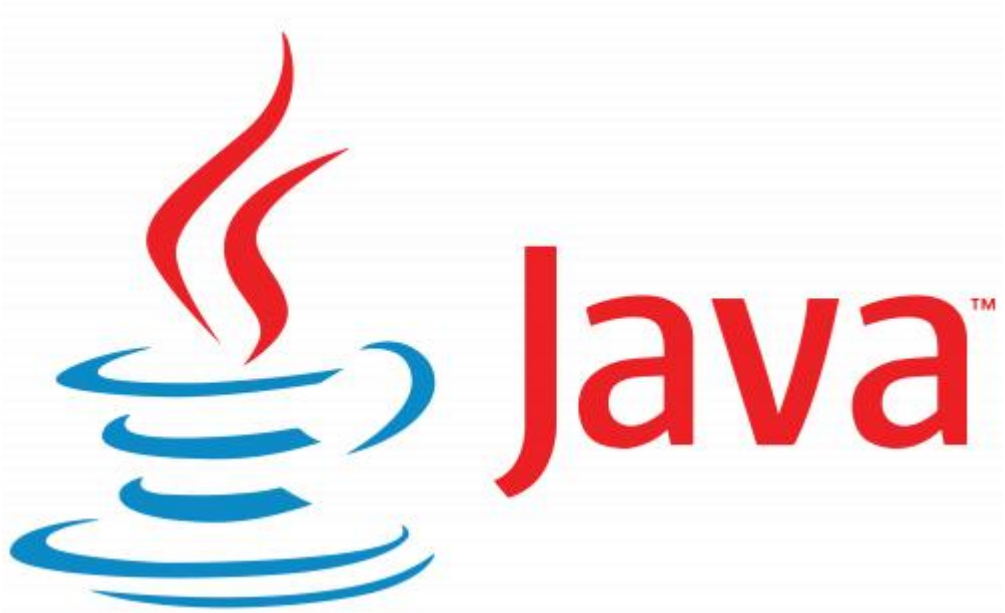
Your puzzle skills will be put to the test with this new pie-shaped puzzle game. Fit the pieces neatly in place, and create cute images.

Working Mechanics:

1) Put together a solving routine. ...
2) Check for the obvious. ...
3) Use a pattern. ...
4) Keep track of your work. ...
5) Dig out the details. ...
6) Take a break if you're stuck on a puzzle.

Technologies Used:

- **JAVA**



Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA),[17] meaning that compiled Java code can run on all platforms that support Java without the need to recompile.[18] Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture.

Pros and Cons:

The game is a good mind freshener and has an emerging experience. The UI is also attractive which makes the game addictive and engages users, giving a full-fledged game experience.

Some of the weak points of the game are that it does not save user scores which could enhance the game experience and make it more challenging.

Code:

PuzzleEx.java

```
import javax.imageio.ImageIO;
import javax.swing.AbstractAction;
import javax.swing.BorderFactory;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComponent;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.GridLayout;
import java.awt.Image;
import java.awt.Graphics2D;
import java.awt.Point;
import java.awt.event.ActionEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.image.BufferedImage;
import java.awt.image.CropImageFilter;
import java.awt.image.FilteredImageSource;
import java.io.File;
```

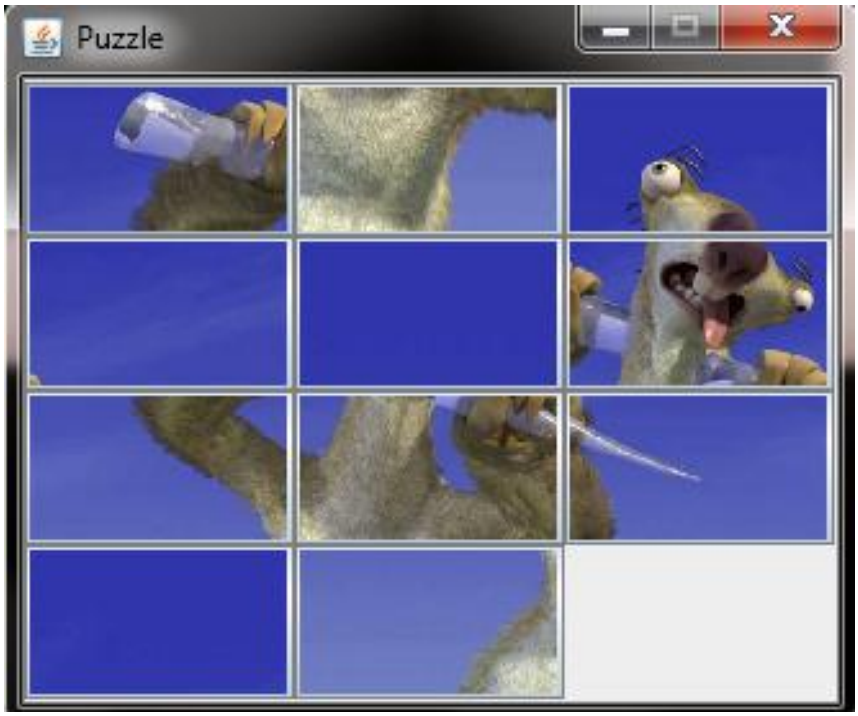
```
import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
class MyButton extends JButton {
    private boolean isLastButton;
    public MyButton() {
        super();
        initUI();
    }
    public MyButton(Image image) {
        super(new ImageIcon(image));
        initUI();
    }
    private void initUI() {
        isLastButton = false;
        BorderFactory.createLineBorder(Color.gray);
        addMouseListener(new MouseAdapter() {
            @Override
            public void mouseEntered(MouseEvent e) {
                setBorder(BorderFactory.createLineBorder(Color.yellow));
            }
            @Override
            public void mouseExited(MouseEvent e) {
                setBorder(BorderFactory.createLineBorder(Color.gray));
            }
        });
    }
    public void setLastButton() {
        isLastButton = true;
    }
    public boolean isLastButton() {
        return isLastButton;
    }
}
public class PuzzleEx extends JFrame {
    private JPanel panel;
    private BufferedImage source;
    private BufferedImage resized;
    private Image image;
    private MyButton lastButton;
    private int width, height;
    private List<MyButton> buttons;
    private List<Point> solution;
    private final int NUMBER_OF_BUTTONS = 12;
    private final int DESIRED_WIDTH = 300;
    public PuzzleEx() {
        initUI();
    }
```

```
    private void initUI() {
        solution = new ArrayList<>();
        solution.add(new Point(0, 0));
        solution.add(new Point(0, 1));
        solution.add(new Point(0, 2));
        solution.add(new Point(1, 0));
        solution.add(new Point(1, 1));
        solution.add(new Point(1, 2));
        solution.add(new Point(2, 0));
        solution.add(new Point(2, 1));
        solution.add(new Point(2, 2));
        solution.add(new Point(3, 0));
        solution.add(new Point(3, 1));
        solution.add(new Point(3, 2));
        buttons = new ArrayList<>();
        panel = new JPanel();
        panel.setBorder(BorderFactory.createLineBorder(Color.gray));
        panel.setLayout(new GridLayout(4, 3, 0, 0));
        try {
            source = loadImage();
            int h = getNewHeight(source.getWidth(), source.getHeight());
            resized = resizeImage(source, DESIRED_WIDTH, h,
                    BufferedImage.TYPE_INT_ARGB);
        } catch (IOException ex) {
            JOptionPane.showMessageDialog(this, "Could not load image",
"Error",
                    JOptionPane.ERROR_MESSAGE);
        }
        width = resized.getWidth(null);
        height = resized.getHeight(null);
        add(panel, BorderLayout.CENTER);
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 3; j++) {
                image = createImage(new
FilteredImageSource(resized.getSource(),
                        new CropImageFilter(j * width / 3, i * height /
4,
                                (width / 3), height / 4)));
                MyButton button = new MyButton(image);
                button.putClientProperty("position", new Point(i, j));
                if (i == 3 && j == 2) {
                    lastButton = new MyButton();
                    lastButton.setBorderPainted(false);
                    lastButton.setContentAreaFilled(false);
                    lastButton.setLastButton();
                    lastButton.putClientProperty("position", new Point(i
j));
                } else {
                    buttons.add(button);
```

```
                }
            }
        }
        Collections.shuffle(buttons);
        buttons.add(lastButton);
        for (int i = 0; i < NUMBER_OF_BUTTONS; i++) {
            MyButton btn = buttons.get(i);
            panel.add(btn);
            btn.setBorder(BorderFactory.createLineBorder(Color.gray));
            btn.addActionListener(new ClickAction());
        }
        pack();
        setTitle("Puzzle");
        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
    }
    private int getNewHeight(int w, int h) {
        double ratio = DESIRED_WIDTH / (double) w;
        int newHeight = (int) (h * ratio);
        return newHeight;
    }
    private BufferedImage loadImage() throws IOException {
        BufferedImage bimg = ImageIO
                .read(new File("./icesid.jpg"));
        return bimg;
    }
    private BufferedImage resizeImage(BufferedImage originalImage, int
width,
            int height, int type) {
        BufferedImage resizedImage = new BufferedImage(width, height,
type);
        Graphics2D g = resizedImage.createGraphics();
        g.drawImage(originalImage, 0, 0, width, height, null);
       g.dispose();
        return resizedImage;
    }
    private class ClickAction extends AbstractAction {
        @Override
        public void actionPerformed(ActionEvent e) {
            checkButton(e);
            checkSolution();
        }
        private void checkButton(ActionEvent e) {
            int lidx = 0;
            for (MyButton button : buttons) {
                if (button.isLastButton()) {
                    lidx = buttons.indexOf(button);
                }
```

```
            }
            JButton button = (JButton) e.getSource();
            int bidx = buttons.indexOf(button);
            if ((bidx - 1 == lidx) || (bidx + 1 == lidx)
                    || (bidx - 3 == lidx) || (bidx + 3 == lidx)) {
                Collections.swap(buttons, bidx, lidx);
                updateButtons();
            }
        }
        private void updateButtons() {
            panel.removeAll();
            for (JComponent btn : buttons) {
                panel.add(btn);
            }
            panel.validate();
        }
    }
    private void checkSolution() {
        ArrayList<Point> current = new ArrayList<Point>();
        for (JComponent btn : buttons) {
            current.add((Point) btn.getClientProperty("position"));
        }
        if (compareList(solution, current)) {
            JOptionPane.showMessageDialog(panel, "Finished",
                    "Congratulation", JOptionPane.INFORMATION_MESSAGE);
        }
    }
    public static boolean compareList(List ls1, List ls2) {
        return ls1.toString().contentEquals(ls2.toString());
    }
    public static void main(String[] args) {
        EventQueue.invokeLater(() -> {
            PuzzleEx puzzle = new PuzzleEx();
            puzzle.setVisible(true);
        });
    }
}
```

The Game:



References:

Steps for Puzzle Game

Conclusion:

Thus, we built a Puzzle game which is a mind refreshing game using Java. Through this project we learnt about the applications of Java and learnt to work as a team.