JAVA PROJECT REPORT

(Project Term January-May 2023)

OCTO-PUZZLE

Submitted by

Amandeep Singh **Registration Number: 12101332**

Course Code: CSE310

Under the Guidance of

Dr. Ranjith Kumar A (Assistant Professor)

School of Computer Science and Engineering

LOVELY
PROFESSIONAL
UNIVERSITY

DECLARATION

We hereby declare that the project work entitled ("8-Puzzle game") is an authentic record of our own work carried out as requirements of the project for the award of B.Tech degree in (Computer Science and Engineering) from Lovely Professional University, Phagwara, under the guidance of (Dr. Ranjith Kumar A), during January to April 2023. All the information furnished in this project report is based on our own intensive work and is genuine.

Group Project

Name of Student 1: …Amandeep Singh………
Registration Number: ...12101332…………….

Table of contents

Introduction

Puzzle games have been a popular form of entertainment for decades, challenging players to solve complex problems and think creatively. These games often require critical thinking and problem-solving skills, providing a mental workout for players. Puzzle games come in many different forms, from classic jigsaw puzzles to modern digital games. They can be played alone or with friends, making them a versatile option for any social setting. With the rise of mobile gaming, puzzle games have become more accessible than ever before. Many puzzle games are free or inexpensive, making them a great way to pass the time on a budget. Whether you're a seasoned puzzle master or just looking for a fun challenge, there's a puzzle game out there for everyone.

Puzzle games can be both relaxing and stimulating, as they allow players to focus on a task and forget about the outside world for a while. They often come with soothing music and visuals that enhance the gameplay experience. Some puzzle games are even designed to be meditative and calming, helping players to reduce stress and anxiety. Additionally, puzzle games can improve cognitive skills such as memory, attention, and spatial reasoning. As players progress through the levels, they may encounter more difficult challenges that require them to think outside the box and develop new strategies. Overall, puzzle games are a fun and rewarding way to exercise the mind and pass the time.

SCOPE OF THE PROJECT

1->**Huge audience**: Puzzle games have a massive audience worldwide, including casual players, hardcore gamers, and people of all ages and backgrounds.

2->**Diverse Platforms**: With the increasing availability of smartphones, tablets, and other devices, puzzle games can be played on multiple platforms, including mobile, PC, and consoles.

3->**High Demand**: The demand for puzzle games continues to increase due to their accessibility, addictive gameplay, and low barrier to entry.

4->**Innovation**: Developers are continuously pushing the boundaries of puzzle game design and mechanics, creating unique and engaging experiences that captivate players.

5->**Variety**: There is a wide variety of puzzle game types available, including jigsaw puzzles, match-three games, crossword puzzles, and more, providing endless possibilities for game creation.

6->**Creativity**: Puzzle games offer plenty of room for creativity and innovation, allowing developers to experiment with different game mechanics, themes, and art styles.

7->**Education and Therapy**: Puzzle games have been found to have therapeutic and educational benefits, making them ideal for individuals seeking entertainment as well as mental stimulation.

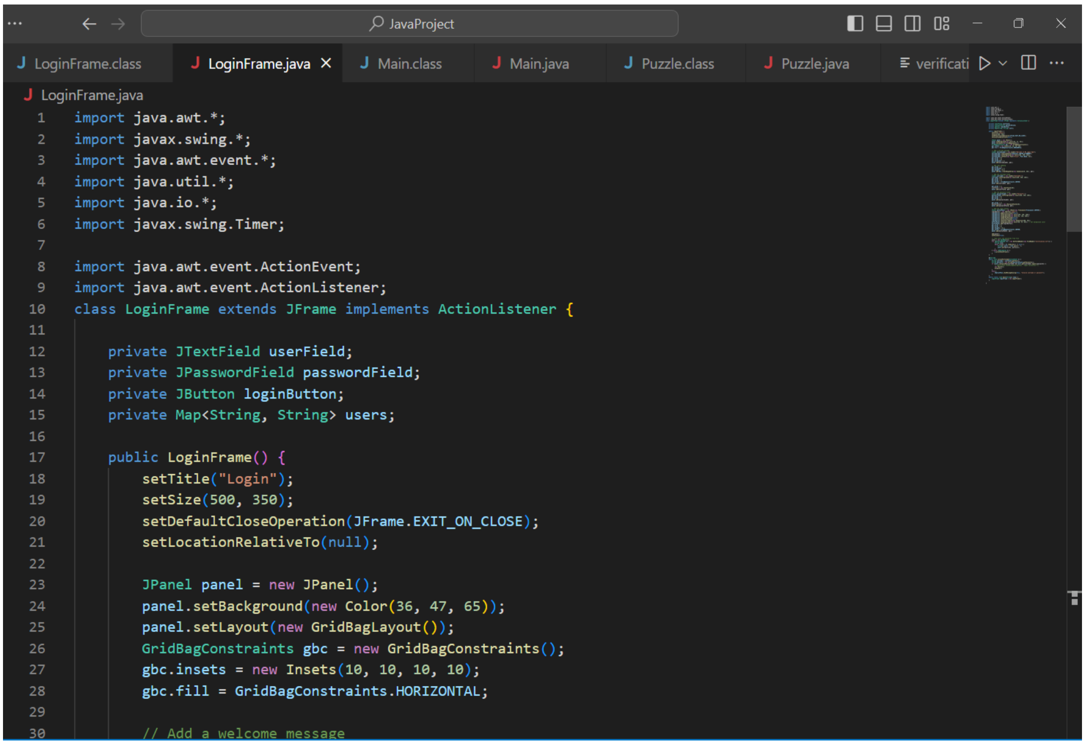
8->**Monetization**: Puzzle games can be monetized in several ways, including in-app purchases, ads, and subscriptions, making them a potentially lucrative opportunity for developers.

9->**Competitive Space**: The puzzle game market is highly competitive, which means developers must continually innovate to stand out and keep players engaged.

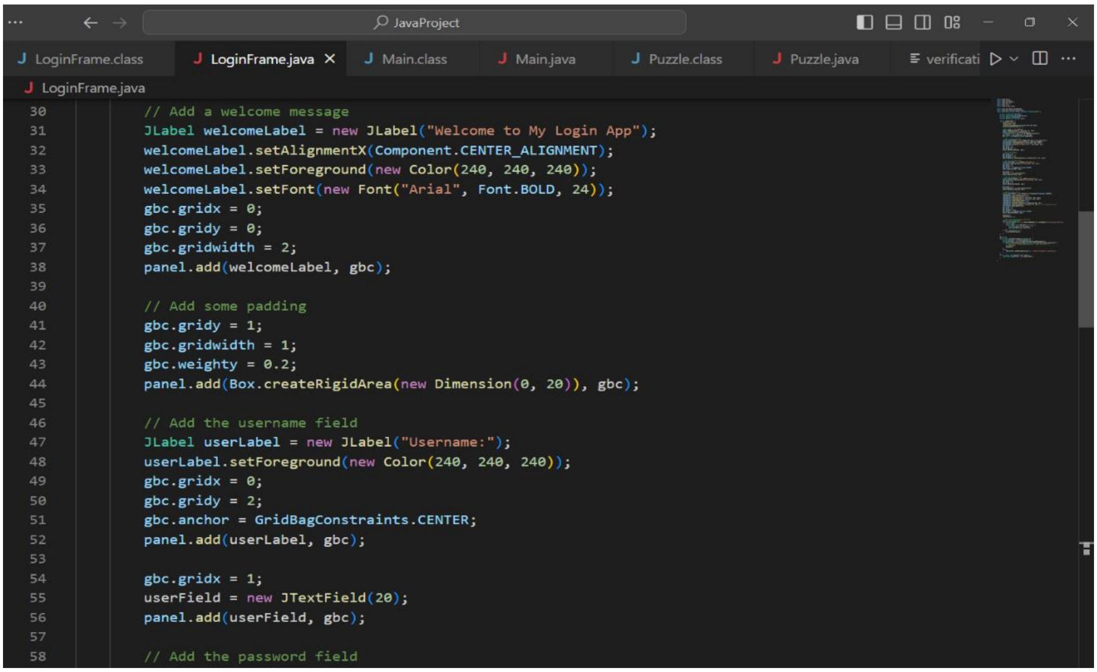
10->**Community Building**: Puzzle games can help build a strong community of dedicated players who share strategies, tips, and feedback, which can lead to further improvements and success for the game.

MODULES WITH SOURCE CODE

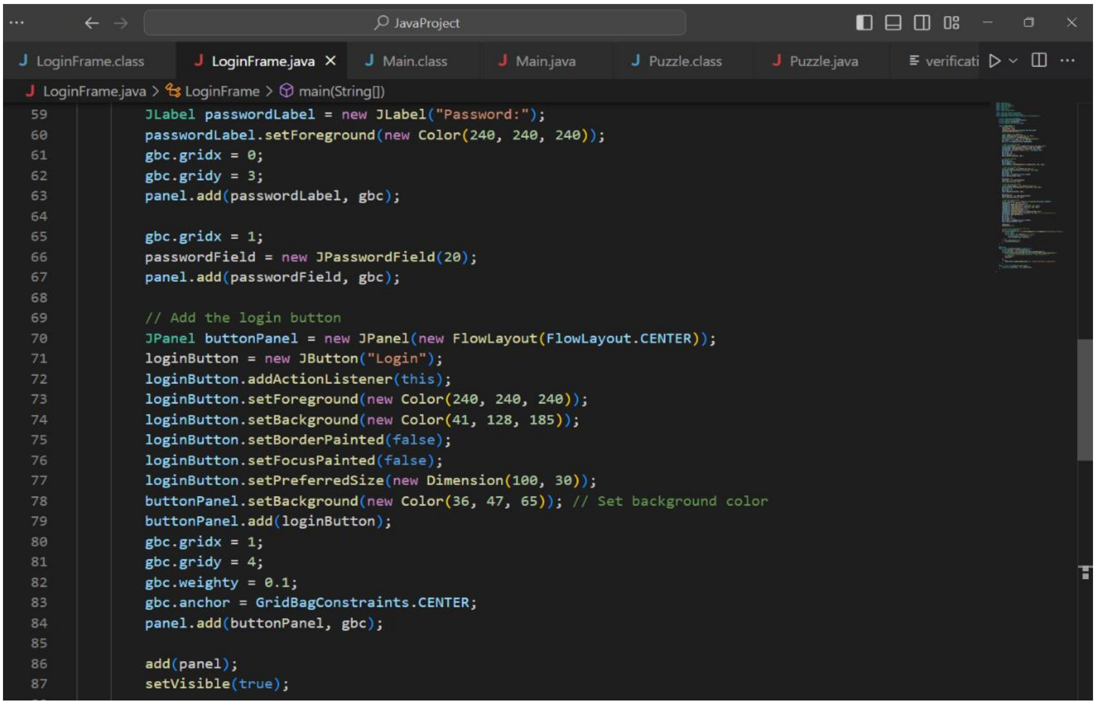
1. **LOGIN PAGE:** A login page is the page that requires users to enter their login credentials, such as a username and password, to gain access to the application. The login page typically includes a form where users can enter their credentials, as well as a login button to initiate the login process.



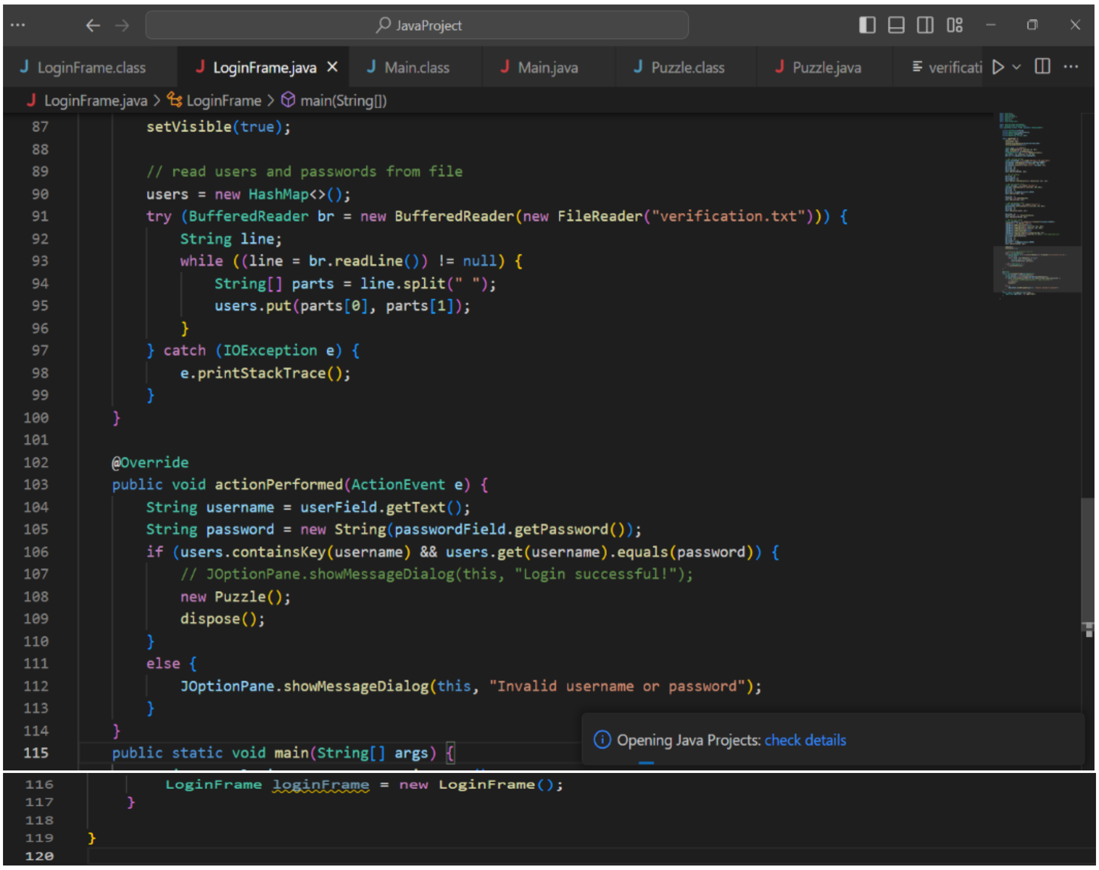
```java
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.util.*;
import java.io.*;
import javax.swing.Timer;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
class LoginFrame extends JFrame implements ActionListener {

    private JTextField userField;
    private JPasswordField passwordField;
    private JButton loginButton;
    private Map<String, String> users;

    public LoginFrame() {
        setTitle("Login");
        setSize(500, 350);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        JPanel panel = new JPanel();
        panel.setBackground(new Color(36, 47, 65));
        panel.setLayout(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10);
        gbc.fill = GridBagConstraints.HORIZONTAL;

        // Add a welcome message
```



```java
        // Add a welcome message
        JLabel welcomeLabel = new JLabel("Welcome to My Login App");
        welcomeLabel.setAlignmentX(Component.CENTER_ALIGNMENT);
        welcomeLabel.setForeground(new Color(240, 240, 240));
        welcomeLabel.setFont(new Font("Arial", Font.BOLD, 24));
        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.gridwidth = 2;
        panel.add(welcomeLabel, gbc);

        // Add some padding
        gbc.gridy = 1;
        gbc.gridwidth = 1;
        gbc.weighty = 0.2;
        panel.add(Box.createRigidArea(new Dimension(0, 20)), gbc);

        // Add the username field
        JLabel userLabel = new JLabel("Username:");
        userLabel.setForeground(new Color(240, 240, 240));
        gbc.gridx = 0;
        gbc.gridy = 2;
        gbc.anchor = GridBagConstraints.CENTER;
        panel.add(userLabel, gbc);

        gbc.gridx = 1;
        userField = new JTextField(20);
        panel.add(userField, gbc);

        // Add the password field
```

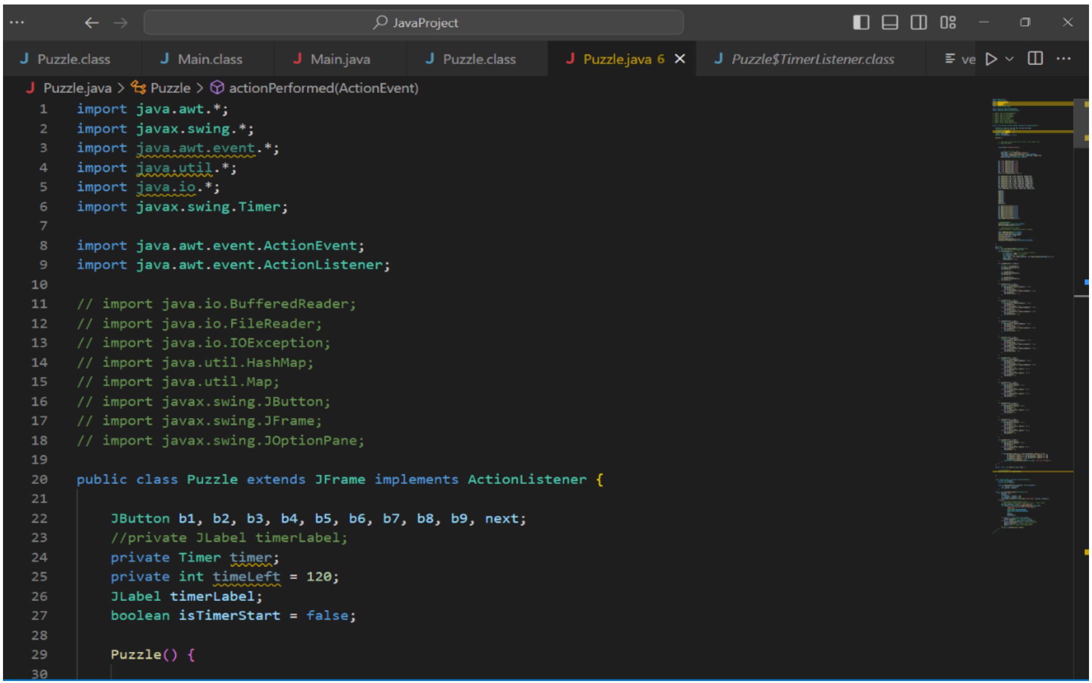


```java
        JLabel passwordLabel = new JLabel("Password:");
        passwordLabel.setForeground(new Color(240, 240, 240));
        gbc.gridx = 0;
        gbc.gridy = 3;
        panel.add(passwordLabel, gbc);

        gbc.gridx = 1;
        passwordField = new JPasswordField(20);
        panel.add(passwordField, gbc);

        // Add the login button
        JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
        loginButton = new JButton("Login");
        loginButton.addActionListener(this);
        loginButton.setForeground(new Color(240, 240, 240));
        loginButton.setBackground(new Color(41, 128, 185));
        loginButton.setBorderPainted(false);
        loginButton.setFocusPainted(false);
        loginButton.setPreferredSize(new Dimension(100, 30));
        buttonPanel.setBackground(new Color(36, 47, 65)); // Set background color
        buttonPanel.add(loginButton);
        gbc.gridx = 1;
        gbc.gridy = 4;
        gbc.weighty = 0.1;
        gbc.anchor = GridBagConstraints.CENTER;
        panel.add(buttonPanel, gbc);

        add(panel);
        setVisible(true);
```

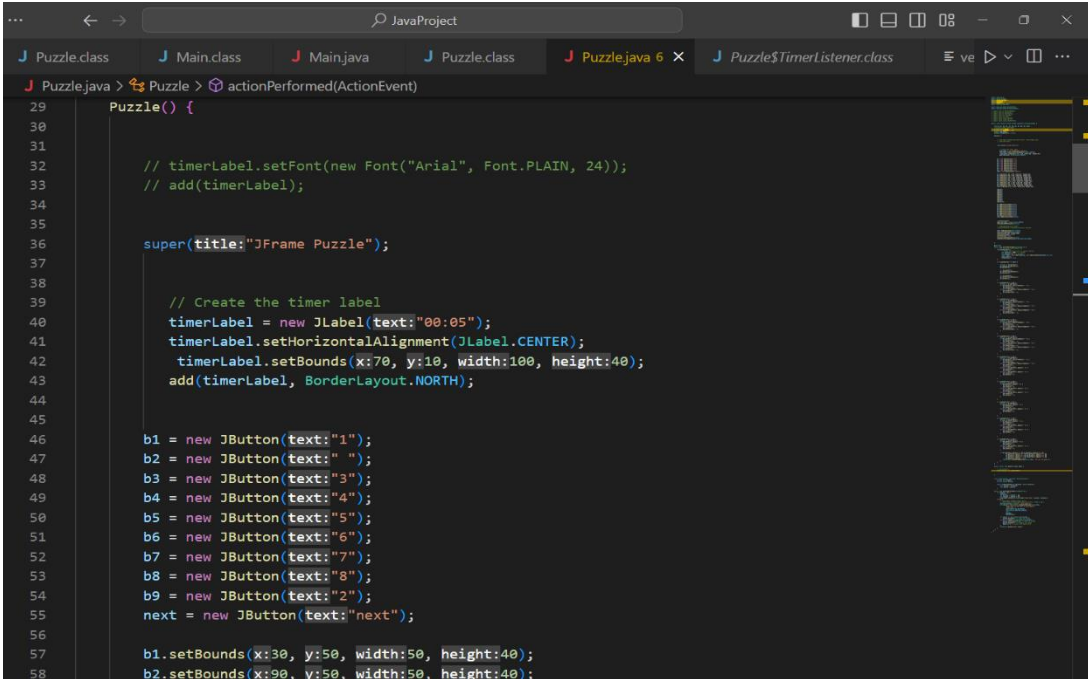


```java
        setVisible(true);

        // read users and passwords from file
        users = new HashMap<>();
        try (BufferedReader br = new BufferedReader(new FileReader("verification.txt"))) {
            String line;
            while ((line = br.readLine()) != null) {
                String[] parts = line.split(" ");
                users.put(parts[0], parts[1]);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        String username = userField.getText();
        String password = new String(passwordField.getPassword());
        if (users.containsKey(username) && users.get(username).equals(password)) {
            // JOptionPane.showMessageDialog(this, "Login successful!");
            new Puzzle();
            dispose();
        }
        else {
            JOptionPane.showMessageDialog(this, "Invalid username or password");
        }
    }
    public static void main(String[] args) {
        LoginFrame loginFrame = new LoginFrame();
    }

}
```

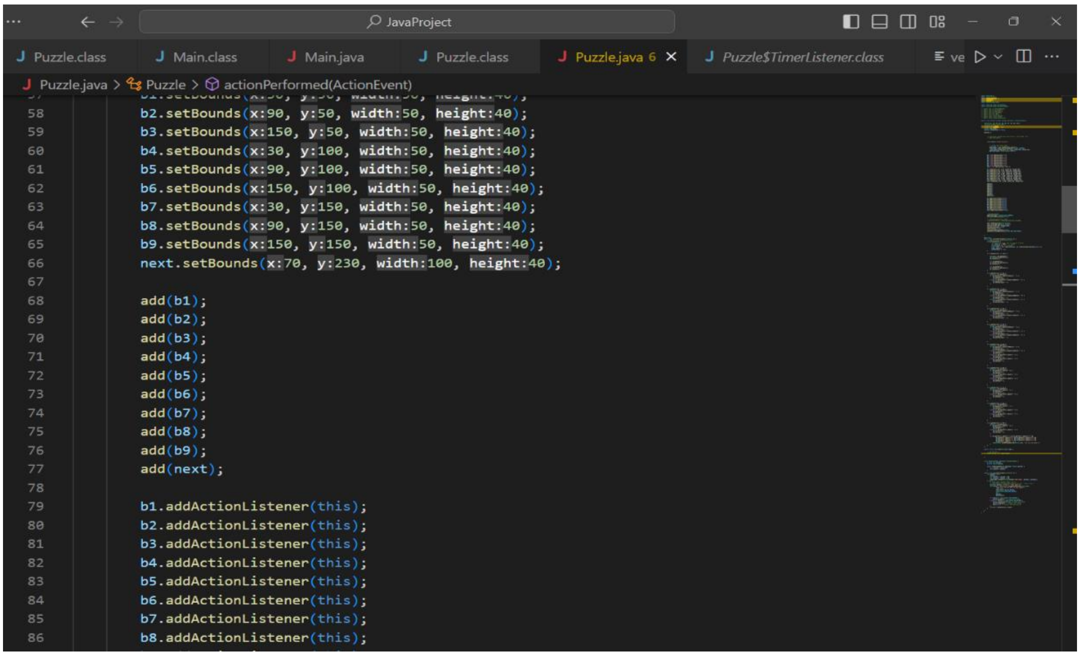
2. **PUZZLE MODULE:** The puzzle game code typically includes several components that work together to create an interactive gaming experience. The code will generally include a graphical user interface (GUI) that allows the user to interact with the game. This GUI includes buttons, sliders, or other interactive elements that the user can manipulate to solve the puzzle. The core logic of the puzzle game will also be included in the code. This logic will define the rules of the puzzle, determine how the pieces interact with each other, and determine how the game is won or lost. The game code may also include algorithms to generate new puzzles, so that the game can offer a variety of different challenges to the user. The game also features a timer of a set time completion of which ends the session and the you are given the choice to retry or exit the game.


JavaProject
Puzzle.class
Main.class
Main.java
Puzzle.class
Puzzle.java 6
Puzzle$TimerListener.class
Puzzle.java > Puzzle > actionPerformed(ActionEvent)

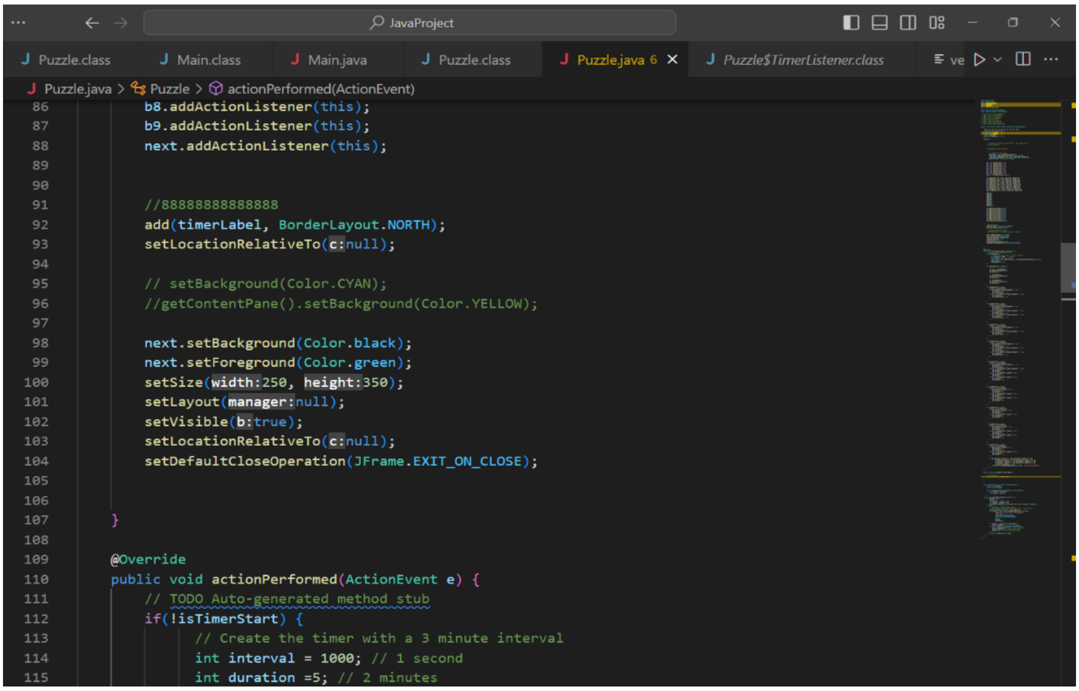
```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.util.*;
import java.io.*;
import javax.swing.Timer;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

// import java.io.BufferedReader;
// import java.io.FileReader;
// import java.io.IOException;
// import java.util.HashMap;
// import java.util.Map;
// import javax.swing.JButton;
// import javax.swing.JFrame;
// import javax.swing.JOptionPane;

public class Puzzle extends JFrame implements ActionListener {

    JButton b1, b2, b3, b4, b5, b6, b7, b8, b9, next;
    //private JLabel timerLabel;
    private Timer timer;
    private int timeLeft = 120;
    JLabel timerLabel;
    boolean isTimerStart = false;

    Puzzle() {

```


JavaProject
Puzzle.class
Main.class
Main.java
Puzzle.class
Puzzle.java 6
Puzzle$TimerListener.class
Puzzle.java > Puzzle > actionPerformed(ActionEvent)

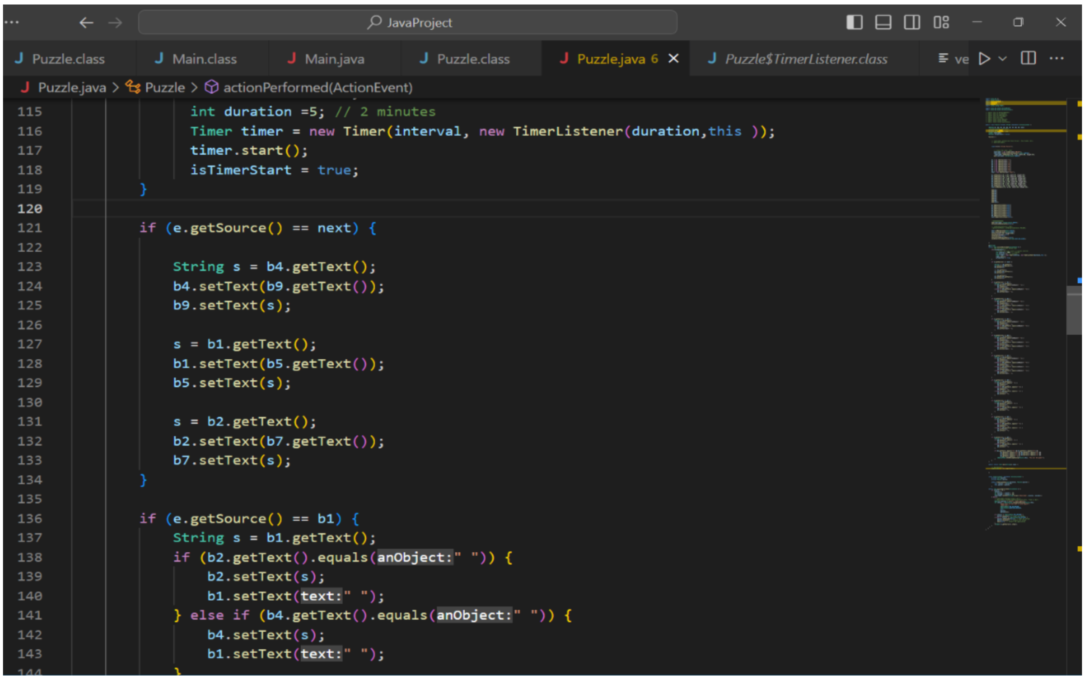
```
    Puzzle() {


        // timerLabel.setFont(new Font("Arial", Font.PLAIN, 24));
        // add(timerLabel);


        super(title:"JFrame Puzzle");


           // Create the timer label
           timerLabel = new JLabel(text:"00:05");
           timerLabel.setHorizontalAlignment(JLabel.CENTER);
            timerLabel.setBounds(x:70, y:10, width:100, height:40);
           add(timerLabel, BorderLayout.NORTH);


        b1 = new JButton(text:"1");
        b2 = new JButton(text:" ");
        b3 = new JButton(text:"3");
        b4 = new JButton(text:"4");
        b5 = new JButton(text:"5");
        b6 = new JButton(text:"6");
        b7 = new JButton(text:"7");
        b8 = new JButton(text:"8");
        b9 = new JButton(text:"2");
        next = new JButton(text:"next");

        b1.setBounds(x:30, y:50, width:50, height:40);
        b2.setBounds(x:90, y:50, width:50, height:40);
```


JavaProject
Puzzle.class
Main.class
Main.java
Puzzle.class
Puzzle.java 6
Puzzle$TimerListener.class
ve
Puzzle.java > Puzzle > actionPerformed(ActionEvent)

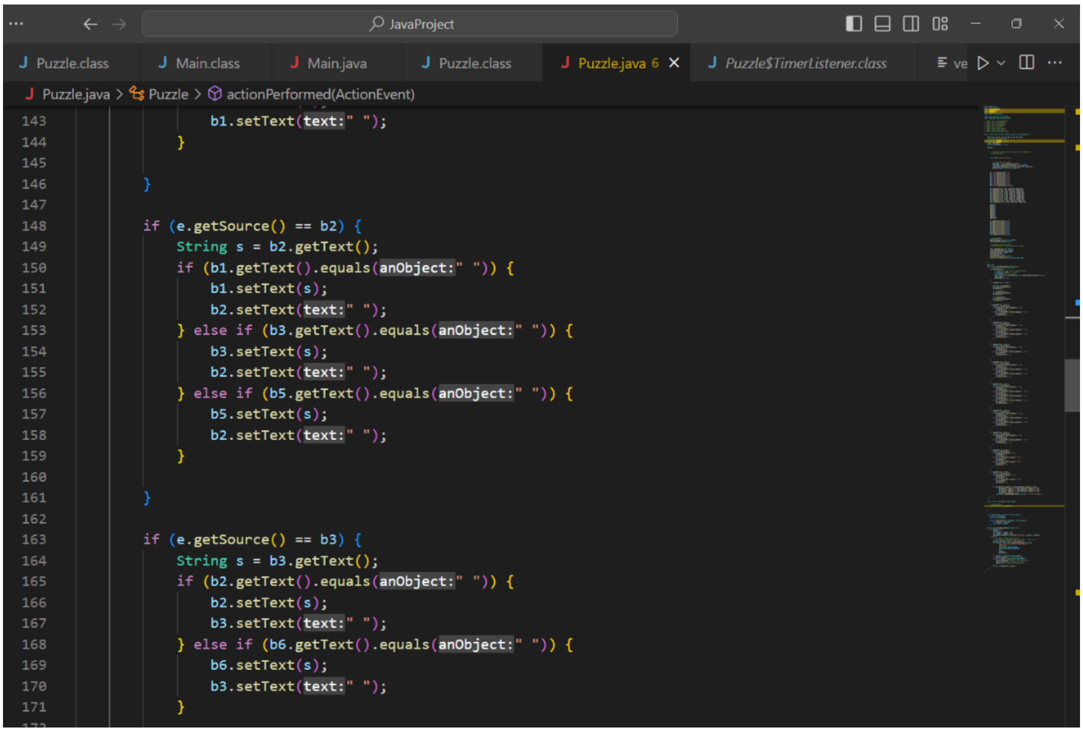
```java
        b1.setBounds(x:30, y:50, width:50, height:40);
        b2.setBounds(x:90, y:50, width:50, height:40);
        b3.setBounds(x:150, y:50, width:50, height:40);
        b4.setBounds(x:30, y:100, width:50, height:40);
        b5.setBounds(x:90, y:100, width:50, height:40);
        b6.setBounds(x:150, y:100, width:50, height:40);
        b7.setBounds(x:30, y:150, width:50, height:40);
        b8.setBounds(x:90, y:150, width:50, height:40);
        b9.setBounds(x:150, y:150, width:50, height:40);
        next.setBounds(x:70, y:230, width:100, height:40);

        add(b1);
        add(b2);
        add(b3);
        add(b4);
        add(b5);
        add(b6);
        add(b7);
        add(b8);
        add(b9);
        add(next);

        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        b4.addActionListener(this);
        b5.addActionListener(this);
        b6.addActionListener(this);
        b7.addActionListener(this);
        b8.addActionListener(this);
```


JavaProject
Puzzle.class
Main.class
Main.java
Puzzle.class
Puzzle.java 6
Puzzle$TimerListener.class
ve
Puzzle.java > Puzzle > actionPerformed(ActionEvent)

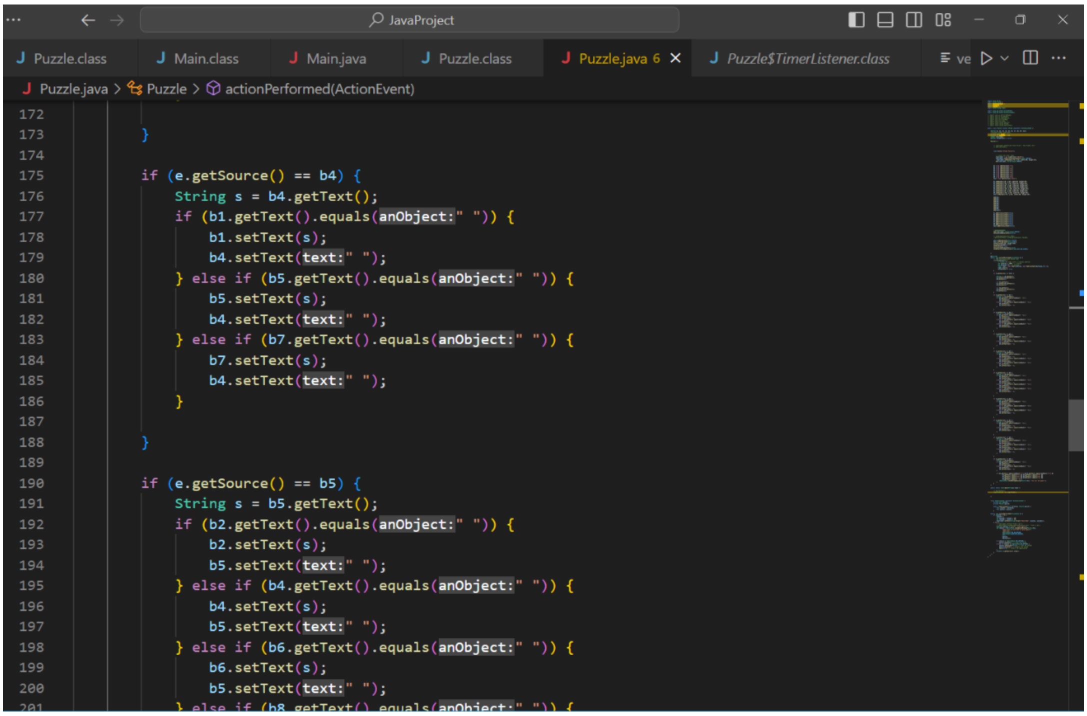
```java
        b8.addActionListener(this);
        b9.addActionListener(this);
        next.addActionListener(this);


        //88888888888888
        add(timerLabel, BorderLayout.NORTH);
        setLocationRelativeTo(c:null);

        // setBackground(Color.CYAN);
        //getContentPane().setBackground(Color.YELLOW);

        next.setBackground(Color.black);
        next.setForeground(Color.green);
        setSize(width:250, height:350);
        setLayout(manager:null);
        setVisible(b:true);
        setLocationRelativeTo(c:null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);


    }

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        if(!isTimerStart) {
            // Create the timer with a 3 minute interval
            int interval = 1000; // 1 second
            int duration =5; // 2 minutes
```


JavaProject
Puzzle.class
Main.class
Main.java
Puzzle.class
Puzzle.java 6
Puzzle$TimerListener.class
Puzzle.java > Puzzle > actionPerformed(ActionEvent)

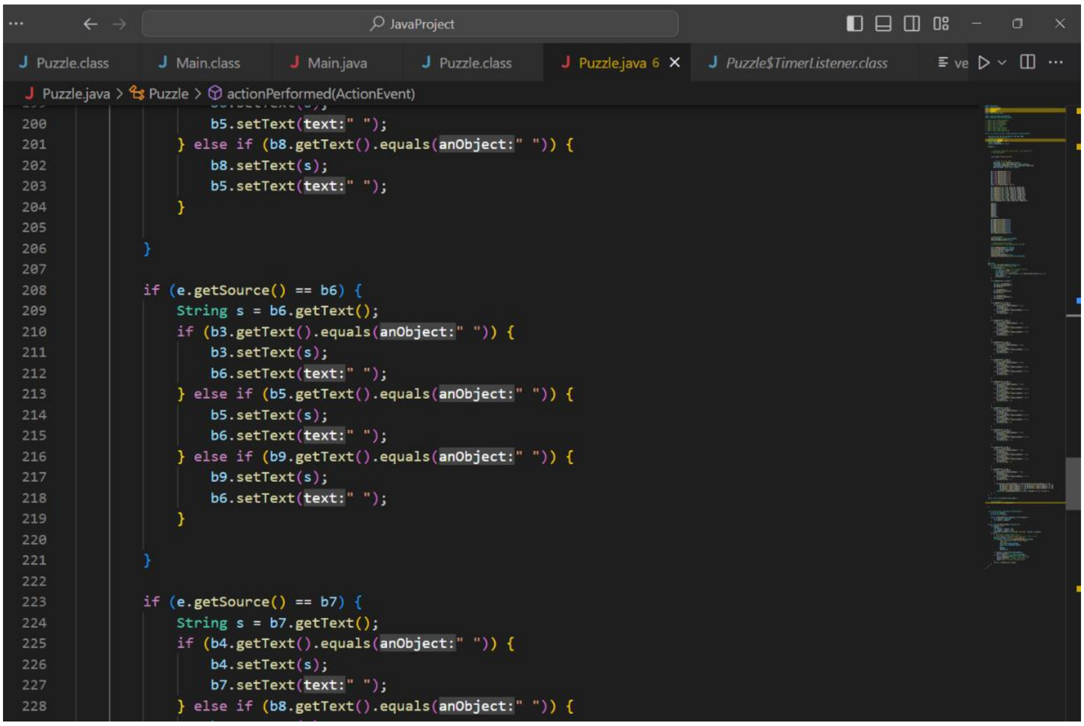
```java
              int duration =5; // 2 minutes
              Timer timer = new Timer(interval, new TimerListener(duration,this ));
              timer.start();
              isTimerStart = true;
        }

        if (e.getSource() == next) {

            String s = b4.getText();
            b4.setText(b9.getText());
            b9.setText(s);

            s = b1.getText();
            b1.setText(b5.getText());
            b5.setText(s);

            s = b2.getText();
            b2.setText(b7.getText());
            b7.setText(s);
        }

        if (e.getSource() == b1) {
            String s = b1.getText();
            if (b2.getText().equals(anObject:" ")) {
                b2.setText(s);
                b1.setText(text:" ");
            } else if (b4.getText().equals(anObject:" ")) {
                b4.setText(s);
                b1.setText(text:" ");
```


JavaProject
Puzzle.class
Main.class
Main.java
Puzzle.class
Puzzle.java 6
Puzzle$TimerListener.class
Puzzle.java > Puzzle > actionPerformed(ActionEvent)

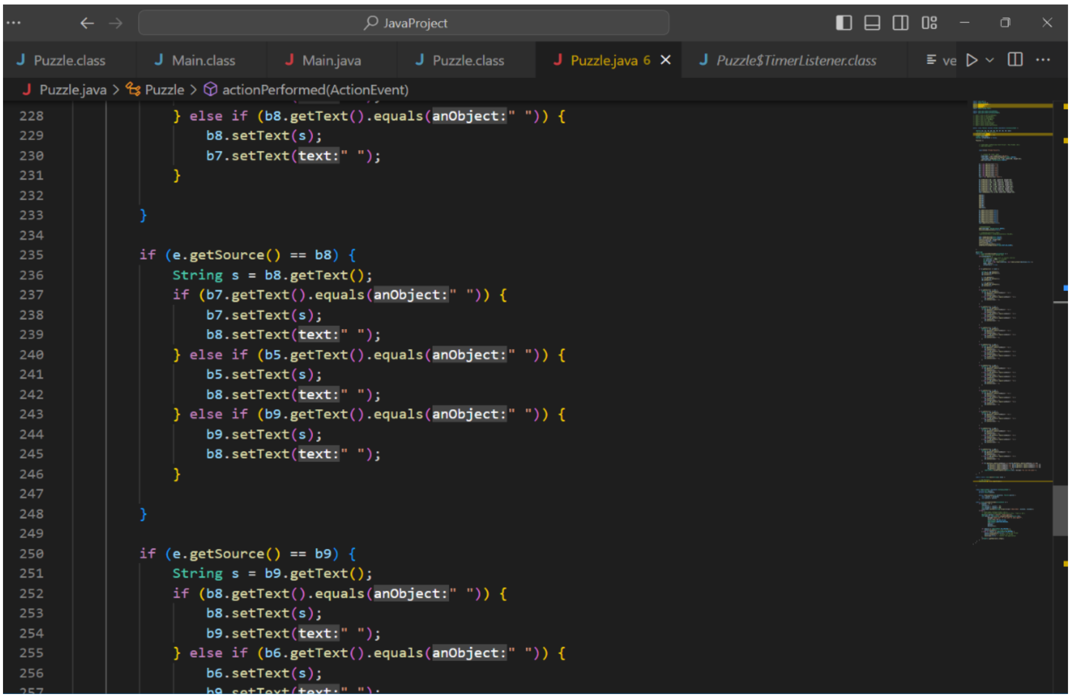
```java
                b1.setText(text:" ");
            }

        }

        if (e.getSource() == b2) {
            String s = b2.getText();
            if (b1.getText().equals(anObject:" ")) {
                b1.setText(s);
                b2.setText(text:" ");
            } else if (b3.getText().equals(anObject:" ")) {
                b3.setText(s);
                b2.setText(text:" ");
            } else if (b5.getText().equals(anObject:" ")) {
                b5.setText(s);
                b2.setText(text:" ");
            }

        }

        if (e.getSource() == b3) {
            String s = b3.getText();
            if (b2.getText().equals(anObject:" ")) {
                b2.setText(s);
                b3.setText(text:" ");
            } else if (b6.getText().equals(anObject:" ")) {
                b6.setText(s);
                b3.setText(text:" ");
            }
```


JavaProject
Puzzle.class
Main.class
Main.java
Puzzle.class
Puzzle.java 6
Puzzle$TimerListener.class
Puzzle.java > Puzzle > actionPerformed(ActionEvent)

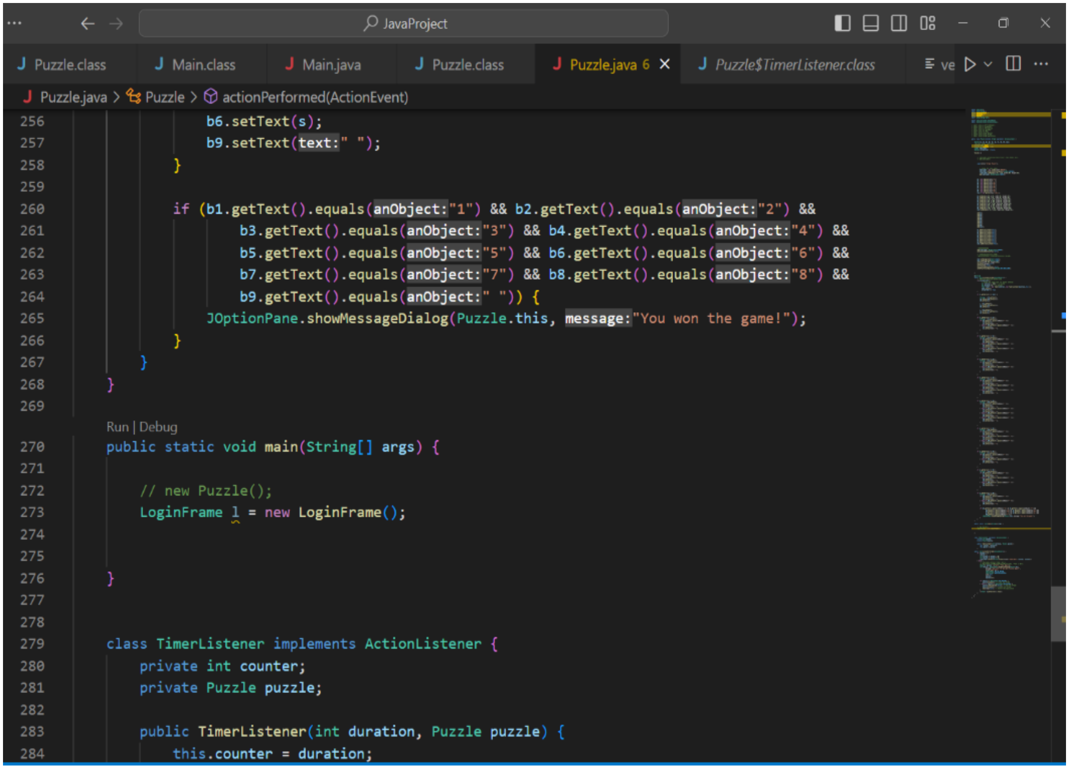
```java

        }

        if (e.getSource() == b4) {
            String s = b4.getText();
            if (b1.getText().equals(anObject:" ")) {
                b1.setText(s);
                b4.setText(text:" ");
            } else if (b5.getText().equals(anObject:" ")) {
                b5.setText(s);
                b4.setText(text:" ");
            } else if (b7.getText().equals(anObject:" ")) {
                b7.setText(s);
                b4.setText(text:" ");
            }

        }

        if (e.getSource() == b5) {
            String s = b5.getText();
            if (b2.getText().equals(anObject:" ")) {
                b2.setText(s);
                b5.setText(text:" ");
            } else if (b4.getText().equals(anObject:" ")) {
                b4.setText(s);
                b5.setText(text:" ");
            } else if (b6.getText().equals(anObject:" ")) {
                b6.setText(s);
                b5.setText(text:" ");
            } else if (b8.getText().equals(anObject:" ")) {
```


JavaProject
Puzzle.class
Main.class
Main.java
Puzzle.class
Puzzle.java 6
Puzzle$TimerListener.class
Puzzle.java > Puzzle > actionPerformed(ActionEvent)

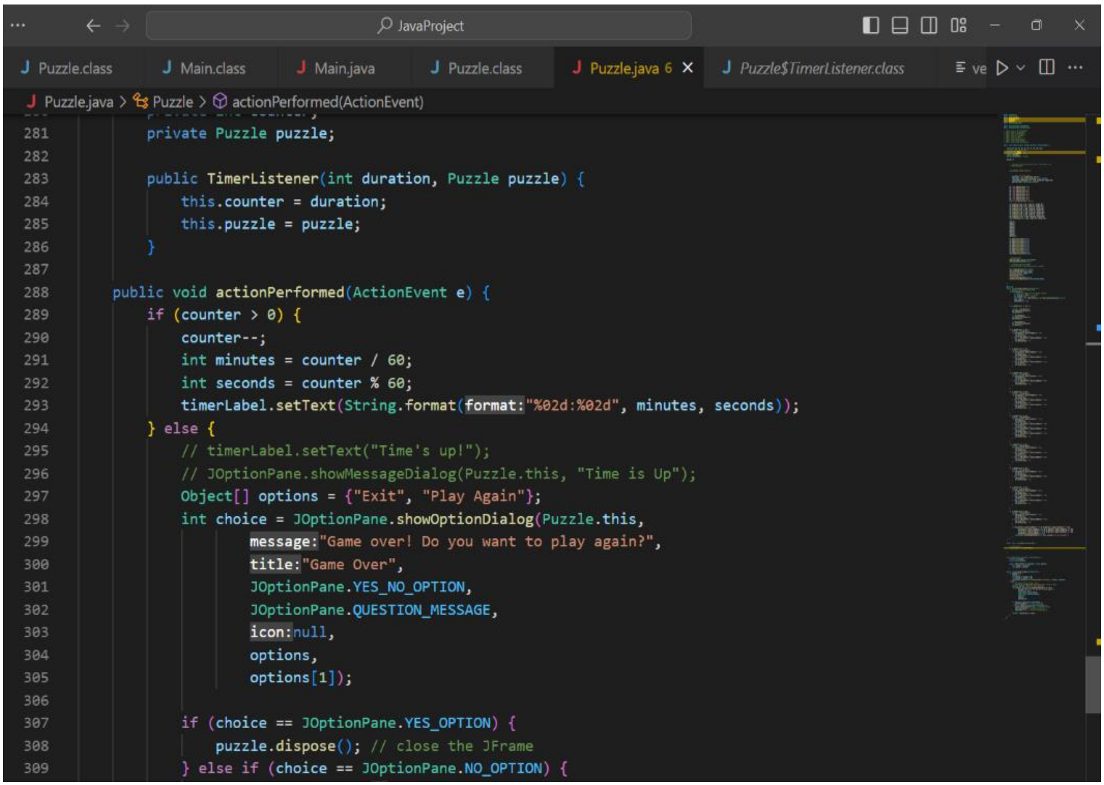
```java
                b5.setText(text:" ");
            } else if (b8.getText().equals(anObject:" ")) {
                b8.setText(s);
                b5.setText(text:" ");
            }

        }

        if (e.getSource() == b6) {
            String s = b6.getText();
            if (b3.getText().equals(anObject:" ")) {
                b3.setText(s);
                b6.setText(text:" ");
            } else if (b5.getText().equals(anObject:" ")) {
                b5.setText(s);
                b6.setText(text:" ");
            } else if (b9.getText().equals(anObject:" ")) {
                b9.setText(s);
                b6.setText(text:" ");
            }

        }

        if (e.getSource() == b7) {
            String s = b7.getText();
            if (b4.getText().equals(anObject:" ")) {
                b4.setText(s);
                b7.setText(text:" ");
            } else if (b8.getText().equals(anObject:" ")) {
```


JavaProject
Puzzle.class
Main.class
Main.java
Puzzle.class
Puzzle.java 6
Puzzle$TimerListener.class
Puzzle.java > Puzzle > actionPerformed(ActionEvent)

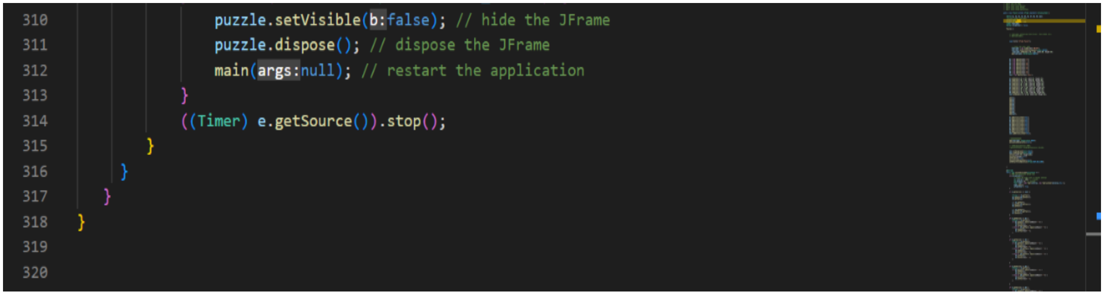
```
            } else if (b8.getText().equals(anObject:" ")) {
                b8.setText(s);
                b7.setText(text:" ");
            }

        }

        if (e.getSource() == b8) {
            String s = b8.getText();
            if (b7.getText().equals(anObject:" ")) {
                b7.setText(s);
                b8.setText(text:" ");
            } else if (b5.getText().equals(anObject:" ")) {
                b5.setText(s);
                b8.setText(text:" ");
            } else if (b9.getText().equals(anObject:" ")) {
                b9.setText(s);
                b8.setText(text:" ");
            }

        }

        if (e.getSource() == b9) {
            String s = b9.getText();
            if (b8.getText().equals(anObject:" ")) {
                b8.setText(s);
                b9.setText(text:" ");
            } else if (b6.getText().equals(anObject:" ")) {
                b6.setText(s);
                b9.setText(text:" ");
```


JavaProject
Puzzle.class
Main.class
Main.java
Puzzle.class
Puzzle.java 6
Puzzle$TimerListener.class
Puzzle.java > Puzzle > actionPerformed(ActionEvent)

```
                b6.setText(s);
                b9.setText(text:" ");
            }

            if (b1.getText().equals(anObject:"1") && b2.getText().equals(anObject:"2") &&
                    b3.getText().equals(anObject:"3") && b4.getText().equals(anObject:"4") &&
                    b5.getText().equals(anObject:"5") && b6.getText().equals(anObject:"6") &&
                    b7.getText().equals(anObject:"7") && b8.getText().equals(anObject:"8") &&
                    b9.getText().equals(anObject:" ")) {
                JOptionPane.showMessageDialog(Puzzle.this, message:"You won the game!");
            }
        }
    }

    Run | Debug
    public static void main(String[] args) {

        // new Puzzle();
        LoginFrame l = new LoginFrame();


    }


    class TimerListener implements ActionListener {
        private int counter;
        private Puzzle puzzle;

        public TimerListener(int duration, Puzzle puzzle) {
            this.counter = duration;
```

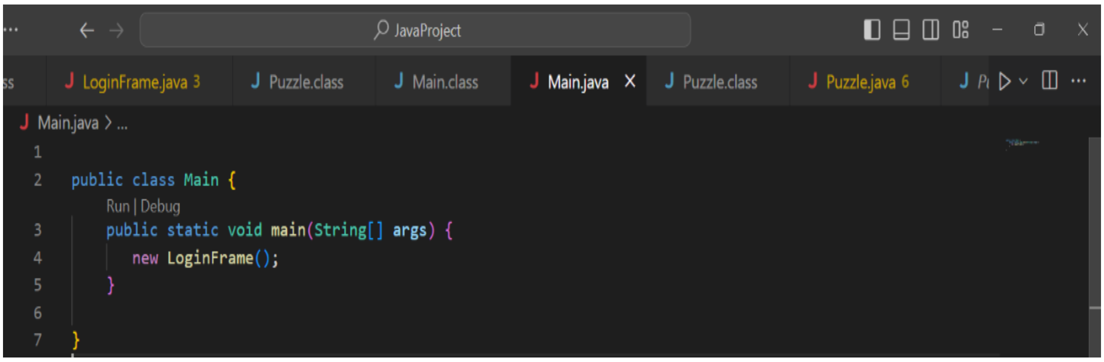

JavaProject
Puzzle.class
Main.class
Main.java
Puzzle.class
Puzzle.java 6
Puzzle$TimerListener.class
Puzzle.java > Puzzle > actionPerformed(ActionEvent)

```java
        private Puzzle puzzle;

        public TimerListener(int duration, Puzzle puzzle) {
            this.counter = duration;
            this.puzzle = puzzle;
        }

    public void actionPerformed(ActionEvent e) {
        if (counter > 0) {
            counter--;
            int minutes = counter / 60;
            int seconds = counter % 60;
            timerLabel.setText(String.format(format:"%02d:%02d", minutes, seconds));
        } else {
            // timerLabel.setText("Time's up!");
            // JOptionPane.showMessageDialog(Puzzle.this, "Time is Up");
            Object[] options = {"Exit", "Play Again"};
            int choice = JOptionPane.showOptionDialog(Puzzle.this,
                    message:"Game over! Do you want to play again?",
                    title:"Game Over",
                    JOptionPane.YES_NO_OPTION,
                    JOptionPane.QUESTION_MESSAGE,
                    icon:null,
                    options,
                    options[1]);

            if (choice == JOptionPane.YES_OPTION) {
                puzzle.dispose(); // close the JFrame
            } else if (choice == JOptionPane.NO_OPTION) {
```



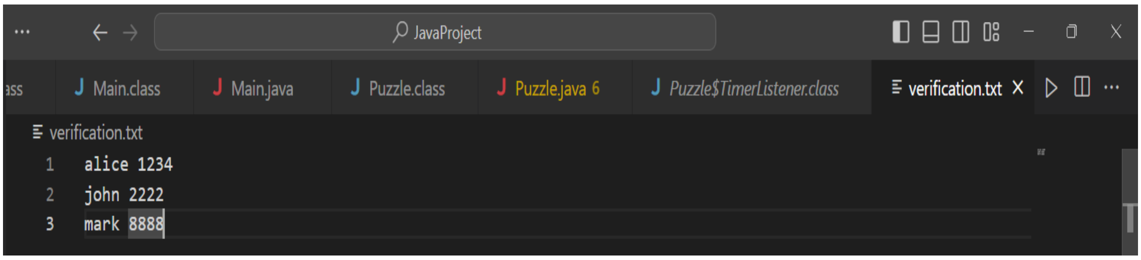
```java
                puzzle.setVisible(b:false); // hide the JFrame
                puzzle.dispose(); // dispose the JFrame
                main(args:null); // restart the application
            }
            ((Timer) e.getSource()).stop();
        }
    }
  }
}
```

3.MAIN CLASS: This segment or say module of the project will run all the interconnected modules in One go. This is the main and final module of the project which runs all other modules simultaneously.



```java

public class Main {
    public static void main(String[] args) {
        new LoginFrame();
    }

}
```

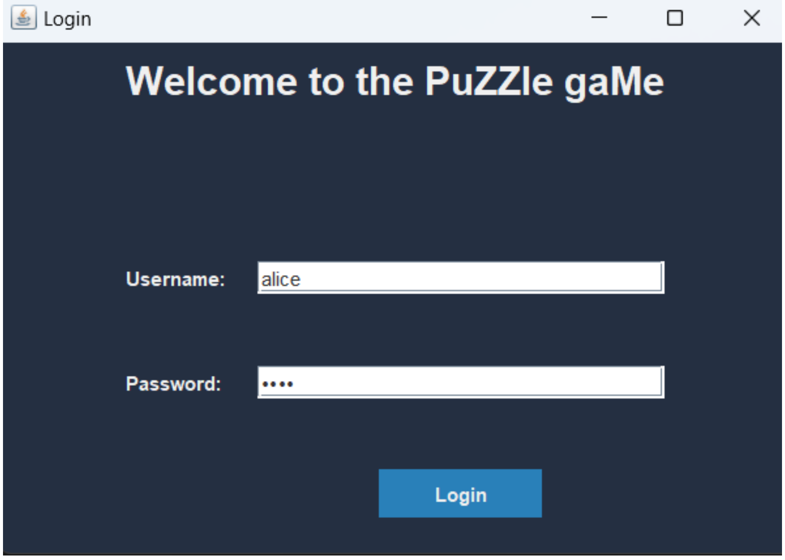
4.VERIFICATION TEXT: This contains the login credentials of some players which are valid and authorized to access the game. It includes the username and password of those players for login page.



```
alice 1234
john 2222
mark 8888
```

OUTPUT:

Login Page➔

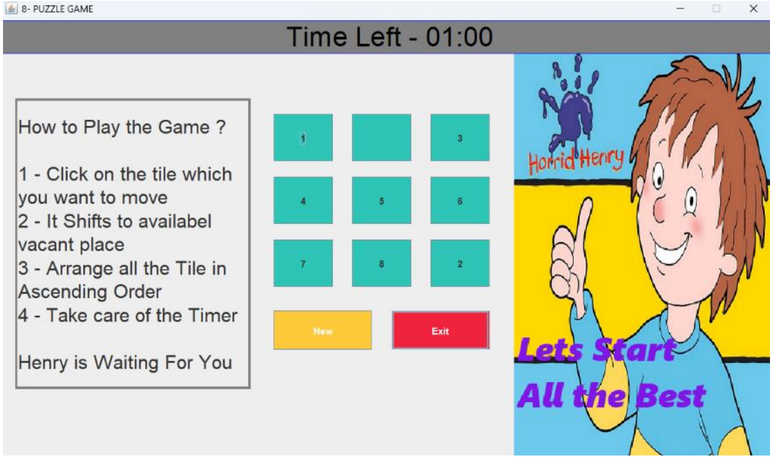


Welcome to the PuZZle gaMe

Username: alice

Password: ••••

Puzzle interface➔



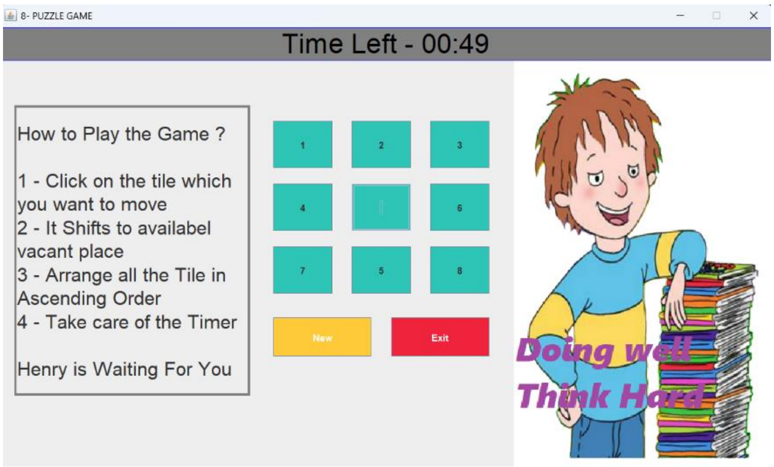
Time Left - 01:00

How to Play the Game ?

1 - Click on the tile which you want to move
2 - It Shifts to availabel vacant place
3 - Arrange all the Tile in Ascending Order
4 - Take care of the Timer

Henry is Waiting For You

1 START OF GAME



8- PUZZLE GAME

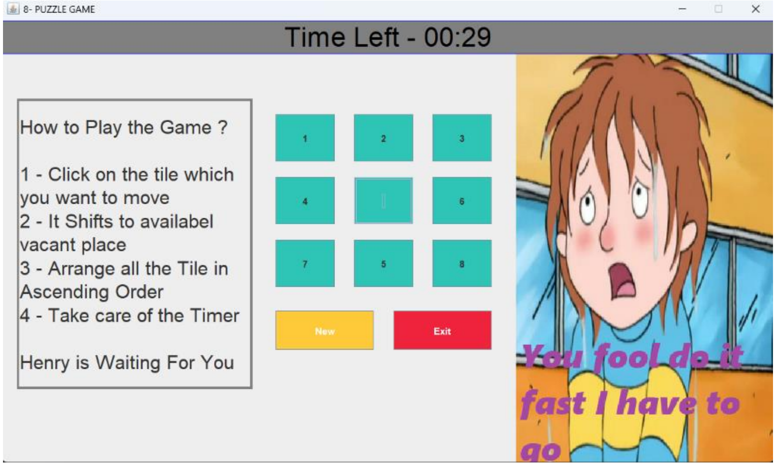
Time Left - 00:49

How to Play the Game ?

1 - Click on the tile which you want to move
2 - It Shifts to availabel vacant place
3 - Arrange all the Tile in Ascending Order
4 - Take care of the Timer

Henry is Waiting For You

2 TIMER RUNNING OUT



8- PUZZLE GAME

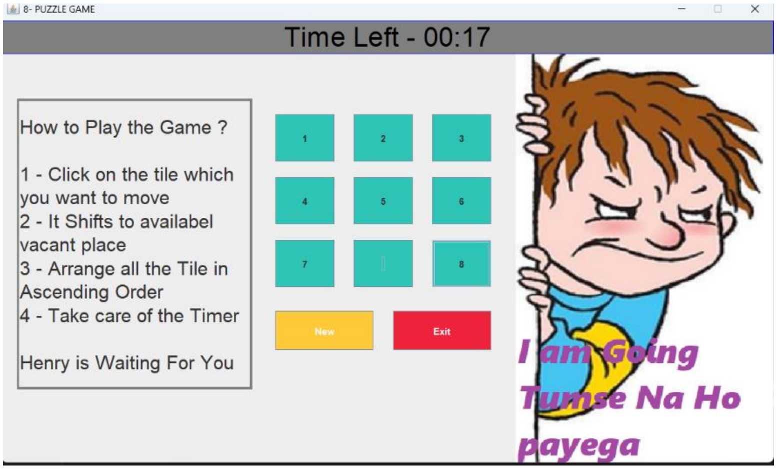
Time Left - 00:29

How to Play the Game ?

1 - Click on the tile which you want to move
2 - It Shifts to availabel vacant place
3 - Arrange all the Tile in Ascending Order
4 - Take care of the Timer

Henry is Waiting For You

3 TIMER RUNNING OUT



8- PUZZLE GAME

Time Left - 00:17

How to Play the Game ?

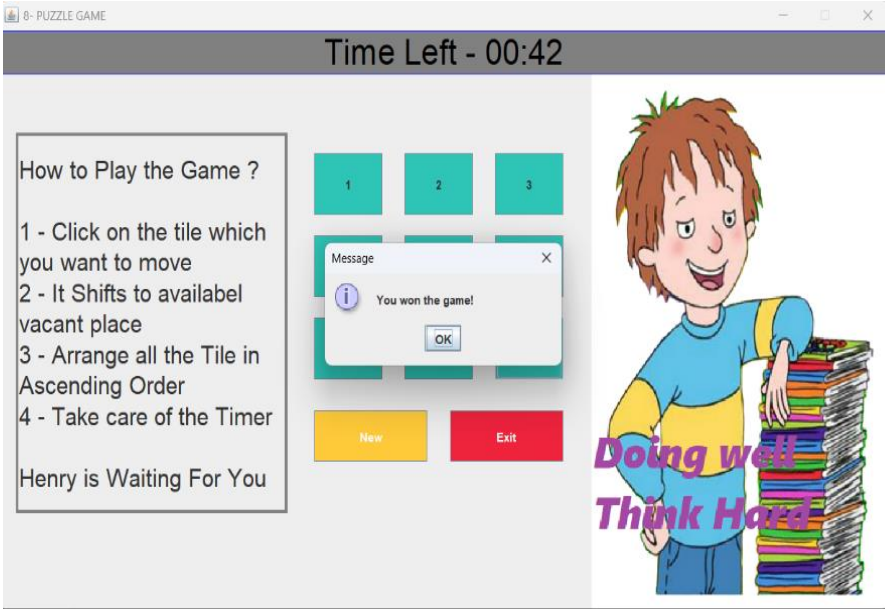
1 - Click on the tile which you want to move
2 - It Shifts to availabel vacant place
3 - Arrange all the Tile in Ascending Order
4 - Take care of the Timer

Henry is Waiting For You

4 TIMER RU

Game Results➔

First outcome: Game is won.



8- PUZZLE GAME

Time Left - 00:42

How to Play the Game ?

1 - Click on the tile which you want to move
2 - It Shifts to availabel vacant place
3 - Arrange all the Tile in Ascending Order
4 - Take care of the Timer

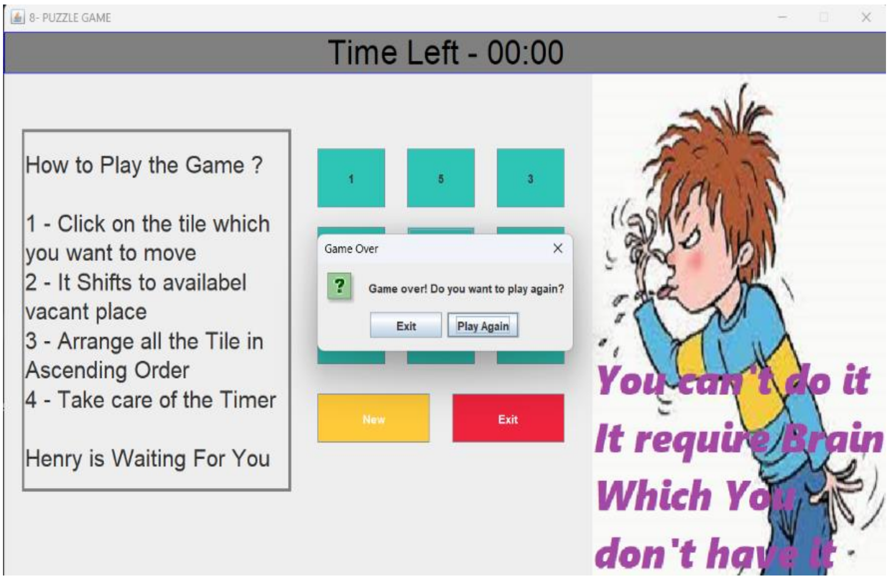
Henry is Waiting For You

Message

You won the game!

5 Game ends

Second outcome: Game is Lost.



8- PUZZLE GAME

Time Left - 00:00

How to Play the Game ?

1 - Click on the tile which you want to move
2 - It Shifts to availabel vacant place
3 - Arrange all the Tile in Ascending Order
4 - Take care of the Timer

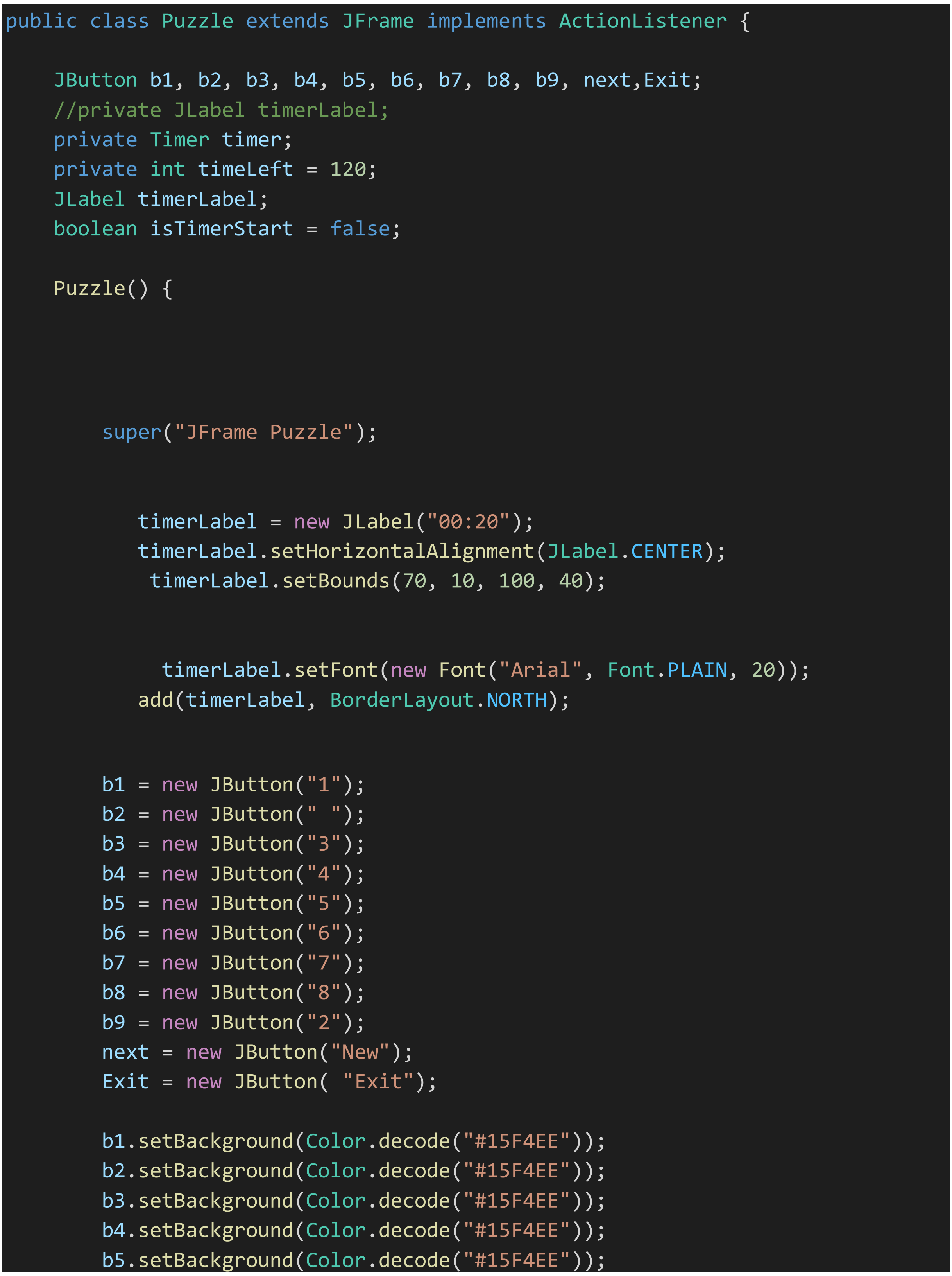
Henry is Waiting For You

Game Over

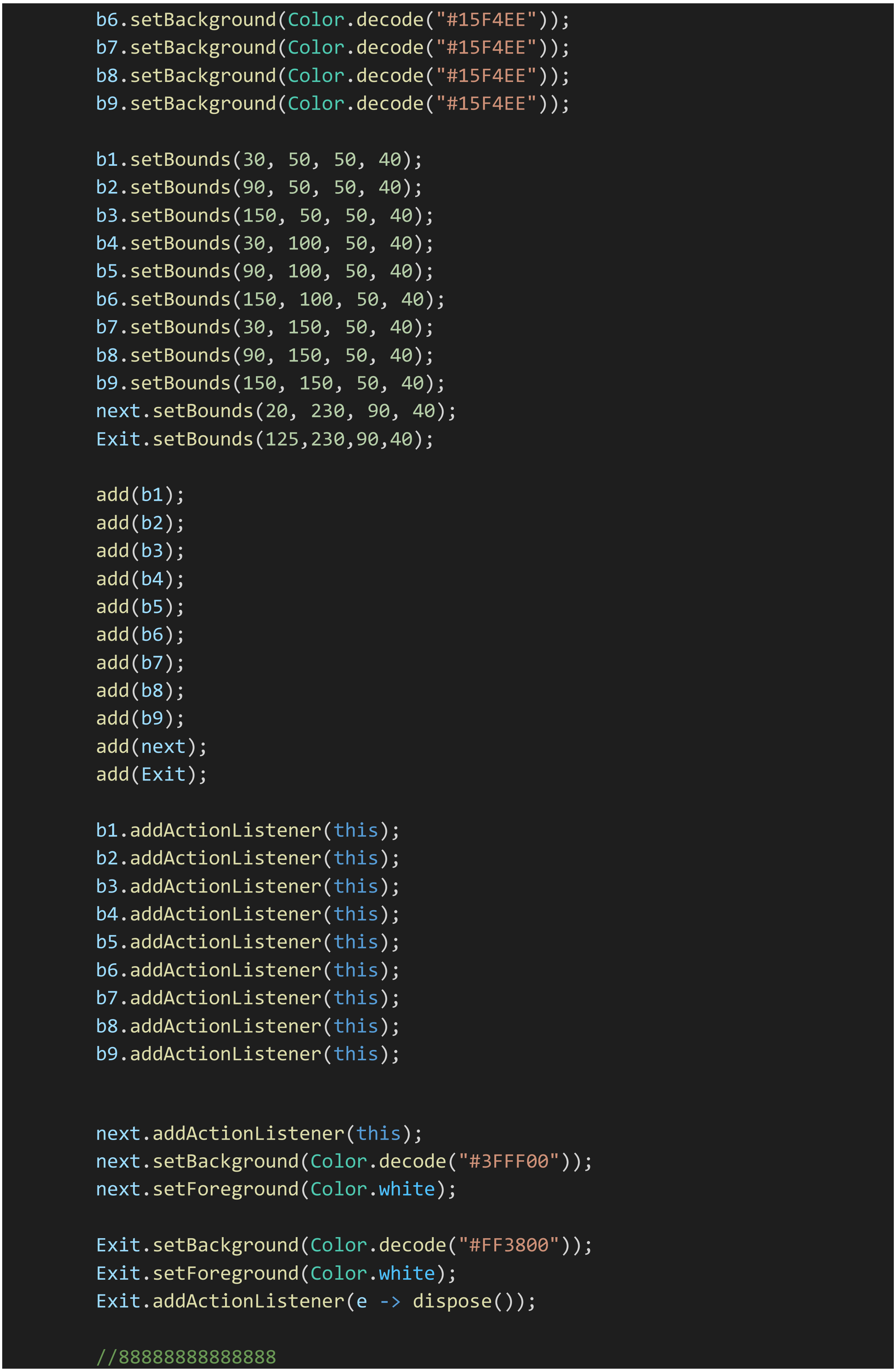
Game over! Do you want to play again?

6 Game ends

Sample code:



```java
public class Puzzle extends JFrame implements ActionListener {

    JButton b1, b2, b3, b4, b5, b6, b7, b8, b9, next,Exit;
    //private JLabel timerLabel;
    private Timer timer;
    private int timeLeft = 120;
    JLabel timerLabel;
    boolean isTimerStart = false;

    Puzzle() {




        super("JFrame Puzzle");


           timerLabel = new JLabel("00:20");
           timerLabel.setHorizontalAlignment(JLabel.CENTER);
            timerLabel.setBounds(70, 10, 100, 40);


             timerLabel.setFont(new Font("Arial", Font.PLAIN, 20));
           add(timerLabel, BorderLayout.NORTH);


        b1 = new JButton("1");
        b2 = new JButton(" ");
        b3 = new JButton("3");
        b4 = new JButton("4");
        b5 = new JButton("5");
        b6 = new JButton("6");
        b7 = new JButton("7");
        b8 = new JButton("8");
        b9 = new JButton("2");
        next = new JButton("New");
        Exit = new JButton( "Exit");

        b1.setBackground(Color.decode("#15F4EE"));
        b2.setBackground(Color.decode("#15F4EE"));
        b3.setBackground(Color.decode("#15F4EE"));
        b4.setBackground(Color.decode("#15F4EE"));
        b5.setBackground(Color.decode("#15F4EE"));
```



```
        b6.setBackground(Color.decode("#15F4EE"));
        b7.setBackground(Color.decode("#15F4EE"));
        b8.setBackground(Color.decode("#15F4EE"));
        b9.setBackground(Color.decode("#15F4EE"));

        b1.setBounds(30, 50, 50, 40);
        b2.setBounds(90, 50, 50, 40);
        b3.setBounds(150, 50, 50, 40);
        b4.setBounds(30, 100, 50, 40);
        b5.setBounds(90, 100, 50, 40);
        b6.setBounds(150, 100, 50, 40);
        b7.setBounds(30, 150, 50, 40);
        b8.setBounds(90, 150, 50, 40);
        b9.setBounds(150, 150, 50, 40);
        next.setBounds(20, 230, 90, 40);
        Exit.setBounds(125,230,90,40);

        add(b1);
        add(b2);
        add(b3);
        add(b4);
        add(b5);
        add(b6);
        add(b7);
        add(b8);
        add(b9);
        add(next);
        add(Exit);

        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        b4.addActionListener(this);
        b5.addActionListener(this);
        b6.addActionListener(this);
        b7.addActionListener(this);
        b8.addActionListener(this);
        b9.addActionListener(this);


        next.addActionListener(this);
        next.setBackground(Color.decode("#3FFF00"));
        next.setForeground(Color.white);

        Exit.setBackground(Color.decode("#FF3800"));
        Exit.setForeground(Color.white);
        Exit.addActionListener(e -> dispose());

        //88888888888888
        add(timerLabel, BorderLayout.NORTH);
        setLocationRelativeTo(null);



        // setBackground(Color.CYAN);
        getContentPane().setBackground(Color.decode("#6290c3"));


        setSize(250, 350);
        setLayout(null);
        setVisible(true);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);


    }
```

Conclusion

In conclusion, the 8-puzzle game project is an exciting project that offers several advantages and future scope for development. The game is a popular puzzle game that challenges players to rearrange the tiles on a board to form a specific pattern.

One of the primary advantages of this project is that it helps develop problem-solving skills, algorithmic thinking, and programming abilities. the 8-puzzle game project is an exciting project that offers several advantages and future scope for development. The project helps students develop problem-solving skills, algorithmic thinking, and programming abilities. The project can be extended to solve more complex puzzles, use different search algorithms, and include graphical user interfaces and scoring systems.

Future Enhancements

The future scope of this project is vast. The project can be extended to solve more complex puzzles. In terms of features, the 8-puzzle game project can include several enhancements to make it more engaging and challenging.

1➔The game can include a scoring system that rewards players for completing the puzzle quickly or with fewer moves.

2➔ The game can store the progress of each user separately in a database based on which he is given an experience-level or a rank in the game.

3➔ The game can also include different difficulty levels, which can be adjusted based on the player's skill level.

At the end, the 8-puzzle game project is an exciting project that offers several advantages and future scope for development.

-------------------------------------**THANK YOU-------------------------------------**