A MINI PROJECT REPORT
ON
PUZZLE GAME USING JAVA

Submitted in partial fulfillment of requirements for the award of 4th semester,

BACHELOR OF ENGINEERING

IN

ARTIFICIAL INTELLIGENCE AND
MACHINE LEARNING

Submitted By:
**CICIYA SEBASTIN, GAGANA D,
HEMANTH, N SUFIYA NOORAIN**
USN: 1MJ20AI010, 1MJ20AI013, 1MJ20AI016, 1MJ20AI028

Under the Guidance of
Mrs. TAMILARASI R
Professor, Department of Computer Science & Engineering

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING
MVJ COLLEGE OF ENGINEERING
BANGALORE-67
ACADEMIC YEAR 2021-22

MVJ COLLEGE OF ENGINEERING
Whitefield, Near ITPB, Bangalore-67

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

MVJ COLLEGE OF ENGINEERING
An Autonomous Institute
(Affiliated to Visvesvaraya Technological University, Belagavi
Approved By AICTE, New Delhi,
Recognized by UGC under 2(f) & 12(B)
Accredited by NBA and NAAC)

CERTIFICATE

This is to certify that the mini project entitled "**PUZZLE GAME USING JAVA**" is a Bonafide work carried out by **CICIYA SEBASTIN, GAGANA D, HEMANTH, N SUFIYA NOORAIN (1MJ20AI010,1MJ20AI013, 1MJ20AI016, 1MJ20AI028),** bonafide students of MVJ College of Engineering in partial fulfillment for the award of degree of Bachelor of Engineering in Artificial Intelligence and Machine Learning during the year 2021-22. It is certified that all the corrections/suggestions indicated for Internal Assessment have been incorporated in the Report. The Mini Project Report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said degree.

_____   _____
Signature of the Guide   **Signature of the HOD**
(Prof. Tamilarasi. R)   **(Prof. Tamilarasi. R)**

MVJ COLLEGE OF ENGINEERING
Whitefield, Near ITPB, Bangalore-67

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

<u>DECLARATION</u>

We, **CICIYA SEBASTIN, GAGANA D, HEMANTH, N SUFIYA NOORAIN** hereby declare that the entire work titled **"PUZZLE GAME USING JAVA"** embodied in this project report has been carried out by us during the 4th semester of BE degree at MVJCE, Bangalore under the esteemed guidance of **Prof. Tamilarasi. R,** Head of the Department of CSE, MVJCE. The work embodied in this dissertation work is original and it has not been submitted in part of full for any other degree in any University.

CICIYA SEBASTIN	1MJ20AI010	————————
GAGANA D	1MJ20AI013	————————
HEMANTH	1MJ20AI016	————————
N SUFIYA NOORAIN	1MJ20AI028	————————

Place:

Date:

ABSTRACT

This project aims to bring the fun and simplicity of Puzzle game with some new features. It will include computer controlled intelligent opponents whose aim will be to challenge the human players. It will also have the multiplayer feature that will allow more than one players to play the game over a network.

This project explores a new dimension in the traditional Puzzle game to make it more interesting and challenging. The simplicity of this game makes it an ideal candidate for a minor project as we can focus on advanced topics like multiplayer functionality and implementation of computer controlled intelligent opponents.

This game aims to change the way people think of traditional Puzzle game. It will offer the experience of commercial multilayer games to the player retaining the simplicity of traditional Puzzle game.

The major objectives of this project are:

● Create a Puzzle game that will have all the functionality of traditional Puzzle games.

● Introduce multilayer functionality in the game that will allow several players to play a game simultaneously. It should be able to give the experience of a real time multiplayer game to the players.

● Introduce computer controlled intelligent opponent (unique feature of this game) to make the game more challenging and interesting.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success.

We are thankful to the Management of **MVJ College of Engineering Bangalore** for their continuous support and encouragement for carrying out the project work.

I express my sincere gratitude to our Principal **Dr.P.Mahabaleshwarappa**, & Vice Principal **Dr.M.Brindha,** MVJ College of Engineering for providing facilities.

I express my sincere gratitude to our **Dr. M A Lourdu Antony Raj** registrar and controller for examinations, MVJCE, Bengaluru for persistent guidance.

I wish to place on record my grateful thanks to **Prof. Tamilarasi. R**, Head of the Department of Computer Science and Engineering, MVJ College of Engineering, Bangalore for providing encouragement and guidance.

I consider it a privilege and honour to express my sincere gratitude to my guide **Mrs. Tamilarasi. R, Professor,** Department of Computer Science & Engineering for their valuable guidance throughout the tenure of this mini project work and whose support and encouragement made this work possible.

It's also a great pleasure to express our deepest gratitude to all faculty members of our department for their cooperation and constructive criticism offered, which helped us a lot during our project work.

Finally, we would like to thank all our family members and friends whose encouragement and support was invaluable.

Thank you.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Playing games is fun and exciting. It gives us relief from stress and unwinds from our stressful work. Many of us spend our free time or others that use most of their time in playing and exploring new games. Today, with the raped development of technology we have, games that are rising up together with it. Nowadays with technology we have many games that are developed for computers specifically for windows. With the high technology equipped with these computer games become robust and attract many people to buy or have this gadget for them to experience what's inside it which makes it a trend for the new generation of gadget. Puzzle game is a computer action game, whose goal is to arrange the alphabets in the correct order.

It has been around since the earliest days of home computing and has re-emerged in recent years on mobile phones. It isn't the world's greatest game, but it does give you an idea of what you can achieve with a simple python program and perhaps the basis by which to extend the principles and create more interesting games on your own. To play the Puzzle game, use up arrow for up, down arrow for down, left arrow for left and right arrow for right. Or you can even click on the alphabet to direct it to the unoccupied box. The alphabets will be jumbled. And the aim or goal of the game is to arrange the alphabets in the correct order. Once you enter the correct order the game ends.

CHAPTER 2

CODE DESCRIPTION

The core focus of our project was to determine which algorithms would be more effective in a hard real-time environment. The domain in this case is the Puzzle Game, which will, in turn, attempt to identify an, or even the, algorithm that can not only play the game but compete with human players. The Puzzle Game is a classic arcade style game where it is a single-player game but the focus is to solve the puzzle at the least possible time thus competing with yourself and others.

To play the game one can use the mouse to click on the alphabets you need to move. The alphabet will move only when there is an empty space near the selected alphabet. The alphabets will be jumbled. And the aim or goal of the game is to arrange the alphabets in the correct order. Once you enter the correct order the game ends.

The domain provides a very interesting problem given that the alphabet always moves after a given timing delay till the puzzle is solved. In this problem, we want to write a game where there is a graphical representation of a Puzzle. There also needs to be boxes which will contain the graphical representation of the alphabets in bold big capital letters. The alphabets must move when a mouse action is applied on it. To write this program we are going to need: 1. A way of representing the block of puzzle. 2. A way to display the alphabets in the boxes, 3. A way for our instructions to solve the Puzzle, 4. A way to know when we've completed the game and that we've won the game. Our system is going to involve working with both hardware and software, and so we will need to understand what we have available in hardware that can assist us. If we build our software so that the Puzzle is controlled by directional arrows on the keyboard. Now that understand how our hardware will work in the design of our system, let's move on to starting the design of our software system

Swing is a Java Foundation Classes [JFC] library and an extension of the Abstract Window Toolkit [AWT]. Swing offers much-improved functionality over AWT, new components, expanded components features, excellent event handling with drag and drop support.

Swing has about four times the number of User Interface [UI] components as AWT and is part of the standard Java distribution. By today's application GUI requirements, AWT is a limited

implementation, not quite capable of providing the components required for developing complex GUI's required in modern commercial applications. The AWT component set has quite a few bugs and really does take up a lot of system resources when compared to equivalent Swing resources. Netscape introduced its Internet Foundation Classes [IFC] library for use with Java. Its Classes became very popular with programmers creating GUI's for commercial applications.

Swing is a Set Of API (API- Set Of Classes and Interfaces)

Swing is Provided to Design a Graphical User Interfaces

Swing is an Extension library to the AWT (Abstract Window Toolkit)

Includes New and improved Components that have been enhancing the looks and Functionality of GUI's

Swing can be used to build(Develop) The Standalone swing GUI Apps Also as Servlets And Applets

It Employs model/view design architecture

Swing is more portable and more flexible than AWT, The Swing is built on top of the AWT

Swing is Entirely written in Java

Java Swing Components are Platform-independent And The Swing Components are lightweight

Swing Supports Pluggable look and feels And Swing provides more powerful components such as tables, lists, Scrollpanes, Colourchooser, tabbedpane, etc

Further Swing Follows MVC

JFrame is a top-level container that provides a window on the screen. A frame is actually a base window on which other components rely, namely the menu bar, panels, labels, text fields, buttons, etc. Almost every other Swing application starts with the JFrame window. Unlike a frame, JFrame has the option to hide or close the window with the help of the method setDefaultCloseOperation(int).

JPanel, a part of the Java Swing package, is a container that can store a group of components. The main task of JPanel is to organize components, various layouts can be set in JPanel which provide better organization of components, however, it does not have a title bar.

Java AWT (Abstract Window Toolkit) is an API to develop Graphical User Interface (GUI) or windows-based applications in Java. Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system. AWT is heavy weight i.e. its components are using the resources of underlying operating system (OS).

Java AWT calls the native platform calls the native platform (operating systems) subroutine for creating API components like TextField, ChechBox, button, etc.

For example, an AWT GUI with components like TextField, label and button will have different look and feel for the different platforms like Windows, MAC OS, and Unix. The reason for this is the platforms have different view for their native components and AWT directly calls the native subroutine that creates those components.

In simple words, an AWT application will look like a windows application in Windows OS whereas it will look like a Mac application in the MAC OS.

Java adapter classes provide the default implementation of listener interfaces. If you inherit the adapter class, you will not be forced to provide the implementation of all the methods of listener interfaces. So it saves code.

An abstract adapter class for receiving keyboard events. The methods in this class are empty. This class exists as convenience for creating listener objects.
Extend this class to create a KeyEvent listener and override the methods for the events of interest. (If you implement the KeyListener interface, you have to define all of the methods in it. This abstract class defines null methods for them all, so you can only have to define methods for events you care about.)

Create a listener object using the extended class and then register it with a component using the component's addKeyListener method. When a key is pressed, released, or typed, the relevant method in the listener object is invoked, and the KeyEvent is passed to it.

The SCREEN_WIDTH and SCREEN_HEIGHT constants determine the size of the board. The UNIT_SIZE is the size of the apple and the dot of the Puzzle.The DELAY constant determines the speed of the game.

Intx[] and inty[],These two arrays store the x and y coordinates of all joints of a Puzzle.

In the startGame() method we create the Puzzle, randomly locate an apple on the board, and start the timer.

If the apple collides with the head, we increase the number of joints of the Puzzle. We call the newApple() method which randomly positions a new apple object.

CHAPTER 3

IMPLEMENTATION

5.1 Introduction to software used:

Description

To run this project you must have installed Eclipse is an integrated development environment (IDE) used in computer programming.[5] It contains a base workspace and an extensible plug-in system for customizing the environment. It is the second-most-popular IDE for Java development, and, until 2016, was the most popular.[6] Eclipse is written mostly in Java and its primary use is for developing Java applications,[7] but it may also be used to develop applications in other programming languages via plug-ins, including Ada, ABAP, C, C++, C#, Clojure, COBOL, D, Erlang, Fortran, Groovy, Haskell, JavaScript, Julia,[8] Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Rust, Scala, and Scheme. It can also be used to develop documents with LaTeX (via a TeXlipse plug-in) and packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM VisualAge.[9] The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Since the introduction of the OSGi implementation (Equinox) in version 3 of Eclipse, plug-ins can be plugged-stopped dynamically and are termed (OSGI) bundles.[10]

Eclipse software development kit (SDK) is free and open-source software, released under the terms of the Eclipse Public License, although it is incompatible with the GNU General Public License.[11] It was one of the first IDEs to run under GNU Classpath and it runs without problems under IcedTea.

5.2 Source Code:

```java
import java.awt.*;
import java.awt.event.*;
import javax.swing.JOptionPane;


public class AlphabetGame extends Frame implements ActionListener {
        Button b1, b2, b3, b4, b5, b6, b7, b8, b9;

        AlphabetGame() {
                super("Puzzle - SimpliGame");
                b1 = new Button("A");
                b1.setBounds(50, 100, 40, 40);
                b2 = new Button("B");
                b2.setBounds(100, 100, 40, 40);
                b3 = new Button("C");
                b3.setBounds(150, 100, 40, 40);
                b4 = new Button("D");
                b4.setBounds(50, 150, 40, 40);
                b5 = new Button("E");
                b5.setBounds(100, 150, 40, 40);
                b6 = new Button("F");
                b6.setBounds(150, 150, 40, 40);
                b7 = new Button("G");
                b7.setBounds(50, 200, 40, 40);
                b8 = new Button("");
                b8.setBounds(100, 200, 40, 40);
                b9 = new Button("H");
                b9.setBounds(150, 200, 40, 40);

                b1.addActionListener(this);
                b2.addActionListener(this);
                b3.addActionListener(this);
                b4.addActionListener(this);
```

```java
        b5.addActionListener(this);
        b6.addActionListener(this);
        b7.addActionListener(this);
        b8.addActionListener(this);
        b9.addActionListener(this);

        add(b1);
        add(b2);
        add(b3);
        add(b4);
        add(b5);
        add(b6);
        add(b7);
        add(b8);
        add(b9);
        setSize(400, 400);
        setLayout(null);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == b1) {
            String label = b1.getLabel();
            if (b2.getLabel().equals("")) {
                b2.setLabel(label);
                b1.setLabel("");
            }
            if (b4.getLabel().equals("")) {
                b4.setLabel(label);
                b1.setLabel("");
            }
        }
        if (e.getSource() == b2) {
            String label = b2.getLabel();
```

```java
                if (b1.getLabel().equals("")) {
                        b1.setLabel(label);
                        b2.setLabel("");
                }
                if (b3.getLabel().equals("")) {
                        b3.setLabel(label);
                        b2.setLabel("");
                }
                if (b5.getLabel().equals("")) {
                        b5.setLabel(label);
                        b2.setLabel("");
                }
        }
        if (e.getSource() == b3) {
                String label = b3.getLabel();
                if (b2.getLabel().equals("")) {
                        b2.setLabel(label);
                        b3.setLabel("");
                }
                if (b6.getLabel().equals("")) {
                        b6.setLabel(label);
                        b3.setLabel("");
                }
        }
        if (e.getSource() == b4) {
                String label = b4.getLabel();
                if (b1.getLabel().equals("")) {
                        b1.setLabel(label);
                        b4.setLabel("");
                }
                if (b7.getLabel().equals("")) {
                        b7.setLabel(label);
                        b4.setLabel("");
                }
```

```java
                if (b5.getLabel().equals("")) {
                        b5.setLabel(label);
                        b4.setLabel("");
                }
        }
        if (e.getSource() == b5) {
                String label = b5.getLabel();
                if (b2.getLabel().equals("")) {
                        b2.setLabel(label);
                        b5.setLabel("");
                }
                if (b6.getLabel().equals("")) {
                        b6.setLabel(label);
                        b5.setLabel("");
                }
                if (b4.getLabel().equals("")) {
                        b4.setLabel(label);
                        b5.setLabel("");
                }
                if (b8.getLabel().equals("")) {
                        b8.setLabel(label);
                        b5.setLabel("");
                }
        }
        if (e.getSource() == b6) {
                String label = b6.getLabel();
                if (b9.getLabel().equals("")) {
                        b9.setLabel(label);
                        b6.setLabel("");
                }
                if (b3.getLabel().equals("")) {
                        b3.setLabel(label);
                        b6.setLabel("");
                }
```

```java
                if (b5.getLabel().equals("")) {
                        b5.setLabel(label);
                        b6.setLabel("");
                }
        }
        if (e.getSource() == b7) {
                String label = b7.getLabel();
                if (b4.getLabel().equals("")) {
                        b4.setLabel(label);
                        b7.setLabel("");
                }
                if (b8.getLabel().equals("")) {
                        b8.setLabel(label);
                        b7.setLabel("");
                }
        }
        if (e.getSource() == b8) {
                String label = b8.getLabel();
                if (b9.getLabel().equals("")) {
                        b9.setLabel(label);
                        b8.setLabel("");
                }
                if (b7.getLabel().equals("")) {
                        b7.setLabel(label);
                        b8.setLabel("");
                }
                if (b5.getLabel().equals("")) {
                        b5.setLabel(label);
                        b8.setLabel("");
                }
        }
        if (e.getSource() == b9) {
                String label = b9.getLabel();
                if (b6.getLabel().equals("")) {
```
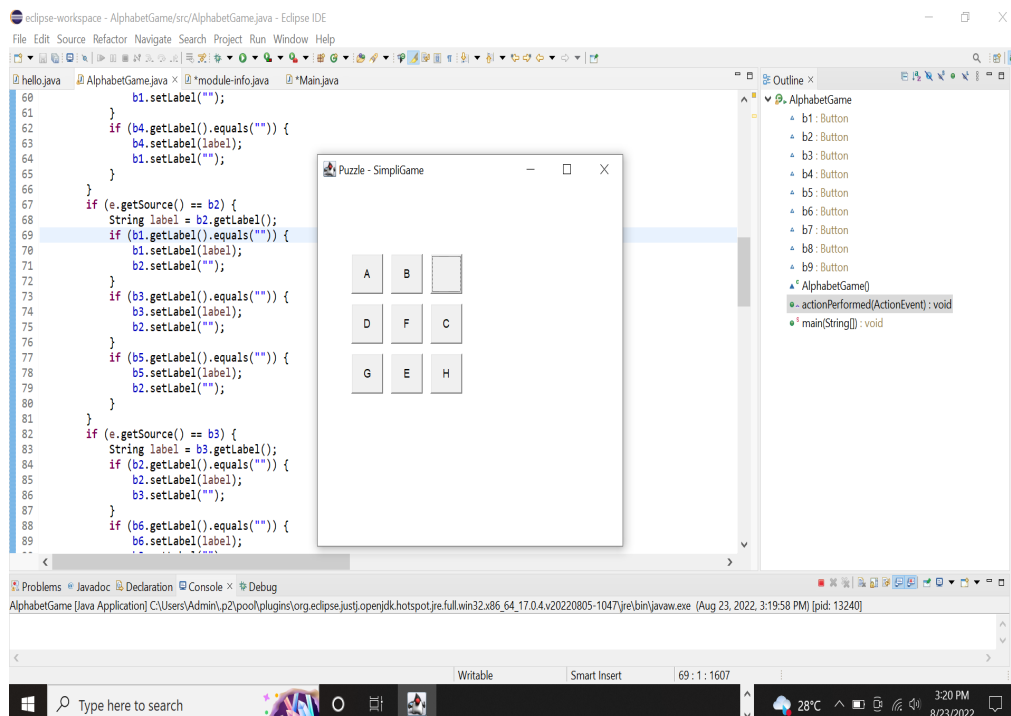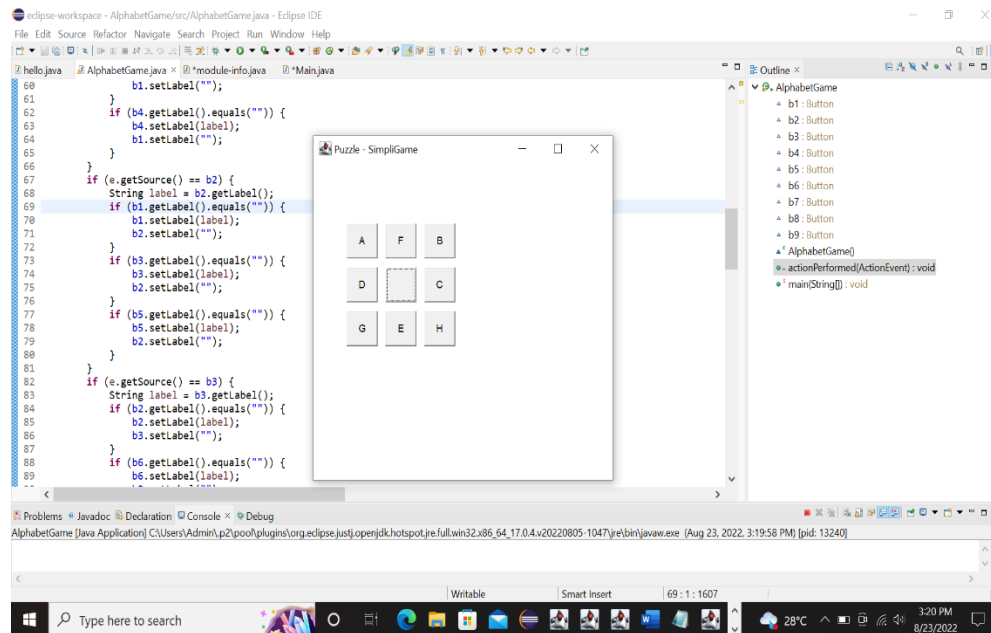
```java
                    b6.setLabel(label);
                    b9.setLabel("");
                }
                if (b8.getLabel().equals("")) {
                    b8.setLabel(label);
                    b9.setLabel("");
                }
            }

        if (b1.getLabel().equals("A") && b2.getLabel().equals("B") &&
b3.getLabel().equals("C") && b4.getLabel().equals("D") && b5.getLabel().equals("E")
&& b6.getLabel().equals("F") && b7.getLabel().equals("G") &&
b8.getLabel().equals("H") && b9.getLabel().equals("")) {
                JOptionPane.showMessageDialog(this, "Congratulations! You
won.");
            }
        }

    public static void main(String[] args) {
        new AlphabetGame();
    }
}
```
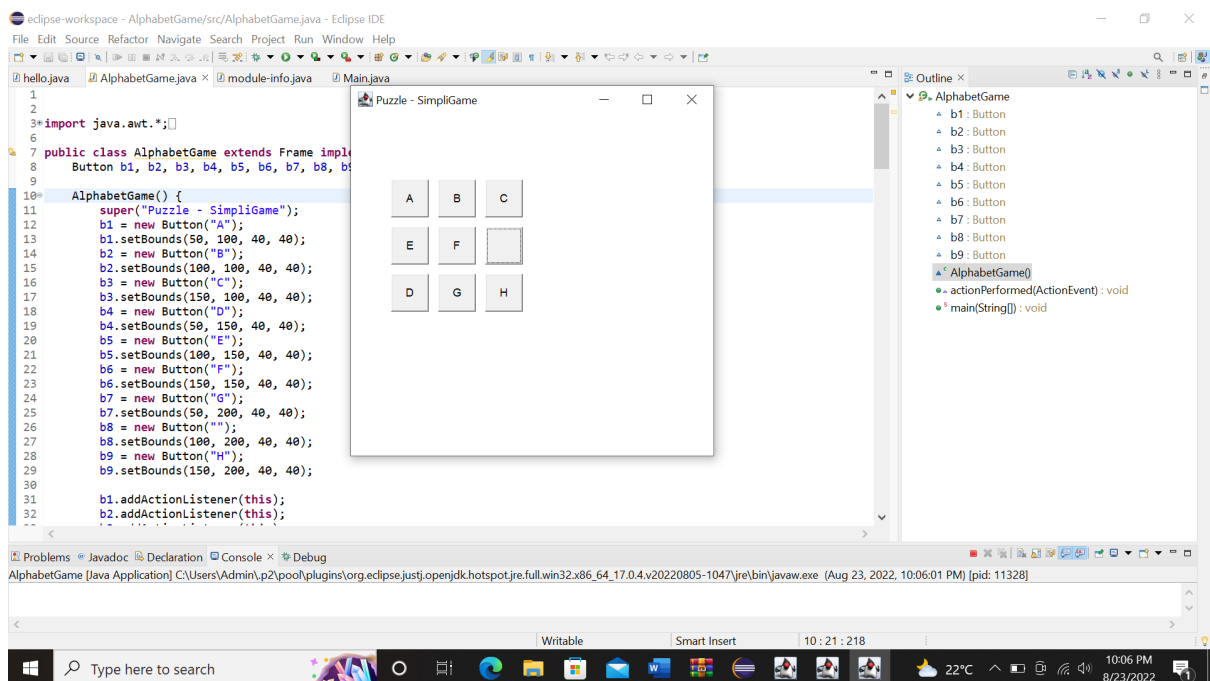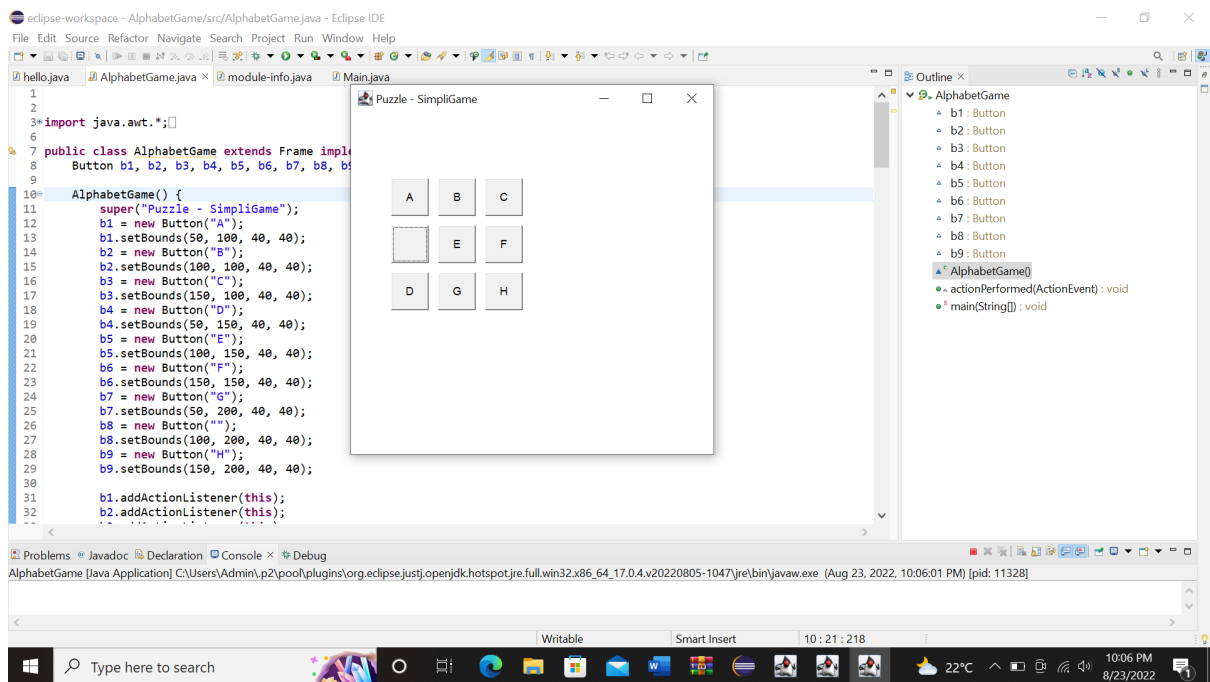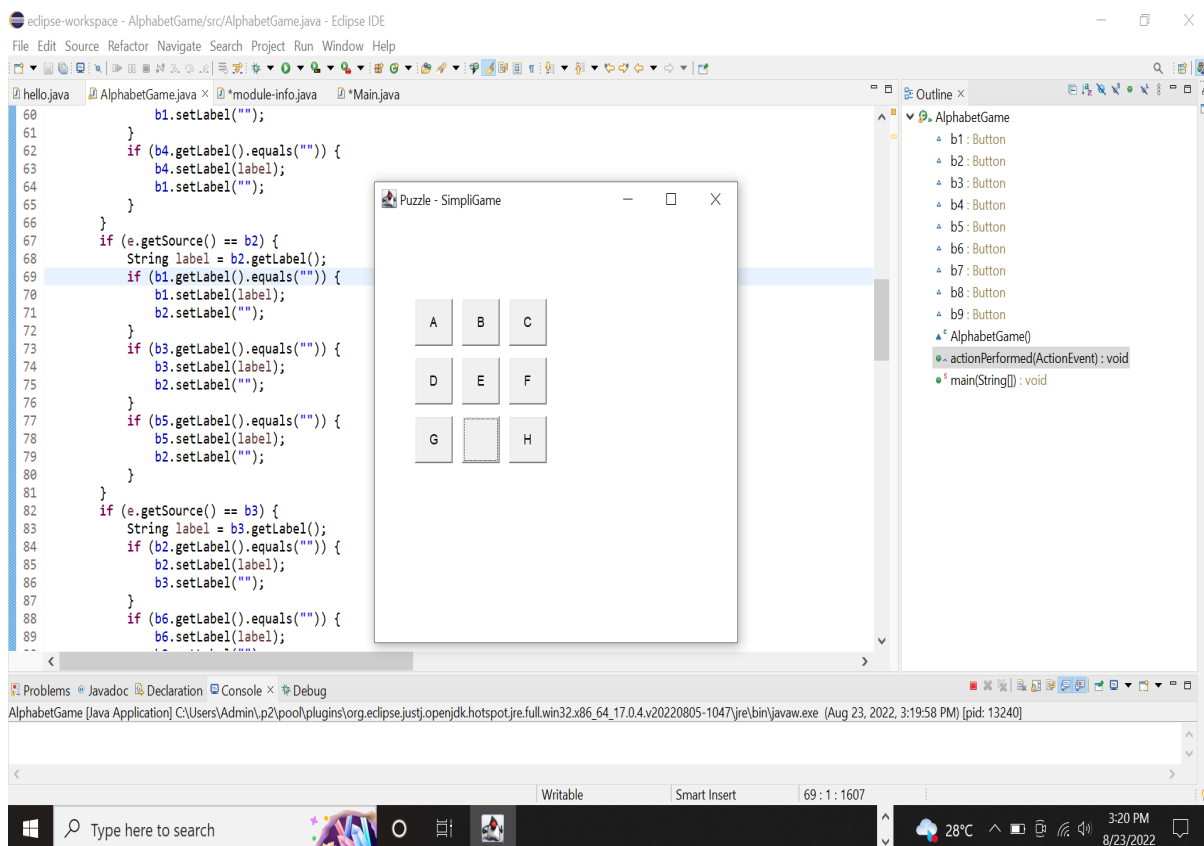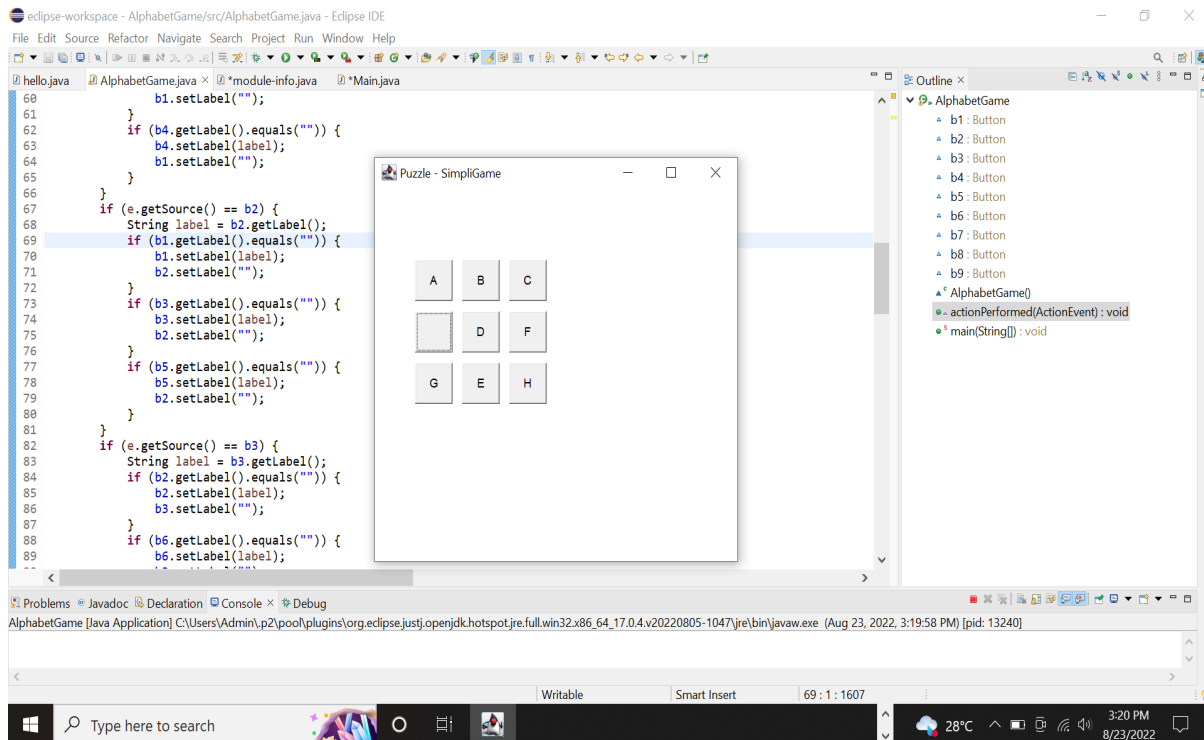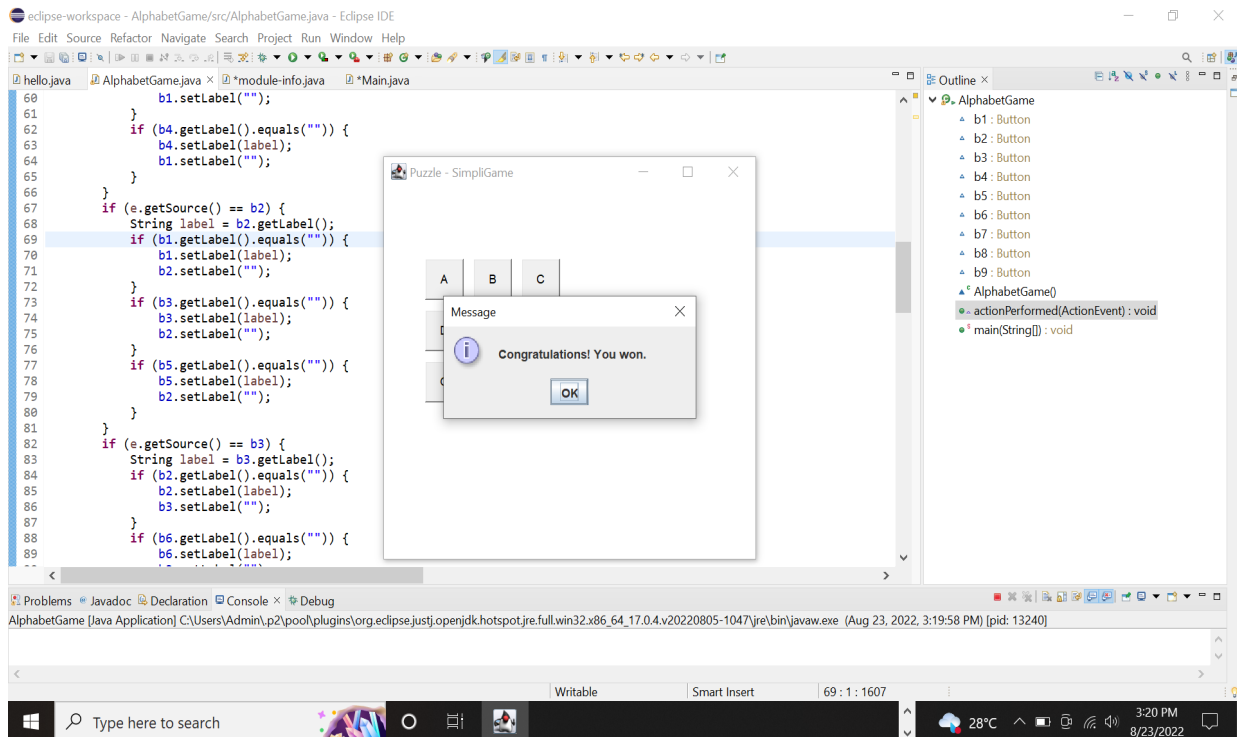
CHAPTER 4

SCREENSHOTS

Fig: 4.1 Screenshots of PUZZLE GAME using JAVA

CHAPTER 5
CONCLUSION & FUTURE SCOPE

The project in java programming of Puzzle Game is a simple console application with very simple graphics. In this project, you can play the popular "Puzzle Game" just like you played it elsewhere. You have to use the up, down, right, or left arrows to move the alphabets. You can shuffle the alphabets as you wish, and you can move left or right accordingly to arrange the alphabets in order. Once the alphabets are in correct order, the game ends with a "Congratulations, You Won" tab.

It isn't the world's greatest game, but it does give you an idea of what you can achieve with relatively simple java programming, and perhaps the basis by which to extend the principles and create more interesting games on your own.

This project gives us more thrilling, frustrating and also gives us more pleasure. It helps us in many sectors like- planning, designing, developing, managing, programming skill, socket programming and so on.

REFERENCES

➢ During the course of this project reference to the following materials were made.

1. https://www.javatpoint.com/xampp

2. https://www.geeksforgeeks.org/dbms/