Plutonium-0.3 SNL

ChopStick (Hand Game)

[GENRE: COMBINATORIAL GAMES]

Project Report

Developed By Sahill N Lalani

Void Matrix[:] Studio

NOTE: I've made this all by myself nobody has helped or guided.

(Degree)

Bachelor Of Computer Applications (BCA)

Supervised By Vilas Vadher

Principal: dr. Jigar Raval

Submitted To-

Shree k.m. savjani & smt. k.k. savjani college Veraval

Affiliated With

October 2022(BCA Sem 5)

Certificate Of Originality

This is to certify that the project report entitled **ChopStick(Hand Game)** submitted to **km & kk savjani college affiliated with BKNMU** in partial fulfilment of the requirement for the award of the degree of **Bachelor Of Computer Application(BCA)** , is an authentic and original work carried out by **Sahil N Lalani** ,SPU ID-**2020001747**.

This project is a genuine work done by the student and has not been submitted to any other university/institute for the fulfilment of the requirement of any course of study.

Dilas vadher

Supervisor

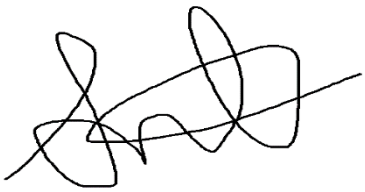
Statement Of Originality

I, the undersigned

Name: Sahil Naushadali Lalani

SPU ID: 2020001747

Certify that this project was made all by myself and isn't copied or plagiarized from any people if it would be proven that this project was plagiarized or copied from other's then I will be ready to accept a sanction



Sahil N Lalani
Student

Acknowledgments

First of all, I would like to thank my parents who always support me and then I would like to thank my friends Ghanshyam and Hardik and one of my college professors Bhavik sir for providing me valuable feedback during my game development, and thanks to this college so that I got a chance to create the game like this.

Preface

The aim of this project is to improve the reasoning power of people it is a game that is played in physically but by the implementation of my logic now it can be played on computer as well. I am inspired by Mojang studio's "MINECARFT" game.

Chopsticks is a hand game for two or more players, in which players extend a number of fingers from each hand and transfer those scores by taking turns to tap one hand against another. Chopsticks is an example of a combinatorial game, and is solved in the sense that with perfect play, an optimal strategy from any point is known.

Purely java programming is used to make this game, for the front-end AWT, swing and window-builder, for back-end mysql, JAVA mail service - apache common mails and to create

an exe file I have used Launch4j, eclipse IDE for development and db4free.net website for remote MySQL server.

Abstract

I am fond of playing games I never had a console game like xbox or play station or Nintendo but I definitely have been playing android games although I have played on console at my friend's, this chopstick game I started playing in my 11 the grade, this is a game played physically through our hands.

This my very first game that I have made and this will be very helpful for my future steps into game development.

I am so glad to introduce a physical game that now can be played on computer which I have implemented by myself.

I have used java programming language for making this game, the swing utility and window builder is used for designing and for game logic core java and some advanced java concepts like java mail API and jdbc. For sprites and some images I have used flaticon website and bing browser respectively and for editing them MS picture manager.

Contents

Section 1: Introduction

This chapter covers the project proposal and feasibility of the proposal along with background study, product and business perspective, the scopes and some preliminary idea of our game.

1.1 Gaming in the Field of Software Engineering

In the fast growing field of software engineering and development and even more rapidly growing sector of game development the future is hard to predict. I am working with this game as my software project is a 3 credit course and as part of our degree we choose this type of work for doing better with development cycle, development period, graphics, scripting, adopting new technology, animation.

In general software project is a project focusing on the creation of software. Consequently,

Success can be measured by taking a look at the resulting software.

In a game project, the product is a game. But and here comes the point: A game is much more than just its software. It has to provide content to become enjoyable. Just like a web server. Without content the server is useless, and the quality cannot be measured. This has an important effect on the game project as a whole. The software part of the project is not the only one, and it must be considered in connection to all other parts: The environment of the game, the story, characters, game plays, the artwork, and so on.

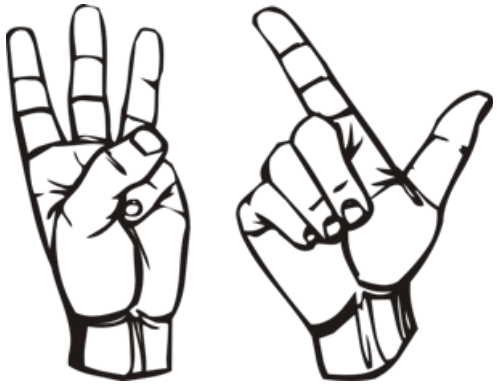
1.2 Background of this Project

Background is a set of events invented for a plot, presented as preceding and leading up to that plot. It is a literary device of a narrative history all chronologically earlier than the narrative of primary interest. In our project it's a single and dual player strategy game emphasizing logical thinking and planning. They often stress resource and time management, Tactical organization and

execution are necessary, and the game creators usually place the decision-making skills and delivery of commands in the player's hands.

1.3 Game overview

Chopsticks is a game of strategy as well as basic math. It has roots in Japan and can also called Finger Chess, Swords, Split, Magic Fingers, Chinese Fingers, Cherries, Sticks, and Twiddly Dinks. Though there are many variations of rules and different names, the overall theory and spirit of the game remains the same. This game is not the widely known beginner's piano song though it shares the same name.



Rules

This official set of rules is called rollover where five fingers are subtracted should a hand's sum exceeds 5 as described below.

1. A hand is live if it has at least one finger, and this is indicated by raising at least one finger. If a hand has zero fingers, the hand is dead, and this is indicated by raising zero fingers (i.e. a closed fist).
2. If any hand of any player reaches exactly five fingers, then the hand is dead.
3. Each player begins with one finger raised on each hand. After the first player turns proceed clockwise.
4. On a player's turn, they must either attack or split. There are two types of splits, transfers and divisions.
5. To attack, a player uses one of their live hands to strike an opponent's live hand. The number of fingers on the opponent's struck hand will increase by the number of fingers on the hand used to strike.

6. To transfer, a player strikes their own two hands together, and transfers raised fingers from one hand to the other as desired. However, a player cannot transfer fingers to make a hand have more than 4 fingers.
7. If a player has a dead hand, the player can divide the fingers between the other hand and the dead hand by transferring fingers from the other hand to the dead hand. However, players are required to attack at least once during the game.
8. A player with two dead hands is eliminated from the game.
9. A player wins once all opponents are eliminated.
10. If you go over 5 you subtract the sum of all of the numbers by 5

1.4 About the Game Project

In this the players will get two buttons for controlling their hands' fingers according to various contexts, players can also create account for saving status of work and delete it whenever they want. Players can also play without creating account as a guest user player will receive an email after creating the Void Matrix[:] account.

After they will get to the main frame where they will be seeing several options which are “play against computer”, ”player vs player”, “settings”, “how to play”, “profile”, “about game”, “feedback” and “sign out”.

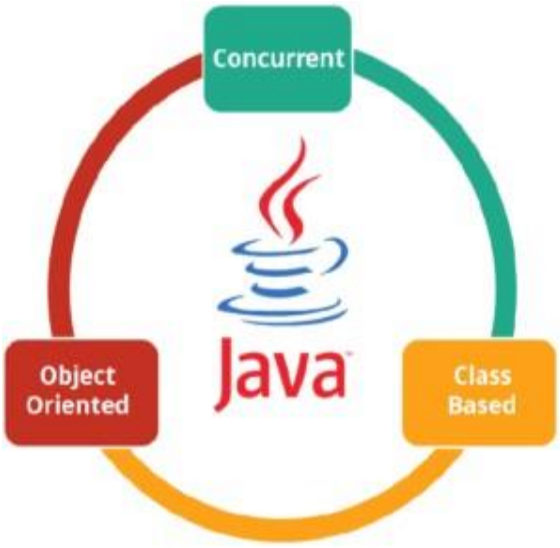
Section 2: Technical Overview

2.1 Introduction to JAVA

The language used for developing this project is JAVA

What is JAVA?

It is an object-oriented language similar to C++, but with advanced and simplified features. Java is free to access and can run on all platforms.



In simple words, it is a computing platform where you can develop applications.

What Can JAVA Do?

Development Tools: The development tools provide everything you'll need for compiling, running, monitoring, debugging, and documenting your applications. As a new developer, the main tools you'll be using are the javac compiler, the java launcher, and the javadoc documentation tool.

Application Programming Interface (API): The API provides the core functionality of the Java programming language. It offers a wide array of useful classes ready for use in your own applications

Deployment Technologies: The JDK software provides standard mechanisms such as the Java Web Start software and Java Plug-In software for deploying your applications to end users.

User Interface Toolkits: The JavaFX, Swing, and Java 2D toolkits make it possible to create sophisticated Graphical User Interfaces (GUIs).

Integration Libraries: Integration libraries such as the Java IDL API, JDBC

API, Java Naming and Directory Interface (JNDI) API, Java RMI, and Java Remote Method Invocation over Internet Inter-ORB Protocol Technology (Java RMI-IIOP Technology) enable database access and manipulation of remote objects.

□ Why JAVA?

- Object Oriented

In Java, everything is an Object. Java can be easily extended since it is based on the Object model.

- Platform Independent

Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platformindependent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.

- Simple

Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.

- Secure

With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.

- Architecture-neutral

Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.

- Portable

Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. The compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.

- Robust

Java makes an effort to eliminate error-prone situations by emphasizing mainly on compile time error checking and runtime checking.

- Multithreaded

With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.

- Interpreted

Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.

- High Performance

With the use of Just-In-Time compilers, Java enables high performance.

- Distributed

Java is designed for the distributed environment of the internet.

- Dynamic

Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry an extensive amount of run-time information that can be used to verify and resolve accesses to objects at run-time.

2.2 Introduction to AWT

AWT stands for Abstract Window Toolkit. It is a toolkit of classes with which a programmer can develop Graphics and Graphical User Interface components. AWT gives a front-end technology to Java using which a user can interact with a running process; the old style is scanf. Usage of GUI environment is easier even for a non-computer background person. A business man or an elementary school

student is able to use email system and all the credit goes to GUI, which Java supports very extensively.

All the classes required for developing graphics and GUI are placed in the package java.awt and the classes required for event handling are placed separately in the sub package java.awt.event. These two are very big packages and contains lot of classes doing different functionalities. To study these classes, the whole tutorial can be divided into the following topics.

- Drawing graphics
- Studying layout managers
- Developing GUI components
- Handling events – Event handling
- The java.awt package contains the core AWT graphics classes: GUI Component classes, such as Button, TextField, and Label. GUI Container classes, such as Frame and Panel. Layout managers, such as FlowLayout, BorderLayout and GridLayout.
- Custom graphics classes, such as Graphics, Color and Font.

The java.awt.event package supports event handling:

Event classes, such as ActionEvent, MouseEvent, KeyEvent and WindowEvent,

Event Listener Adapter classes, such as MouseAdapter,KeyAdapter and Window Adapter.

AWT provides a platform-independent and device-independent interface to develop graphic programs that runs on all platforms, including Windows, Mac OS X, and Unixes.

Swing in java is part of Java foundation class which is lightweight and platform independent. It is used for creating window based applications. It includes components like button, scroll bar, text field etc. Putting together all these components makes a graphical user interface. In this article, we will go through the concepts involved in the process of building applications using swing in Java. Following are the concepts discussed in this article:

2.3 Introduction to swing

Swing in Java is a lightweight GUI toolkit which has a wide variety of widgets for building optimized window based applications. It is a part of the JFC(Java Foundation Classes). It is build on top of the AWT API and entirely written in java. It is platform independent unlike AWT and has lightweight components.

It becomes easier to build applications since we already have GUI components like button, checkbox etc. This is helpful because we do not have to start from the scratch.

Container Class:

Any class which has other components in it is called as a container class. For building GUI applications at least one container class is necessary.

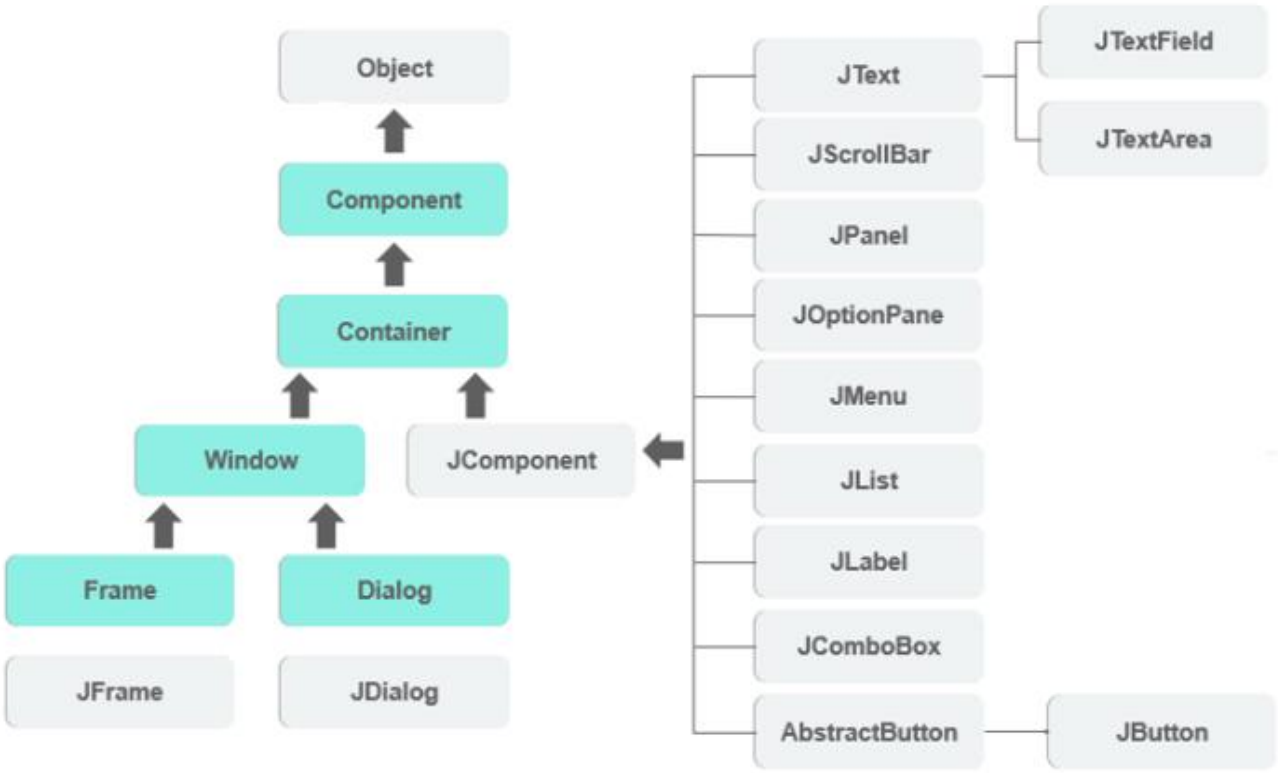
Following are the three types of container classes:

- Panel – It is used to organize components on to a window
- Frame – A fully functioning window with icons and titles
- Dialog – It is like a pop up window but not fully functional like the frame

Difference Between AWT and Swing

AWT	SWING
Platform Dependent	Platform Independent
Does not follow MVC	Follows MVC
Lesser Components	More powerful components
Does not support pluggable look and feel	Supports pluggable look and feel
Heavyweight	Lightweight

Java Swing Class Hierarchy



Explanation: All the components in swing like JButton, JComboBox, JList, JLabel are inherited from the JComponent class which can be added to the container classes. Containers are the windows like frame and dialog boxes. Basic swing components are the building blocks of any gui application. Methods like setLayout override the default layout in each container. Containers like JFrame and JDialog can only add a component to itself, swing provides Layout manager.

2.4 introduction to MySQL

MySQL is an open-source, fast reliable, and flexible relational database management system, typically used with PHP. This chapter is an introductory chapter about MySQL, what is MySQL, and the main features of MySQL are described here.

What is MySQL?

MySQL is a database system used for developing web-based software applications.

MySQL used for both small and large applications.

MySQL is a relational database management system (RDBMS).

MySQL is fast, reliable, and flexible and easy to use.

MySQL supports standard SQL (Structured Query Language).

MySQL is free to download and use.

MySQL was developed by Michael Widenius and David Axmark in 1994.

MySQL is presently developed, distributed, and supported by Oracle Corporation.

MySQL Written in C, C++.

Main Features of MySQL:

MySQL server design is multi-layered with independent modules.

MySQL is fully multithreaded by using kernel threads. It can handle multiple CPUs if they are available.

MySQL provides transactional and non-transactional storage engines.

MySQL has a high-speed thread-based memory allocation system.

MySQL supports in-memory heap table.

MySQL Handles large databases.

MySQL Server works in client/server or embedded systems.

MySQL Works on many different platforms.

Who Uses MySQL

Some of the most famous websites like Facebook, Wikipedia, Google (not for search), YouTube, Flickr.

Content Management Systems (CMS) like WordPress, Drupal, Joomla, phpBB etc.

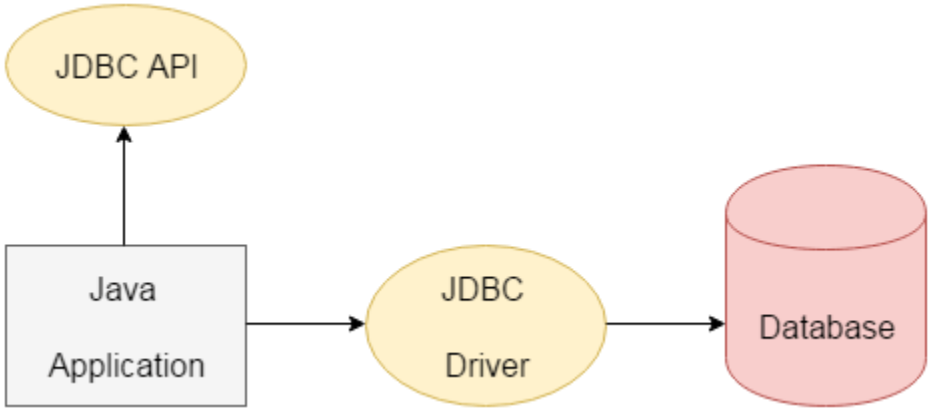
A large number of web developers worldwide are using MySQL to develop web applications.

2.5 introduction JDBC

JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database. It is a part of JavaSE (Java Standard Edition). JDBC API uses JDBC drivers to connect with the database. There are four types of JDBC drivers:

- JDBC-ODBC Bridge Driver,
- Native Driver,
- Network Protocol Driver, and
- Thin Driver

We can use JDBC API to access tabular data stored in any relational database. By the help of JDBC API, we can save, update, delete and fetch data from the database. It is like Open Database Connectivity (ODBC) provided by Microsoft.



Why Should We Use JDBC

Before JDBC, ODBC API was the database API to connect and execute the query with the database. But, ODBC API uses ODBC driver which is written in C language (i.e. platform dependent and unsecured). That is why Java has defined its own API (JDBC API) that uses JDBC drivers (written in Java language).

We can use JDBC API to handle database using Java program and can perform the following activities:

- Connect to the database
- Execute queries and update statements to the database
- Retrieve the result received from the database.

What is API

API (Application programming interface) is a document that contains a description of all the features of a product or software. It represents classes and interfaces that software programs can follow to communicate with each other. An API can be created for applications, libraries, operating systems, etc.

2.6 introduction to JAVA Mail API

The JavaMail is an API that is used to compose, write and read electronic messages (emails).

The JavaMail API provides protocol-independent and plateform-independent framework for sending and receiving mails.

The javax.mail and javax.mail.activation packages contains the core classes of JavaMail API.

The JavaMail facility can be applied to many events. It can be used at the time of registering the user (sending notification such as thanks for your interest to my site), forgot password (sending password to the users email id), sending notifications for important updates etc. So there can be various usage of java mail api.

SMTP

SMTP is an acronym for Simple Mail Transfer Protocol. It provides a mechanism to deliver the email. I have used Apache common mails as an SMTP server. If we use the SMTP server provided by the host provider, authentication is required for sending and receiving emails.

Section 3: Software Requirement Specification (SRC)

This chapter covers the requirements specification of my game “ChopStick”. It includes the specification of this documentation with general description, specific requirements, and analysis of models. It also includes changes management of this requirement specification in case of any change.

2.1 Introduction

In this section the documentation of this report is specified. It specifies the document convention, document scope and also provides a suggestion for the readers of the document.

2.1.1 Purpose of this Chapter

This Software Requirements Specification (SRS) part is intended to give a complete overview of my Project the game “ChopStick” including the action flow, initial user interface and story therein. The SRS document details all features upon which we have currently decided with reference to the manner and importance of their implementation.

2.1.2 Intended Audience and Reading Suggestions

The SRS document also gives project managers a way to ensure the game’s adherence to our original vision. Although the document may be read from front to back for a complete understanding of the project, it was written in sections and hence can be read as such. For an overview of the document and the project itself, see Overall Description. For a detailed description of the game-play elements and their interaction with the player, read System Features. Readers interested in the game-play interface and navigation between different front-end menus should go through External Interface Requirements. Technical standards to which I will hold

the project, it is laid out in Other Nonfunctional Requirements. The development schedule, meanwhile, will be maintained in the Key Milestones.

2.2 General Description

This section includes the perspective of my game, requirements and the system environment.

2.2.1 Hardware requirements

- INTEL I3 Processor or higher.
- 4 GB RAM &above
- Hard disk 6 GB &above

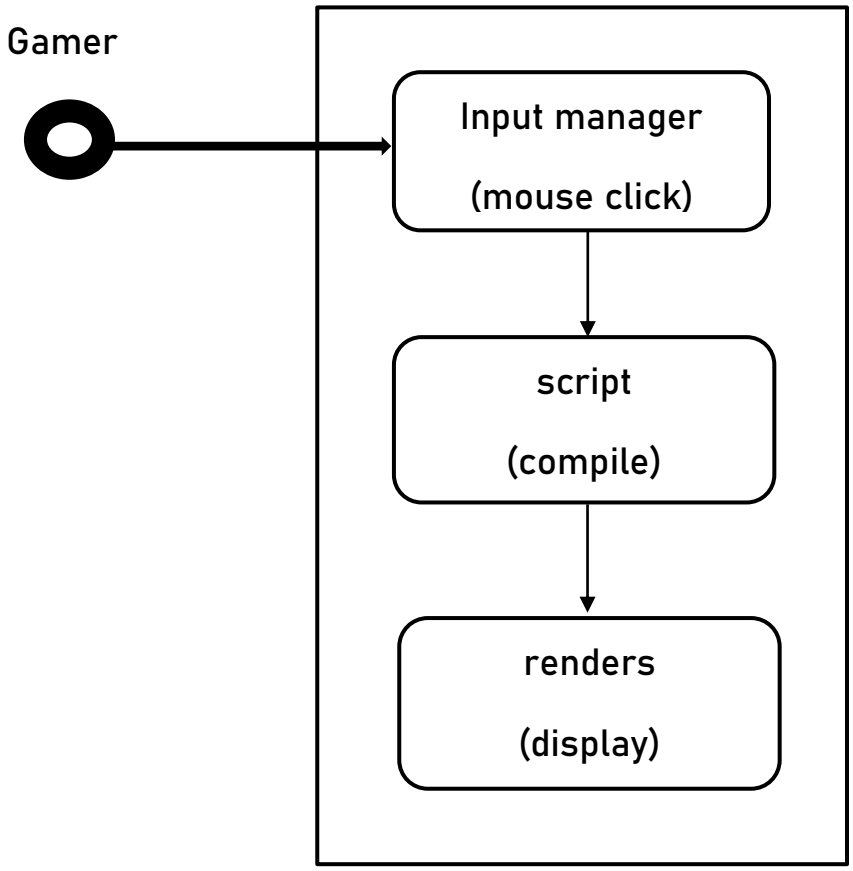
2.2.2 Software requirements

- **Operating system**: windows/mac/ubuntu
- **Language**: JAVA SE 1.7, J2EE
- **Front-end:** AWT, Swing, Window-builder
- **Back-end:** MySQL DB, File DB
- **IDE:** Eclipse
- **for creating .exe file:** Launch4j

2.3 System Environment

Gamer can interact with system by giving input (click to start game) to the system. System gives those inputs to script, if any change occurs (if the value is changed) this object is sent to render to display the things.

Below is the diagram to represent how the game works:



2.3.1. Software Interface

"ChopStick" has been developed using some pretty good development stuff.

Working tools and platform

- Eclipse IDE (J2EE Developers) for whole game development
- Window builder plugin of eclipse for designing
- JDBC Driver for database connection
- Apache common mails API for email
- Flaticon.com for creating sprites
- Images from Bing.com
- MS picture manager for sprites and background-images
- Launch4j for creating exe file

2.3.2 User Characteristics for the System

There could be one user or two at a time in this software and the user interacts with the game (system) in different manner.

So, Gamer is the only one who communicates with the system through playing the game. And this gamer can be any person. The primary requirement is that, the gamer must read the playing procedure provided by me (developers).

2.4 Analysis Model of Our Game Project

This section describes the Software Requirements Specification of my project by analyzing the proper models of requirement engineering.

2.4.1 Scenario Based Model

This Model depicts how the user interacts with the system and the specific sequence of activities that occur as the software is used.

2.5 Requirement Change Management of my System

I as a developer intend to release a complete and fully functional game that follows the guidelines mentioned in the SRS document. updates will likely be required. These updates would consist of any bug fixes that are necessary, compatibility patches for all of the current PCs that support windows/mac/ubuntu and expansions of the content. If the players find any issues or have any comments then they would be able to contact the developers through the official support by providing the review from the “feedback” option of game’s main menu.

The current version of my game is “Plutonium-0.3SNL”.

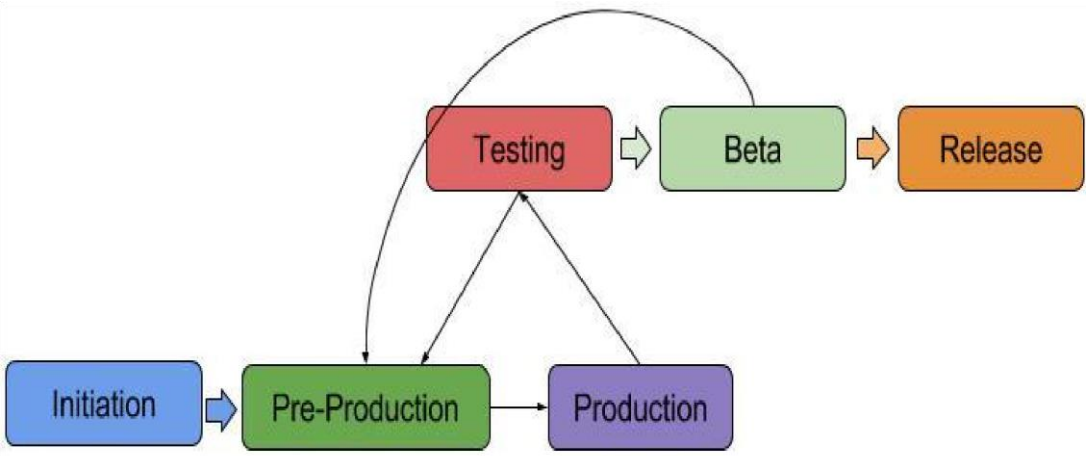
2.5.1 Bugs and Glitches

The players would be able to contact the developers through “feedback” option in game. This is where they would present any bugs or glitches they have detected and if they have any beliefs that the game is not functioning properly. General concerns or comments would also need to be submitted here.

Section 3: Game Development Life Cycle (GDLC)

The Phases

The GDLC typically consists of six phases shown below:



3.1 Initiation phase

The developer decides what kind of game they will make.

3.2 Pre-Production

Before the real production begins.

First pre-production:

- Game Design Document
- First prototype - shows gameplay (has to be fun)

Next cycles -> bug fixing and balancing

3.3 Production phase

The game assets and source code are made.

The result of production is the playable game in form of:

- Formal Details prototype - a playable game, has win-lose rules, co-relations between features, runs well.
- Refinement prototype - most mature prototype which only needs more polishing. Should be feature complete and almost ready to ship.

3.4 Testing phase

Evaluation of game features, value, concept and design.

Task: which questions should the testing answer?

Evaluation of game features, value, concept and design.

Test report -> all the bugs, thing that should be omitted, included or changed.

The test result should decide whether to reiterate or advance to beta testing.

Task: which questions should the testing answer?

Evaluation of game features, value, concept and design.

Some questions the testing should be answer:

1. Is the game still buggy?
2. Is it possible to get stuck in the game?
3. Is there any sign of exploits/glitch?
4. Is the game too easy/hard to beat?
5. Is there any feature missing?
6. Does the game run well on every platform?

3.5 Beta Phase

The beta test is testing cycle conducted by third party:

- publisher,
- potential buyer,
- game reviewer

Result should be a test report.

Decide whether the game is ready for shipping.

3.6 Release Phase

Work may seem to be done but it is not!

- Bugfixing, patching
- Aditions, special events
- Marketing
- Community management

Sub-phases

There are multiple sub-phases that help the game development in the long run.

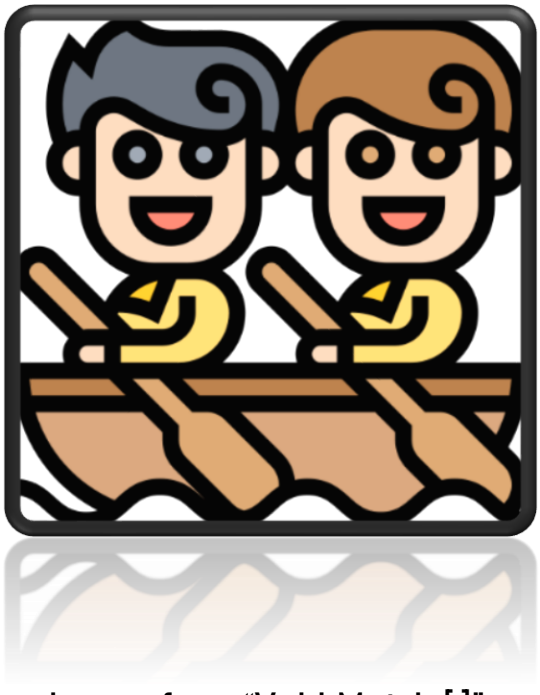
Section 4: Graphics, Sound and Implementation of “ChopStick” game

This chapter covers the project design phases, the system features and also the implementation of the features.

4.1 Implementation Tools Required

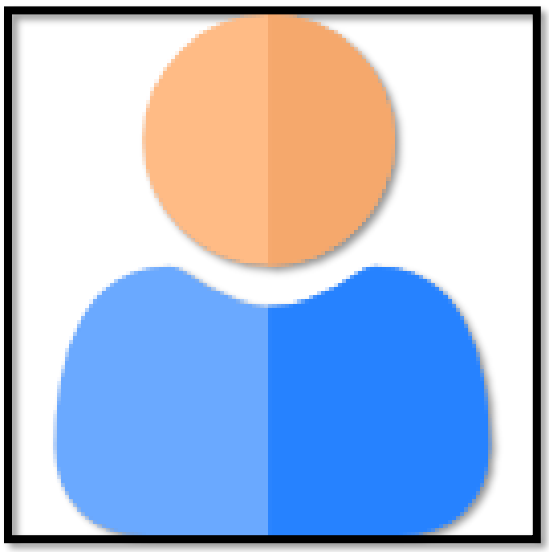
~~	Product of	Tool	Platform type	usage
	Eclipse foundation	Eclipse	IDE	To develop
		Window builder	Plug in	For front-end designing
	Flaticon	Flaticon vector icons and stickers	website	For creating sprites
	Microsoft	Picture manager	image editor	For editing images
		bing	Web browser	For getting background images
	apache	Common mails	API	For working with emails

4.2 Graphics for my game(sprites and images)

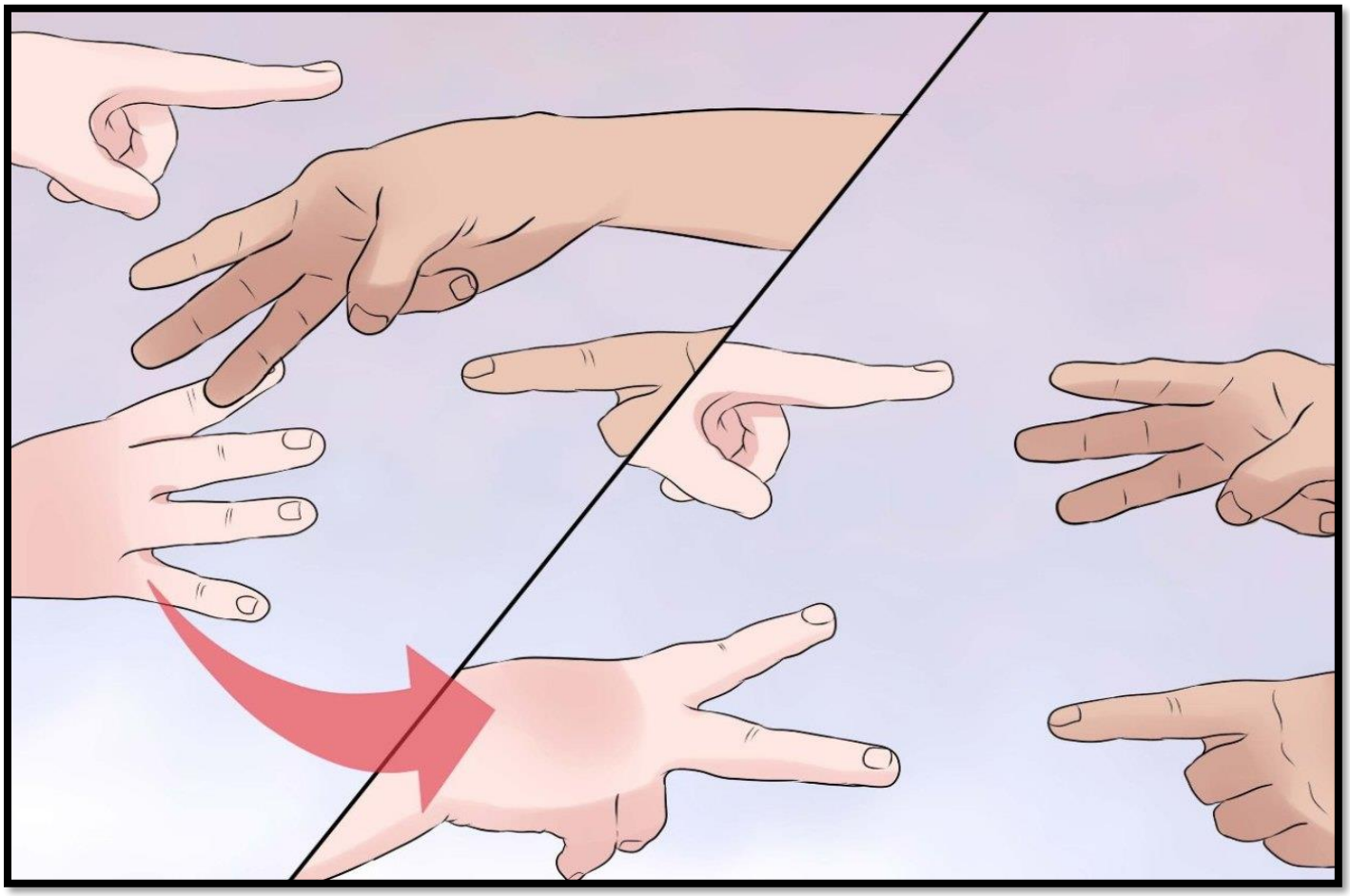


Logo of my “Void Matrix[:]” Studio

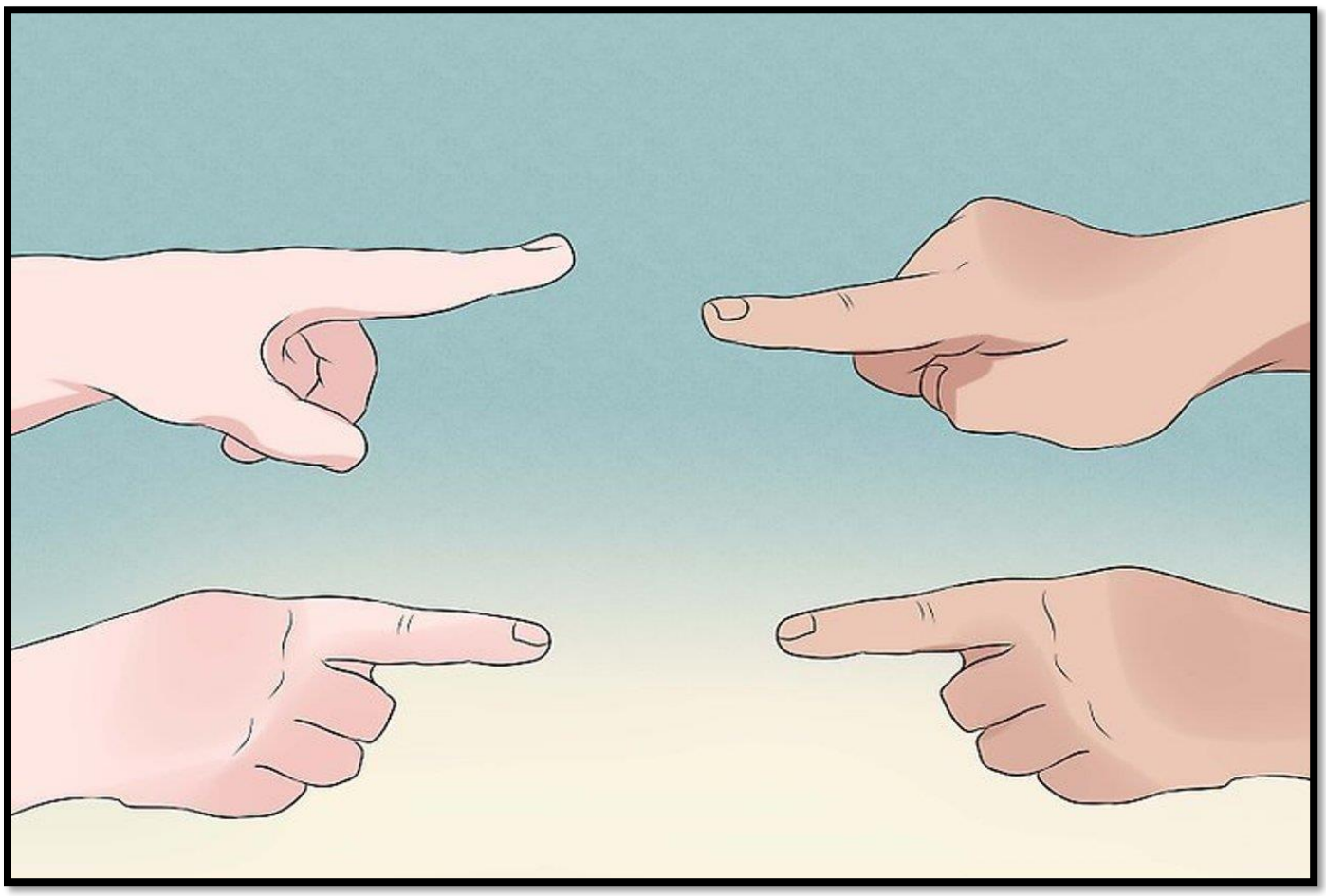
Logo of my “ChopStick” game



Default and unchangeable picture of player/user



Background image of my loading screen



Background image of my main menu/frame

Player & player 1's left hand






Player & player 1's right hand





computer & player 2's right hand




Music: We Got What You Want by Nine one nine

Computer and player 2's left hands

Section 5: Data dictionary & Diagrams

5.1 Data Dictionary

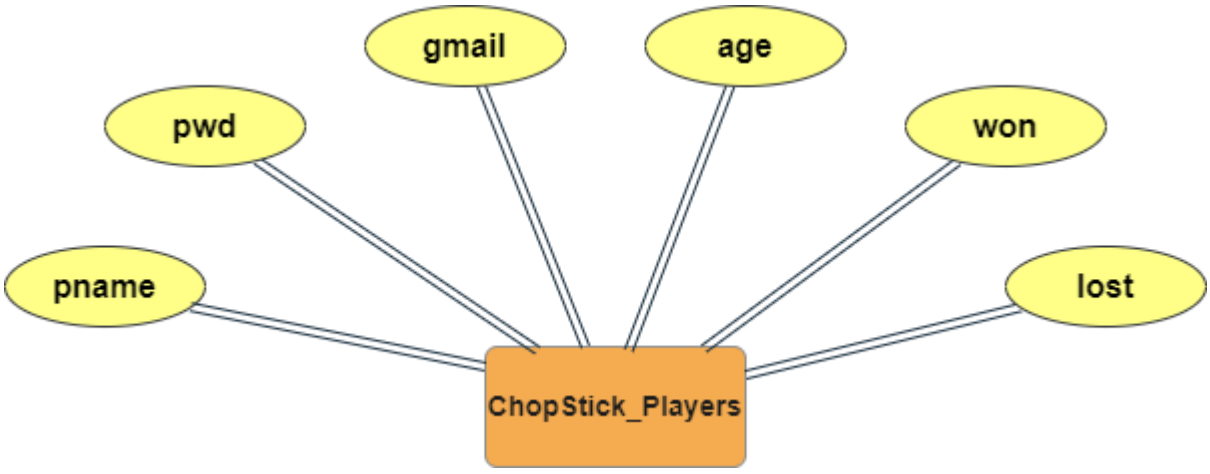
stickchop

ChopStick_Players

Column	Type	Null	Default	Links to	Comments	Media type
pname	varchar(50)	No				
pwd	varchar(40)	No				
gmail	varchar(60)	No				
age	int	No				
won	int	Yes	*NULL*			
lost	int	Yes	*NULL*			

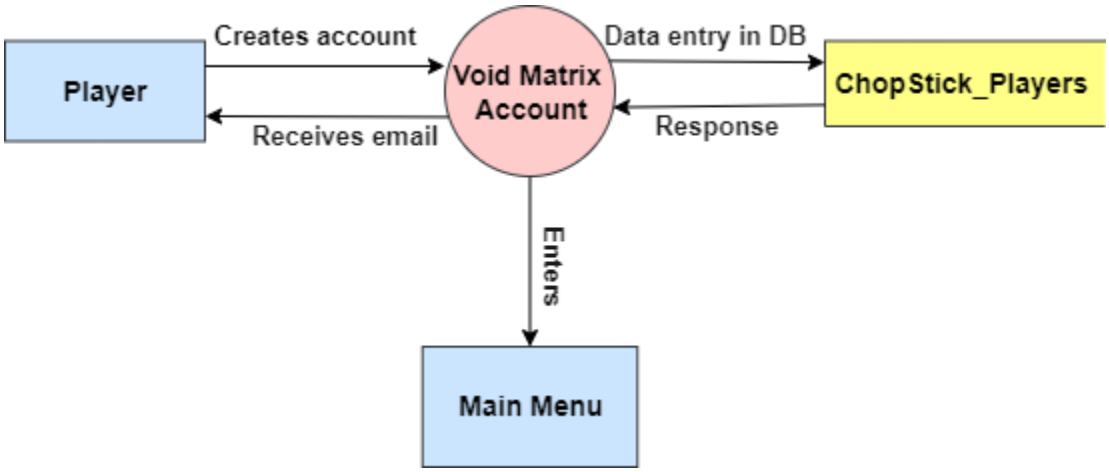
5.2 diagrams

5.2.1 Entity Relationship Diagram (ERD)

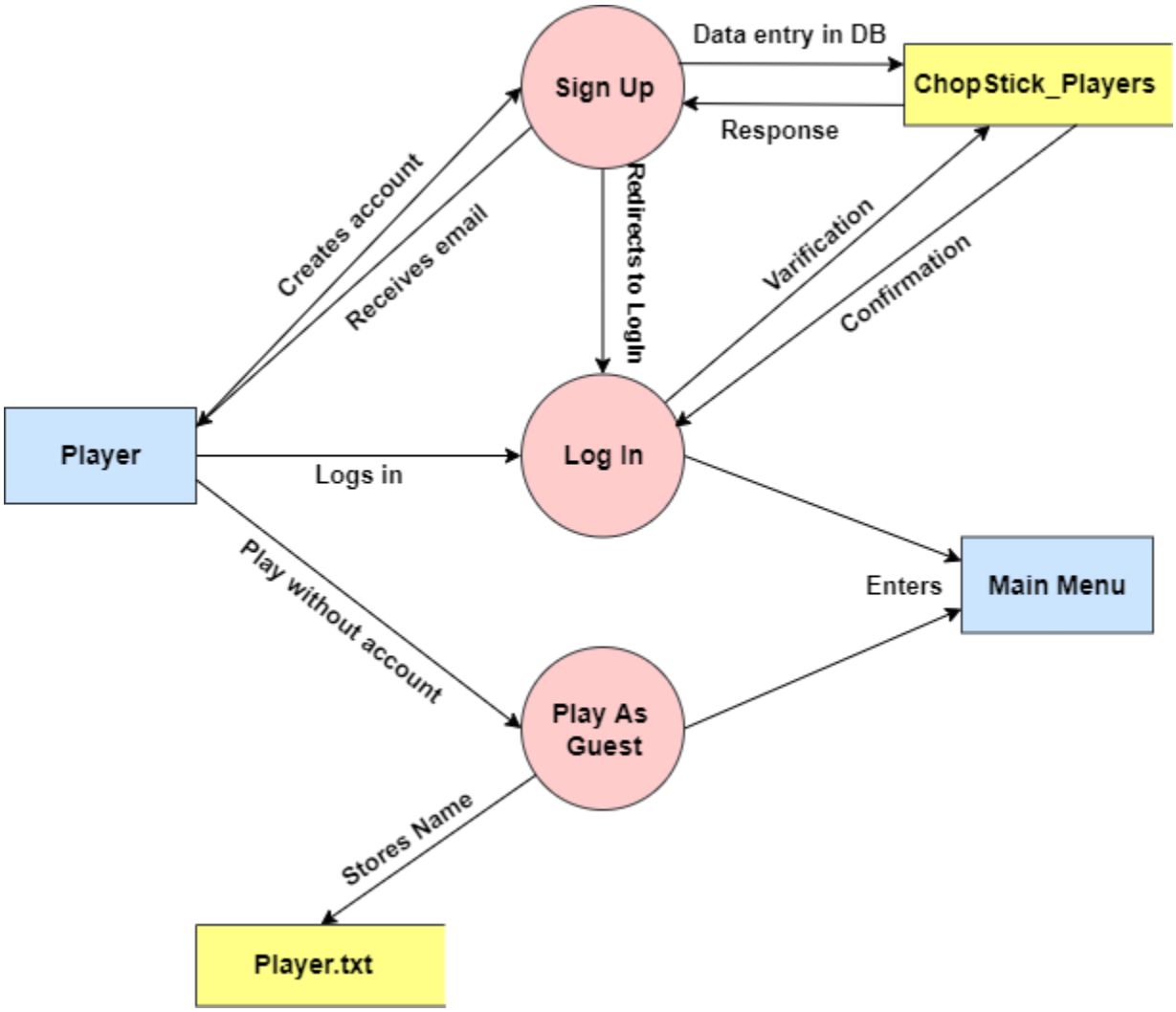


5.2.2 Data Flow Diagram (DFD)

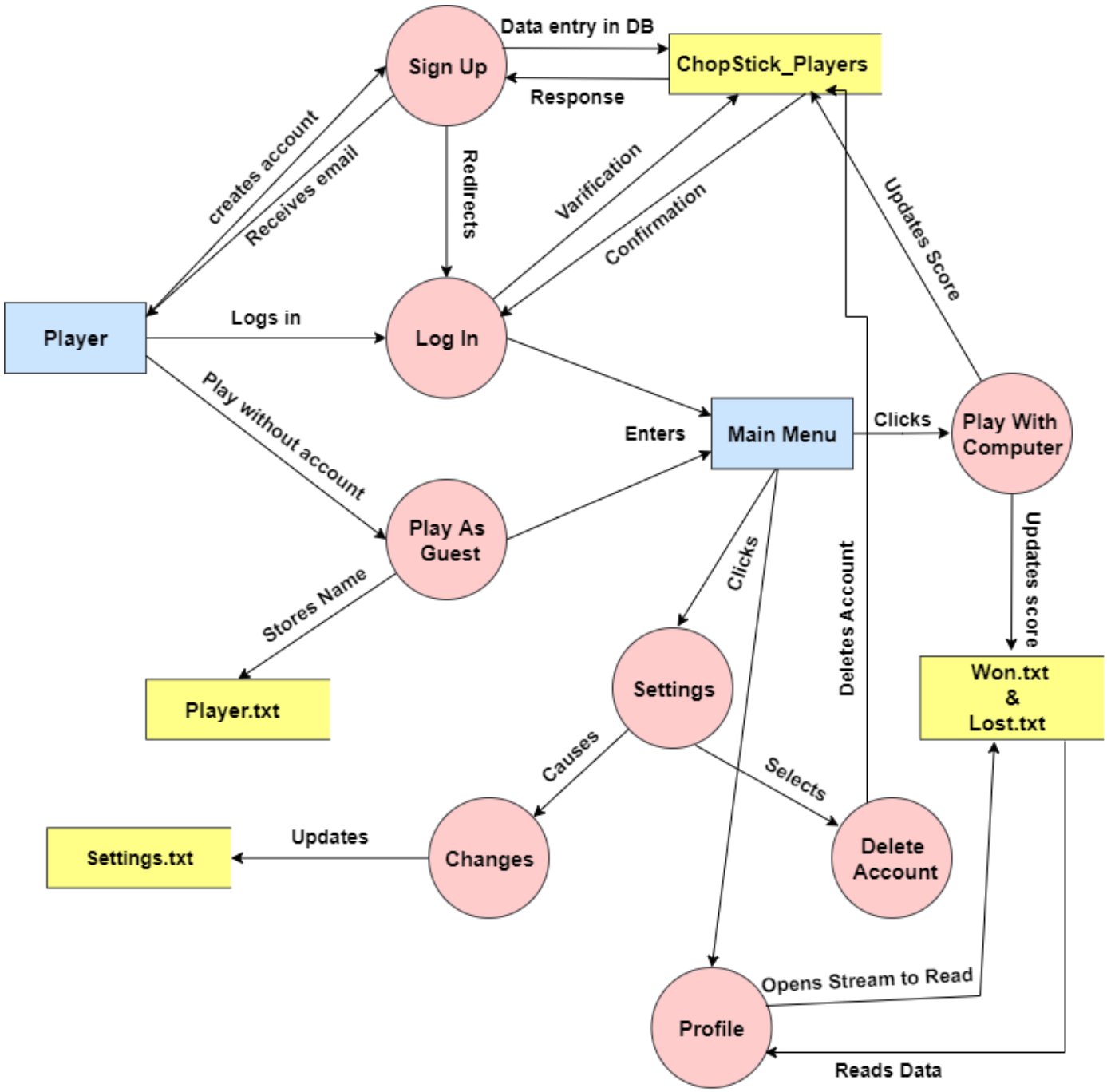
Level 0



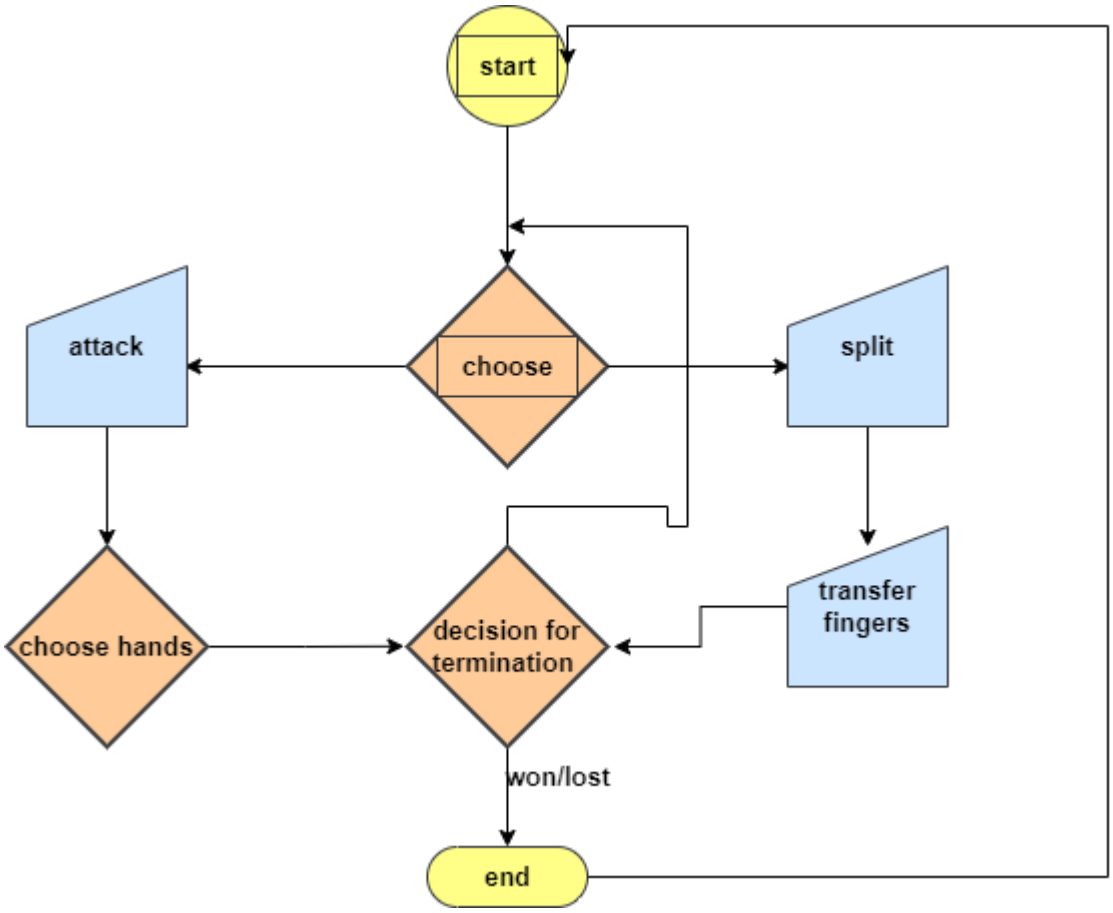
Level 1



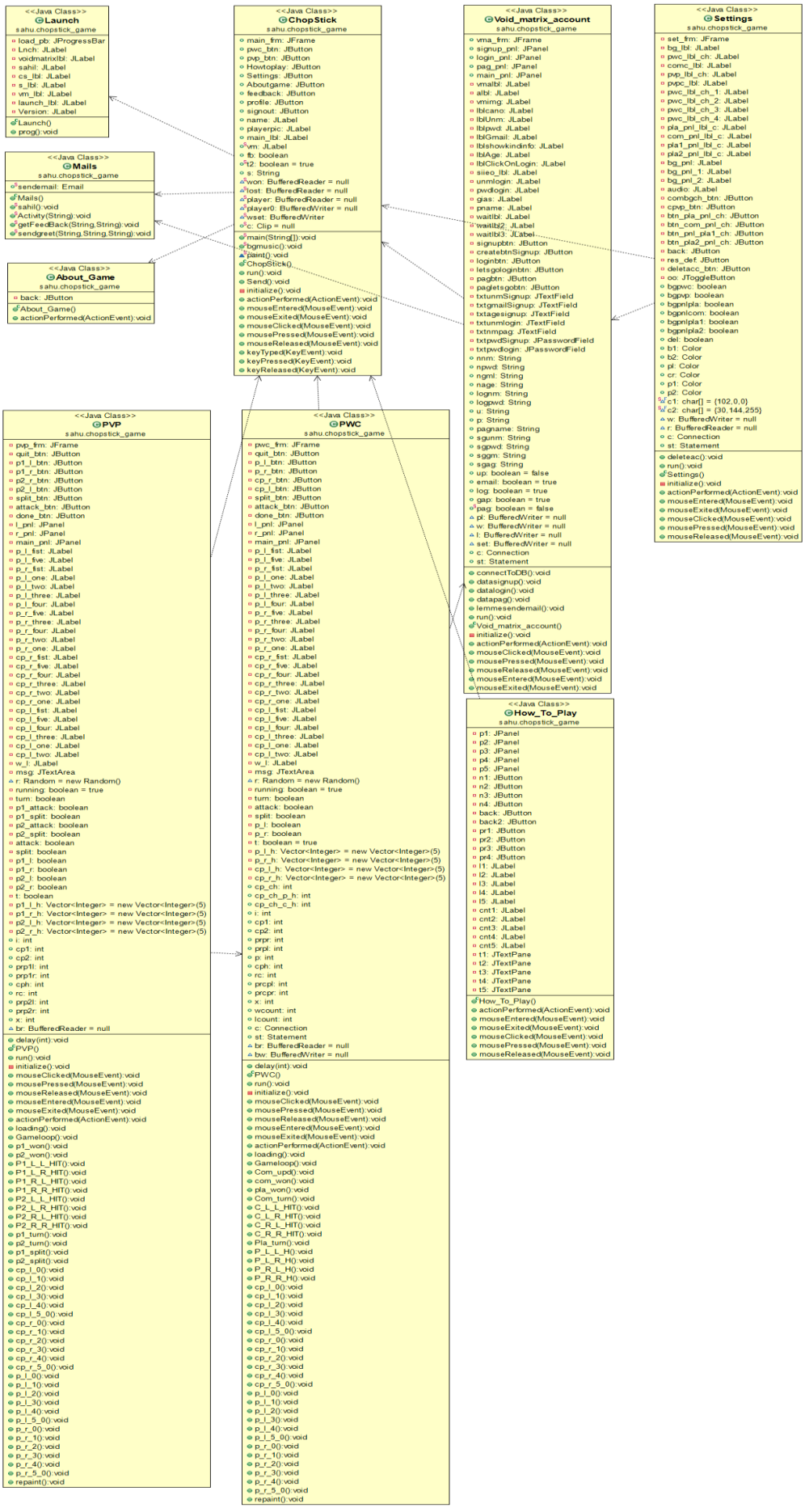
Level 2


Sign Up
Data entry in DB
ChopStick_Players
Response
creates account
Receives email
Redirects
Varification
Confirmation
Updates Score
Player
Logs in
Log In
Play without account
Enters
Main Menu
Clicks
Play With Computer
Play As Guest
Clicks
Deletes Account
Updates score
Stores Name
Player.txt
Settings
Won.txt & Lost.txt
Causes
Selects
Updates
Settings.txt
Changes
Delete Account
Opens Stream to Read
Profile
Reads Data

5.2.3 Flowchart



5.2.5 class diagram



Section 6: Code of ChopStick game

Launch.java

```
package sahu.chopstick_game;

import javax.swing.JFrame;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.JProgressBar;
import java.awt.Color;
import javax.swing.ImageIcon;
import javax.swing.SwingConstants;
import java.awt.Toolkit;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import javax.swing.border.BevelBorder;
import javax.swing.border.SoftBevelBorder;
import javax.swing.border.EtchedBorder;
import javax.swing.border.LineBorder;
import javax.swing.border.MatteBorder;

@SuppressWarnings({ "serial", "unused" })

public class Launch extends JFrame {

	private JProgressBar load_pb;
	private JLabel Lnch,
	               voidmatrixlbl,
	               sahil,
	               cs_lbl,
	               s_lbl,
	               vm_lbl,
	               launch_lbl;
	private JLabel Version;

	public Launch() {
```

```
    setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().g
etResource("logo_F.png")));
          setTitle("ChopStick");
          getContentPane().setBackground(new Color(204, 204, 255));
          getContentPane().setLayout(null);
          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
          setBounds(150, 20, 1250, 785);

          voidmatrixlbl = new JLabel();
          voidmatrixlbl.setBorder(new
BevelBorder(BevelBorder.RAISED, null, null, null, null));
          voidmatrixlbl.setBounds(1010, 187, 105, 82);
          voidmatrixlbl.setIcon(new
ImageIcon(getClass().getResource("team1.png")));
          getContentPane().add(voidmatrixlbl);

          sahil = new JLabel("sAhiL lALaNi");
          sahil.setBorder(new BevelBorder(BevelBorder.RAISED, null,
null, null, null));
          sahil.setHorizontalAlignment(SwingConstants.CENTER);
          sahil.setForeground(new Color(102, 0, 0));
          sahil.setFont(new Font("Tahoma", Font.BOLD, 40));
          sahil.setBounds(457, 0, 340, 90);
          getContentPane().add(sahil);

          load_pb = new JProgressBar();
          load_pb.setBorder(new EtchedBorder(EtchedBorder.RAISED,
null, null));
          load_pb.setFont(new Font("Tahoma", Font.BOLD, 30));
          load_pb.setForeground(new Color(153, 0, 102));
          load_pb.setStringPainted(true);
          load_pb.setBounds(294, 528, 689, 62);
          getContentPane().add(load_pb);

          cs_lbl = new JLabel("ChopStick");
          cs_lbl.setBorder(null);
          cs_lbl.setForeground(new Color(102, 0, 51));
          cs_lbl.setHorizontalAlignment(SwingConstants.CENTER);
          cs_lbl.setFont(new Font("Tahoma", Font.BOLD, 90));
          cs_lbl.setBounds(395, 249, 468, 175);
          getContentPane().add(cs_lbl);

          s_lbl = new JLabel("Studio");
```

```
        s_lbl.setBorder(new BevelBorder(BevelBorder.RAISED, null,
null, null, null));
        getContentPane().add(s_lbl);
        s_lbl.setForeground(new Color(102, 42, 0));
        s_lbl.setFont(new Font("Tahoma", Font.BOLD, 30));
        s_lbl.setHorizontalAlignment(SwingConstants.CENTER);
        s_lbl.setBounds(989, 136, 144, 51);

        Lnch = new JLabel("...Launching...");
        Lnch.setBorder(new SoftBevelBorder(BevelBorder.LOWERED,
null, null, null, null));
        Lnch.setForeground(new Color(153, 0, 0));
        Lnch.setHorizontalAlignment(SwingConstants.CENTER);
        Lnch.setFont(new Font("Tahoma", Font.BOLD, 15));
        Lnch.setBounds(562, 592, 139, 32);
        getContentPane().add(Lnch);

        vm_lbl = new JLabel("Void_Matrix[:]\r\n");
        vm_lbl.setBorder(new BevelBorder(BevelBorder.RAISED,
null, null, null, null));
        getContentPane().add(vm_lbl);
        vm_lbl.setForeground(new Color(102, 0, 0));
        vm_lbl.setHorizontalAlignment(SwingConstants.CENTER);
        vm_lbl.setFont(new Font("Tahoma", Font.BOLD, 40));
        vm_lbl.setBounds(894, 53, 332, 82);

        Version = new JLabel("Plutonium-0.3SNL");
        Version.setHorizontalAlignment(SwingConstants.CENTER);
        Version.setForeground(new Color(102, 0, 0));
        Version.setFont(new Font("Tahoma", Font.BOLD, 40));
        Version.setBorder(new BevelBorder(BevelBorder.RAISED,
null, null, null, null));
        Version.setBounds(10, 699, 380, 39);
        getContentPane().add(Version);

        launch_lbl = new JLabel();
        launch_lbl.setBorder(new
SoftBevelBorder(BevelBorder.RAISED, null, null, null, null));
        launch_lbl.setHorizontalTextPosition(JLabel.CENTER);
        launch_lbl.setIcon(new
ImageIcon(getClass().getResource("csm.jpg")));
        launch_lbl.setVerticalAlignment(SwingConstants.TOP);
        launch_lbl.setForeground(new Color(0, 0, 255));
        launch_lbl.setFont(new Font("Tahoma", Font.BOLD, 99));
        launch_lbl.setBounds(0, 0, 1238, 750);
```

```
            getContentPane().add(launch_lbl);

      }

      public void prog() throws Exception
      {
            int n=0;
            while(n<=100)
            {
                  load_pb.setValue(n);
                  n+=1;
                  Thread.sleep(20);
            }
      }
}
```

Void_matrix_account.java

```
package sahu.chopstick_game;

import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.Toolkit;

import javax.swing.JFrame;
import javax.swing.JDesktopPane;
import javax.swing.JMenuBar;
import javax.swing.JOptionPane;
import javax.swing.JMenu;
import javax.swing.JSeparator;
import javax.swing.JTabbedPane;
import javax.swing.border.MatteBorder;
import javax.swing.JPanel;
import java.awt.ComponentOrientation;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.SwingConstants;
import javax.swing.JTextField;
import javax.swing.border.EtchedBorder;
import javax.swing.border.LineBorder;
```

```
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.awt.event.ActionEvent;
import javax.swing.JPasswordField;
import javax.swing.border.BevelBorder;
import java.awt.Insets;
import javax.swing.DropMode;
import javax.swing.Icon;
import javax.swing.ImageIcon;
import javax.swing.border.SoftBevelBorder;

import org.apache.commons.mail.EmailException;

import javax.swing.border.CompoundBorder;
import java.awt.SystemColor;
import javax.swing.UIManager;

@SuppressWarnings({"unused","deprecation"})
public class Void_matrix_account implements
MouseListener,Runnable,ActionListener{

     public JFrame vma_frm;

     public JPanel signup_pnl,login_pnl,pag_pnl,main_pnl;

     private JLabel vmalbl,albl,vmimg,

lblcano,lblUnm,lblpwd,lblGmail,lblshowkindinfo,lblAge,lblClickOnLog
in,
                    siieo_lbl,unmlogin,pwdlogin,
                    gias,pname,waitlbl,waitlbl2,waitlbl3;

     private JButton signupbtn,createbtnSignup,
                      loginbtn,letsgologinbtn,
                      pagbtn,pagletsgobtn;
```

```
    private JTextField txtunmSignup, txtgmailSignup, txtagesignup,
                        txtunmlogin,
                        txtnmpag;

    private JPasswordField txtpwdSignup,txtpwdlogin;

    public String nnm,npwd,ngml,nage,
                    lognm,logpwd, u,p,
                    pagname,
                    sgunm,sgpwd,sggm,sgag;

    public boolean up=false,email=true,log=true,gap=true;
    public static boolean pag=false;

    BufferedWriter pl=null,w=null,l=null,set=null;

    public Connection c;
    public Statement st;

    public void connectToDB() {
        String
url="jdbc:mysql://db4free.net:3306/stickchop?useSSL=false";
        String usr="voidmatrix";
        String s="voidmatrix2002";
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            c= DriverManager.getConnection(url,usr,s);
            st=c.createStatement();
        } catch (Exception e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(signup_pnl,
                    "So Sorry, some errors occurred\nBut you
can still play\nErr: "+e.getMessage(),
                    "DBConnection failure",
JOptionPane.ERROR_MESSAGE);
        }
    }

    public void datasignup() {
        try {
            PreparedStatement p=c.prepareStatement("insert into
ChopStick_Players values(?,?,?,?,?,?)");
            p.setString(1, sgunm);
            p.setString(2, sgpwd);
            p.setString(3, sggm);
```

```
                p.setInt(4, Integer.parseInt(sgag));
                p.setInt(5, 0);
                p.setInt(6, 0);
                p.execute();
                w=new BufferedWriter(new FileWriter("won.txt"));
                l=new BufferedWriter(new FileWriter("lost.txt"));
                w.write(0);
                l.write(0);
                l.close();
                w.close();
           }catch(Exception e) {
                e.printStackTrace();
                JOptionPane.showMessageDialog(signup_pnl,
                     "So Sorry, some errors occurred\nBut you can
still play\nErr: "+e.getMessage(),
                     "Failed to sign up",
JOptionPane.WARNING_MESSAGE);
                try {
                     w=new BufferedWriter(new FileWriter("won.txt"));
                     l=new BufferedWriter(new
FileWriter("lost.txt"));
                     w.write(0);
                     l.write(0);
                     l.close();
                     w.close();
                }catch(Exception e1) {
                     e1.printStackTrace();
                }
           }
     }

     public void datalogin() {
          try {
                ResultSet rs=st.executeQuery("select * from
signuChopStick_Players");
                while(rs.next()){
                     if(lognm.equals(rs.getString(1)) &&
logpwd.equals(rs.getString(2))) {
                          up=true;
                          pl=new BufferedWriter(new
FileWriter("player.txt"));
                          w=new BufferedWriter(new
FileWriter("won.txt"));
                          l=new BufferedWriter(new
FileWriter("lost.txt"));
```

```
                    pl.write(lognm);
                    w.write(rs.getInt(5));
                    l.write(rs.getInt(6));
                    pl.close();
                    w.close();
                    l.close();
                    break;
                }
            }
        }catch(Exception e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(login_pnl,
                    "So Sorry, some errors occurred\nBut you
can still play\nErr: "+e.getMessage(),
                     "Failed to login",
JOptionPane.ERROR_MESSAGE);
            try {
                pl=new BufferedWriter(new
FileWriter("player.txt"));
                w=new BufferedWriter(new FileWriter("won.txt"));
                l=new BufferedWriter(new
FileWriter("lost.txt"));
                pl.write(lognm);
                w.write(0);
                l.write(0);
                pl.close();
                w.close();
                l.close();
                vma_frm.dispose();
                c.close();
                ChopStick cs=new ChopStick();
                cs.name.setText(lognm);
                cs.main_frm.setVisible(true);
            }catch(Exception e1) {
                e1.printStackTrace();
            }

        }
        if(!up) {
            JOptionPane.showMessageDialog(login_pnl,
                                                "Invalid Username
or Password","Error Logging in",

JOptionPane.ERROR_MESSAGE);
        }
```

```
            else {
                vma_frm.dispose();
                try {
                    c.close();
                    ChopStick cs=new ChopStick();
                    cs.name.setText(lognm);
                    cs.main_frm.setVisible(true);
                }
                catch(Exception e) {
                    e.printStackTrace();
                }
            }
        }

        public void datapag() {
            try {
                pl=new BufferedWriter(new FileWriter("player.txt"));
                w=new BufferedWriter(new FileWriter("won.txt"));
                l=new BufferedWriter(new FileWriter("lost.txt"));
                w.write(0);
                l.write(0);
                pl.write(pagname);
                pl.close();
                w.close();
                l.close();
                PreparedStatement p=c.prepareStatement("insert into
pag values(?)");
                p.setString(1, pagname);
                p.execute();

            }
            catch(Exception e) {
                JOptionPane.showMessageDialog(pag_pnl,
e.getMessage()+"\nbut You can still play",
                                                "Connection Failure",
JOptionPane.ERROR_MESSAGE);
                try {
                    vma_frm.dispose();
                    ChopStick cs=new ChopStick();
                    cs.name.setText(pagname);
                    cs.main_frm.setVisible(true);
                    ChopStick.t2=false;
                    set=new BufferedWriter(new
FileWriter("settings.txt"));
                    set.close();
```

```
                }catch(Exception e1) {
                    e1.printStackTrace();
                }
            }
        }


        public void lemmesendemail() throws Exception {
            email=true;
            while(email) {
                Thread.sleep(1000);
            }
            Mails.sendgreet(sggm, sgunm, sgpwd);
            Mails.Activity(sgunm);
            lemmesendemail();
        }

        @Override
        public void run() {
            try {
                connectToDB();
                lemmesendemail();
            } catch (Exception e) {
                e.printStackTrace();
            }

        }

        public Void_matrix_account() {
            initialize();
            new Thread(this).start();
        }

        private void initialize() {
            vma_frm= new JFrame();
            vma_frm.setResizable(false);
            vma_frm.setForeground(Color.BLACK);

        vma_frm.setIconImage(Toolkit.getDefaultToolkit().getImage(getC
lass().getResource("logo_F.png")));
            vma_frm.setBackground(Color.BLACK);
            vma_frm.setFont(new Font("Dialog", Font.BOLD, 20));
            vma_frm.setTitle("ChopStick");
            vma_frm.setBounds(150, 20, 1250, 785);
            vma_frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        vma_frm.getContentPane().setLayout(null);

        main_pnl = new JPanel();
        main_pnl.setBackground(new Color(0, 255, 0));
        main_pnl.setBorder(new EtchedBorder(EtchedBorder.LOWERED,
null, null));
        main_pnl.setBounds(0, 0, 319, 748);
        vma_frm.getContentPane().add(main_pnl);
        main_pnl.setLayout(null);

        signupbtn = new JButton("Sign Up");
        signupbtn.setBorder(new BevelBorder(BevelBorder.RAISED,
null, null, null, null));
        signupbtn.addMouseListener(this);
        signupbtn.setForeground(new Color(0, 0, 51));
        signupbtn.addActionListener(this);
        signupbtn.setFocusable(false);
        signupbtn.setBackground(new Color(255, 153, 153));
        signupbtn.setFont(new Font("Tahoma", Font.BOLD, 25));
        signupbtn.setBounds(83, 278, 146, 55);
        main_pnl.add(signupbtn);

        loginbtn = new JButton("Log In");
        loginbtn.setBorder(new BevelBorder(BevelBorder.RAISED,
null, null, null, null));
        loginbtn.addMouseListener(this);
        loginbtn.setForeground(new Color(0, 0, 51));
        loginbtn.addActionListener(this);
        loginbtn.setFocusable(false);
        loginbtn.setFont(new Font("Tahoma", Font.BOLD, 25));
        loginbtn.setBackground(new Color(255, 153, 153));
        loginbtn.setBounds(83, 389, 146, 55);
        main_pnl.add(loginbtn);

        pagbtn = new JButton("Play As Guest");
        pagbtn.setBorder(new BevelBorder(BevelBorder.RAISED,
null, null, null, null));
        pagbtn.addMouseListener(this);
        pagbtn.setForeground(new Color(0, 0, 51));
        pagbtn.addActionListener(this);
        pagbtn.setFocusable(false);
        pagbtn.setFont(new Font("Tahoma", Font.BOLD, 25));
        pagbtn.setBackground(new Color(255, 153, 153));
        pagbtn.setBounds(48, 500, 212, 55);
        main_pnl.add(pagbtn);
```

```
        vmalbl = new JLabel("Void_Matrix[:]");
        vmalbl.setBorder(new SoftBevelBorder(BevelBorder.RAISED,
null, null, null, null));
        vmalbl.setForeground(new Color(0, 0, 204));
        vmalbl.setFont(new Font("Tahoma", Font.BOLD, 35));
        vmalbl.setHorizontalAlignment(SwingConstants.CENTER);
        vmalbl.setBounds(20, 140, 274, 56);
        main_pnl.add(vmalbl);

        albl = new JLabel("Account");
        albl.setBorder(new SoftBevelBorder(BevelBorder.RAISED,
null, null, null, null));
        albl.setHorizontalAlignment(SwingConstants.CENTER);
        albl.setForeground(new Color(0, 0, 204));
        albl.setFont(new Font("Tahoma", Font.BOLD, 35));
        albl.setBounds(60, 199, 197, 43);
        main_pnl.add(albl);

        vmimg = new JLabel();
        vmimg.setBorder(new BevelBorder(BevelBorder.RAISED, null,
null, null, null));
        vmimg.setBounds(104, 42, 101, 96);
        vmimg.setIcon(new
ImageIcon(getClass().getResource("team1.png")));
        main_pnl.add(vmimg);

        signup_pnl = new JPanel();
        signup_pnl.setBorder(new
EtchedBorder(EtchedBorder.RAISED, null, null));
        signup_pnl.setBackground(new Color(255, 255, 51));
        signup_pnl.setBounds(318, 0, 918, 748);
        vma_frm.getContentPane().add(signup_pnl);
        signup_pnl.setLayout(null);

        lblcano = new JLabel("Create A New One");
        lblcano.setBorder(new BevelBorder(BevelBorder.RAISED,
null, null, null, null));
        lblcano.setHorizontalAlignment(SwingConstants.CENTER);
        lblcano.setForeground(new Color(0, 0, 51));
        lblcano.setFont(new Font("Tahoma", Font.BOLD, 30));
        lblcano.setBounds(300, 30, 313, 71);
        signup_pnl.add(lblcano);

        lblUnm = new JLabel("Username\t  \t  :");
```

```
        lblUnm.setBorder(new BevelBorder(BevelBorder.RAISED,
null, null, null, null));
        lblUnm.setBackground(Color.LIGHT_GRAY);
        lblUnm.setOpaque(true);
        lblUnm.setHorizontalAlignment(SwingConstants.CENTER);
        lblUnm.setForeground(new Color(0, 0, 51));
        lblUnm.setFont(new Font("Tahoma", Font.BOLD, 25));
        lblUnm.setBounds(10, 131, 228, 54);
        signup_pnl.add(lblUnm);

        lblpwd = new JLabel("Password\t    :");
        lblpwd.setBorder(new BevelBorder(BevelBorder.RAISED,
null, null, null, null));
        lblpwd.setBackground(Color.LIGHT_GRAY);
        lblpwd.setOpaque(true);
        lblpwd.setHorizontalAlignment(SwingConstants.CENTER);
        lblpwd.setForeground(new Color(0, 0, 51));
        lblpwd.setFont(new Font("Tahoma", Font.BOLD, 25));
        lblpwd.setBounds(10, 244, 228, 54);
        signup_pnl.add(lblpwd);

        lblGmail = new JLabel("Gmail Address\t  :");
        lblGmail.setOpaque(true);
        lblGmail.setBorder(new BevelBorder(BevelBorder.RAISED,
null, null, null, null));
        lblGmail.setBackground(Color.LIGHT_GRAY);
        lblGmail.setHorizontalAlignment(SwingConstants.CENTER);
        lblGmail.setForeground(new Color(0, 0, 51));
        lblGmail.setFont(new Font("Tahoma", Font.BOLD, 25));
        lblGmail.setBounds(10, 365, 228, 54);
        signup_pnl.add(lblGmail);

        lblAge = new JLabel("age\t      :");
        lblAge.setBorder(new SoftBevelBorder(BevelBorder.RAISED,
null, null, null, null));
        lblAge.setOpaque(true);
        lblAge.setBackground(Color.LIGHT_GRAY);
        lblAge.setHorizontalAlignment(SwingConstants.CENTER);
        lblAge.setForeground(new Color(0, 0, 51));
        lblAge.setFont(new Font("Tahoma", Font.BOLD, 25));
        lblAge.setBounds(10, 480, 228, 54);
        signup_pnl.add(lblAge);

        txtpwdSignup = new JPasswordField();
        txtpwdSignup.setColumns(30);
```

```
        txtpwdSignup.setBorder(new
EtchedBorder(EtchedBorder.RAISED, null, null));
        txtpwdSignup.setForeground(new Color(51, 255, 0));
        txtpwdSignup.setFont(new Font("Tahoma", Font.BOLD, 25));
        txtpwdSignup.setBounds(237, 245, 528, 54);
        signup_pnl.add(txtpwdSignup);

        txtunmSignup = new JTextField();
        txtunmSignup.setBorder(new
EtchedBorder(EtchedBorder.LOWERED, null, null));
        txtunmSignup.setFont(new Font("Tahoma", Font.BOLD, 25));
        txtunmSignup.setForeground(new Color(0, 255, 0));
        txtunmSignup.setBounds(237, 131, 611, 54);
        signup_pnl.add(txtunmSignup);
        txtunmSignup.setColumns(40);

        txtgmailSignup = new JTextField();
        txtgmailSignup.setBorder(new
EtchedBorder(EtchedBorder.LOWERED, null, null));
        txtgmailSignup.setFont(new Font("Tahoma", Font.BOLD,
25));
        txtgmailSignup.setForeground(new Color(0, 255, 0));
        txtgmailSignup.setColumns(50);
        txtgmailSignup.setBounds(237, 365, 671, 54);
        signup_pnl.add(txtgmailSignup);

        txtagesignup = new JTextField();
        txtagesignup.setBorder(new
EtchedBorder(EtchedBorder.LOWERED, null, null));
        txtagesignup.setForeground(new Color(0, 255, 0));
        txtagesignup.setFont(new Font("Tahoma", Font.BOLD, 25));
        txtagesignup.setColumns(10);
        txtagesignup.setBounds(237, 481, 213, 54);
        signup_pnl.add(txtagesignup);

        lblshowkindinfo = new JLabel("enter a valid gmail address
for recieving mails");
        lblshowkindinfo.setBorder(new
EtchedBorder(EtchedBorder.RAISED, null, null));
        lblshowkindinfo.setOpaque(true);
        lblshowkindinfo.setBackground(Color.BLUE);
        lblshowkindinfo.setFont(new Font("Tahoma", Font.BOLD,
15));
        lblshowkindinfo.setForeground(Color.GREEN);
```

```
    lblshowkindinfo.setHorizontalAlignment(SwingConstants.CENTER);
        lblshowkindinfo.setBounds(237, 418, 671, 30);
        signup_pnl.add(lblshowkindinfo);

        createbtnSignup = new JButton("Create");
        createbtnSignup.addActionListener(this);
        createbtnSignup.addMouseListener(this);
        createbtnSignup.setBorder(new
BevelBorder(BevelBorder.RAISED,
                                        new Color(0, 255, 0), new
Color(0, 255, 51),
                                        new Color(0, 255, 102), new
Color(0, 255, 153)));
        createbtnSignup.setFocusable(false);

    createbtnSignup.setBackground(UIManager.getColor("Button.light
"));
        createbtnSignup.setForeground(new Color(0, 0, 51));
        createbtnSignup.setFont(new Font("Tahoma", Font.BOLD,
30));
        createbtnSignup.setBounds(393, 604, 160, 62);
        signup_pnl.add(createbtnSignup);

        lblClickOnLogin = new JLabel("Click on 'Log In' if you
already have an account ");
        lblClickOnLogin.setOpaque(true);

    lblClickOnLogin.setHorizontalAlignment(SwingConstants.CENTER);
        lblClickOnLogin.setForeground(Color.YELLOW);
        lblClickOnLogin.setFont(new Font("Tahoma", Font.BOLD,
20));
        lblClickOnLogin.setBorder(new
EtchedBorder(EtchedBorder.RAISED, null, null));
        lblClickOnLogin.setBackground(Color.BLUE);
        lblClickOnLogin.setBounds(214, 718, 507, 30);
        signup_pnl.add(lblClickOnLogin);

        waitlbl = new JLabel("...Wait...");
        waitlbl.setVisible(false);
        waitlbl.setFont(new Font("Tahoma", Font.BOLD, 20));
        waitlbl.setHorizontalAlignment(SwingConstants.CENTER);
        waitlbl.setBounds(437, 668, 86, 30);
        signup_pnl.add(waitlbl);
```

```
        login_pnl = new JPanel();
        login_pnl.setVisible(false);
        login_pnl.setBackground(Color.BLUE);
        login_pnl.setBorder(new
SoftBevelBorder(BevelBorder.RAISED, null, null, null, null));
        login_pnl.setBounds(318, 0, 918, 748);
        vma_frm.getContentPane().add(login_pnl);
        login_pnl.setLayout(null);

        waitlbl2 = new JLabel("...Wait...");
        waitlbl2.setForeground(Color.YELLOW);
        waitlbl2.setVisible(false);
        waitlbl2.setFont(new Font("Tahoma", Font.BOLD, 20));
        waitlbl2.setHorizontalAlignment(SwingConstants.CENTER);
        waitlbl2.setBounds(433, 546, 86, 30);
        login_pnl.add(waitlbl2);

        siieo_lbl = new JLabel("Sign In Into Existing One");
        siieo_lbl.setBorder(new BevelBorder(BevelBorder.RAISED,
null, null, null, null));
        siieo_lbl.setFont(new Font("Tahoma", Font.BOLD, 35));
        siieo_lbl.setForeground(new Color(255, 204, 0));
        siieo_lbl.setHorizontalAlignment(SwingConstants.CENTER);
        siieo_lbl.setBounds(179, 144, 579, 84);
        login_pnl.add(siieo_lbl);

        unmlogin = new JLabel("Username  :");
        unmlogin.setOpaque(true);
        unmlogin.setBackground(Color.LIGHT_GRAY);
        unmlogin.setBorder(new
SoftBevelBorder(BevelBorder.LOWERED, null, null, null, null));
        unmlogin.setFont(new Font("Tahoma", Font.BOLD, 30));
        unmlogin.setForeground(new Color(0, 0, 0));
        unmlogin.setHorizontalAlignment(SwingConstants.CENTER);
        unmlogin.setBounds(79, 293, 195, 57);
        login_pnl.add(unmlogin);

        pwdlogin = new JLabel("Password :");
        pwdlogin.setOpaque(true);
        pwdlogin.setBackground(Color.LIGHT_GRAY);
        pwdlogin.setBorder(new
SoftBevelBorder(BevelBorder.LOWERED, null, null, null, null));
        pwdlogin.setHorizontalAlignment(SwingConstants.CENTER);
        pwdlogin.setForeground(new Color(0, 0, 0));
        pwdlogin.setFont(new Font("Tahoma", Font.BOLD, 30));
```

```
        pwdlogin.setBounds(79, 390, 195, 57);
        login_pnl.add(pwdlogin);

        txtpwdlogin = new JPasswordField();
        txtpwdlogin.setBorder(new
EtchedBorder(EtchedBorder.RAISED, null, null));
        txtpwdlogin.setForeground(new Color(51, 255, 0));
        txtpwdlogin.setFont(new Font("Tahoma", Font.BOLD, 30));
        txtpwdlogin.setColumns(30);
        txtpwdlogin.setBounds(284, 390, 540, 57);
        login_pnl.add(txtpwdlogin);

        txtunmlogin = new JTextField();
        txtunmlogin.setBorder(new
EtchedBorder(EtchedBorder.RAISED, null, null));
        txtunmlogin.setForeground(new Color(51, 255, 0));
        txtunmlogin.setFont(new Font("Tahoma", Font.BOLD, 30));
        txtunmlogin.setBounds(284, 293, 540, 57);
        login_pnl.add(txtunmlogin);
        txtunmlogin.setColumns(40);

        letsgologinbtn = new JButton("Let's Go");
        letsgologinbtn.addActionListener(this);
        letsgologinbtn.addMouseListener(this);
        letsgologinbtn.setBorder(new
BevelBorder(BevelBorder.RAISED,
                                                    new
Color(0, 255, 0), new Color(0, 255, 51),
                                                    new
Color(0, 255, 102), new Color(0, 255, 153)));
        letsgologinbtn.setFont(new Font("Tahoma", Font.BOLD,
30));

    letsgologinbtn.setBackground(UIManager.getColor("Button.light"
));
        letsgologinbtn.setForeground(new Color(0, 0, 0));
        letsgologinbtn.setBounds(384, 486, 171, 57);
        login_pnl.add(letsgologinbtn);

        pag_pnl = new JPanel();
        pag_pnl.setBorder(new SoftBevelBorder(BevelBorder.RAISED,
null, null, null, null));
        pag_pnl.setVisible(false);
        pag_pnl.setLayout(null);
        pag_pnl.setBackground(new Color(0, 255, 255));
```

```
        pag_pnl.setBounds(318, 0, 918, 748);
        vma_frm.getContentPane().add(pag_pnl);

        waitlbl3 = new JLabel("...Wait...");
        waitlbl3.setForeground(Color.YELLOW);
        waitlbl3.setVisible(false);
        waitlbl3.setFont(new Font("Tahoma", Font.BOLD, 20));
        waitlbl3.setHorizontalAlignment(SwingConstants.CENTER);
        waitlbl3.setBounds(418, 464, 86, 30);
        pag_pnl.add(waitlbl3);

        gias = new JLabel("Give It A Shot");
        gias.setBorder(new BevelBorder(BevelBorder.RAISED, null,
null, null, null));
        gias.setForeground(new Color(0, 0, 51));
        gias.setFont(new Font("Tahoma", Font.BOLD, 35));
        gias.setHorizontalAlignment(SwingConstants.CENTER);
        gias.setBounds(203, 115, 497, 88);
        pag_pnl.add(gias);

        pname = new JLabel("Player Name");
        pname.setOpaque(true);
        pname.setBackground(Color.LIGHT_GRAY);
        pname.setForeground(new Color(0, 0, 51));
        pname.setFont(new Font("Tahoma", Font.BOLD, 25));
        pname.setHorizontalAlignment(SwingConstants.CENTER);
        pname.setBounds(352, 244, 213, 51);
        pag_pnl.add(pname);

        txtnmpag = new JTextField();
        txtnmpag.setBorder(new
SoftBevelBorder(BevelBorder.LOWERED, null, null, null, null));
        txtnmpag.setForeground(new Color(51, 255, 0));
        txtnmpag.setFont(new Font("Tahoma", Font.BOLD, 25));
        txtnmpag.setBounds(235, 305, 431, 51);
        pag_pnl.add(txtnmpag);
        txtnmpag.setColumns(10);

        pagletsgobtn = new JButton("Let's Go");
        pagletsgobtn.addMouseListener(this);
        pagletsgobtn.addActionListener(this);
        pagletsgobtn.setForeground(new Color(0, 0, 51));
        pagletsgobtn.setFocusable(false);
        pagletsgobtn.setBorder(new
BevelBorder(BevelBorder.LOWERED,
```

```
                                                    new Color(0,
255, 0), new Color(0, 255, 51),
                                                    new Color(0,
255, 102), new Color(0, 255, 153)));

     pagletsgobtn.setBackground(UIManager.getColor("Button.light"))
;
          pagletsgobtn.setFont(new Font("Tahoma", Font.BOLD, 30));
          pagletsgobtn.setBounds(379, 397, 155, 57);
          pag_pnl.add(pagletsgobtn);

          vma_frm.setVisible(true);
     }

     @Override
     public void actionPerformed(ActionEvent e) {
          if(e.getSource()==signupbtn) {
               signup_pnl.setVisible(true);
               login_pnl.setVisible(false);
               pag_pnl.setVisible(false);
          }
          if(e.getSource()==createbtnSignup) {
               sgunm=txtunmSignup.getText();
               sgpwd=txtpwdSignup.getText();
               sggm=txtgmailSignup.getText();
               sgag=txtagesignup.getText();
               if((sgunm.length()==0) || (sgpwd.length()==0) ||
                  (sggm.length()==0) || (sgag.length()==0)) {
                    JOptionPane.showMessageDialog(signup_pnl,
                                                    "Please fill
all the fields", "Short Name",

JOptionPane.ERROR_MESSAGE);
               }
               else if(sgpwd.length()<5) {
                    JOptionPane.showMessageDialog(signup_pnl,
                                                    "Password
Length Must Be At Least 5 Characters",
                                                    "Short
Password", JOptionPane.ERROR_MESSAGE);
               }
               else {
                    waitlbl.setVisible(true);
                    datasignup();
                    email=false;
```

```
                    JOptionPane.showMessageDialog(signup_pnl,
"Created Successfully");
                    signup_pnl.setVisible(false);
                    login_pnl.setVisible(true);
                    txtunmSignup.setText(null);
                    txtpwdSignup.setText(null);
                    txtgmailSignup.setText(null);
                    txtagesignup.setText(null);
                    waitlbl.setVisible(false);
               }
          }

          if(e.getSource()==loginbtn) {
               signup_pnl.setVisible(false);
               login_pnl.setVisible(true);
               pag_pnl.setVisible(false);
          }
          if(e.getSource()==letsgologinbtn) {
               waitlbl2.setVisible(true);
               lognm=txtunmlogin.getText();
               logpwd=txtpwdlogin.getText();
               if(lognm.length()==0 || logpwd.length()==0) {
                    JOptionPane.showMessageDialog(pag_pnl, "Please
fill both of the fields",
                                                       "null field",
JOptionPane.ERROR_MESSAGE);
               }
               else {
                    ChopStick.t2=false;
                    datalogin();
                    try {
                         set=new BufferedWriter(new
FileWriter("settings.txt"));
                         set.close();
                    }
                    catch(Exception e1) {
                         e1.printStackTrace();
                    }
                    waitlbl2.setVisible(false);
               }

          }

          if(e.getSource()==pagbtn) {
               signup_pnl.setVisible(false);
```

```
                login_pnl.setVisible(false);
                pag_pnl.setVisible(true);
           }
           if(e.getSource()==pagletsgobtn) {
                waitlbl3.setVisible(true);
                pagname=txtnmpag.getText();
                if(pagname.length()==0) {
                     JOptionPane.showMessageDialog(pag_pnl, "Field
can not be empty", "Short Name",

     JOptionPane.WARNING_MESSAGE);
                }
                else {
                     datapag();
                     pag=true;
                     try {
                          vma_frm.dispose();
                          ChopStick cs=new ChopStick();
                          cs.name.setText(pagname);
                          cs.main_frm.setVisible(true);
                          ChopStick.t2=false;
                          set=new BufferedWriter(new
FileWriter("settings.txt"));
                          set.close();
                     }catch(Exception e1) {
                          e1.printStackTrace();
                     }
                     waitlbl3.setVisible(false);
                }
           }

     }

     @Override
     public void mouseClicked(MouseEvent e) {

     }

     @Override
     public void mousePressed(MouseEvent e) {

     }

     @Override
     public void mouseReleased(MouseEvent e) {
```

```
    }

    @Override
    public void mouseEntered(MouseEvent e) {
        if(e.getSource()==signupbtn) {

    signupbtn.setBackground(UIManager.getColor("Button.light"));
        }
        if(e.getSource()==loginbtn) {

    loginbtn.setBackground(UIManager.getColor("Button.light"));
        }
        if(e.getSource()==pagbtn) {

    pagbtn.setBackground(UIManager.getColor("Button.light"));
        }
        if(e.getSource()==createbtnSignup) {
            createbtnSignup.setBackground(Color.LIGHT_GRAY);
        }
        if(e.getSource()==letsgologinbtn) {
            letsgologinbtn.setBackground(Color.LIGHT_GRAY);
        }
        if(e.getSource()==pagletsgobtn) {
            pagletsgobtn.setBackground(Color.LIGHT_GRAY);
        }

    }

    @Override
    public void mouseExited(MouseEvent e) {
        if(e.getSource()==signupbtn) {
            signupbtn.setBackground(new Color(255, 153, 153));
        }
        if(e.getSource()==loginbtn) {
            loginbtn.setBackground(new Color(255, 153, 153));
        }
        if(e.getSource()==pagbtn) {
            pagbtn.setBackground(new Color(255, 153, 153));
        }
        if(e.getSource()==createbtnSignup) {

    createbtnSignup.setBackground(UIManager.getColor("Button.light
"));
        }
```

```
            if(e.getSource()==letsgologinbtn) {

        letsgologinbtn.setBackground(UIManager.getColor("Button.light"
));
            }
            if(e.getSource()==pagletsgobtn) {

        pagletsgobtn.setBackground(UIManager.getColor("Button.light"))
;
            }

        }

}
```

ChopStick.java

```
package sahu.chopstick_game;

import javax.swing.JFrame;
import javax.swing.JButton;
import java.awt.Font;
import java.awt.HeadlessException;
import java.awt.MediaTracker;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStream;
import java.net.URL;
import java.util.Random;
import java.awt.event.ActionEvent;
import java.applet.Applet;
import java.applet.AudioClip;
```

```
import java.awt.Color;
import java.awt.Component;
import java.awt.Toolkit;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.print.attribute.standard.Media;
import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.swing.BorderFactory;
import javax.swing.ImageIcon;
import javax.swing.border.BevelBorder;
import javax.swing.UIManager;
import javax.swing.border.SoftBevelBorder;

import org.apache.commons.mail.EmailException;

import javax.swing.SwingConstants;
import javax.swing.JPanel;
import javax.swing.border.EtchedBorder;
import javax.swing.border.LineBorder;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.*;



@SuppressWarnings({ "unused", "removal" })
public class ChopStick implements
MouseListener,Runnable,ActionListener,KeyListener {

    public JFrame main_frm;

    public JButton pwc_btn,
                   pvp_btn,
                   Howtoplay,
                   Settings,
                  Aboutgame,
                   feedback,
                   profile,
                   signout;

    public JLabel name,
                  playerpic,
                  main_lbl;
```

```
    public static JLabel vm;

    public boolean fb;//send feedback
    public static boolean t2=true;//label text animation

    public String s;//feedback string

    static BufferedReader won=null,lost=null,player=null;
    static BufferedWriter player0=null,wset;

   public static Clip c=null;

    public static void main(String[] args) throws Exception{



                        Launch l=new Launch();
                        try {
                             ChopStick.bgmusic();
                        }catch(Exception e) {
                             JOptionPane.showMessageDialog(l,
e.getMessage());
                        }
                        Thread.sleep(10);
                        l.setVisible(true);
                        wset=new BufferedWriter(new
FileWriter("settings.txt"));
                        l.prog();
                        Thread.sleep(10);
                        l.dispose();
                        File f=new File("player.txt");
                        if(f.exists()) {
                             t2=false;
                             new
ChopStick().main_frm.setVisible(true);
                        }
                        else {
                             new Void_matrix_account();
                        }
                        while(t2) {
                             Thread.sleep(1);
                        }
                        paint();
    }
```

```
    public static void bgmusic() throws Exception{
        InputStream f=ChopStick.class.getResourceAsStream("We Got
What You Want.wav");
        InputStream bf=new BufferedInputStream(f);
        AudioInputStream ai=AudioSystem.getAudioInputStream(bf);
        c=AudioSystem.getClip();
        c.open(ai);
        c.start();
        c.loop(100);
    }

    static void paint() throws Exception{
        Random r=new Random();
        while(true) {
            int n1=r.nextInt(256);
            int n2=r.nextInt(256);
            int n3=r.nextInt(256);
            vm.setForeground(new Color(n1,n2,n3));
            Thread.sleep(100);
        }
    }

    public ChopStick() throws Exception {
        initialize();
        new Thread(this).start();

    }

    @Override
    public void run() {
        Send();
    }

    public void Send() {
        fb=true;
        while(fb) {
            try { Thread.sleep(1000); } catch(Exception e) {
e.printStackTrace(); }
        }
        try {
            Mails.getFeedBack(s,name.getText());
            Send();
        }catch(Exception e) {
            e.printStackTrace();
        }
```

```
    }

    private void initialize() throws Exception {
        main_frm = new JFrame();
        main_frm.setResizable(false);
        main_frm.getContentPane().setBackground(new Color(153,
102, 255));
        main_frm.getContentPane().setLayout(null);
        main_frm.setForeground(Color.BLACK);

    main_frm.setIconImage(Toolkit.getDefaultToolkit().getImage(get
Class().getResource("logo_F.png")));
        main_frm.setBackground(Color.BLACK);
        main_frm.setFont(new Font("Dialog", Font.BOLD, 20));
        main_frm.setTitle("ChopStick");
        main_frm.setBounds(150, 20, 1250, 785);
        main_frm.getContentPane().setFocusable(true);
        main_frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        pwc_btn = new JButton("Play Against Computer");
        pwc_btn.addActionListener(this);
        pwc_btn.addMouseListener(this);
        pwc_btn.addKeyListener(this);
        pwc_btn.setToolTipText("by clicking on this you will have
the game started playing against computer");
        pwc_btn.setForeground(new Color(0, 0, 0));
        pwc_btn.setFont(new Font("Tahoma", Font.BOLD, 20));

    pwc_btn.setBackground(UIManager.getColor("Button.light"));
        pwc_btn.setBounds(98, 76, 253, 76);
        pwc_btn.setFocusable(false);
        pwc_btn.setBorder(new BevelBorder(BevelBorder.RAISED,
                                          new Color(0, 153,
255), new Color(0, 255, 255),
                                          new Color(51, 153,
255), new Color(51, 255, 255)));
        main_frm.getContentPane().add(pwc_btn);

        pvp_btn = new JButton("Two Players");
        pvp_btn.addActionListener(this);
        pvp_btn.addMouseListener(this);
        pvp_btn.addKeyListener(this);
        pvp_btn.setToolTipText("by clicking on this two players
can play against each other");
        pvp_btn.setForeground(new Color(0, 0, 0));
```

```
        pvp_btn.setFont(new Font("Tahoma", Font.BOLD, 20));

    pvp_btn.setBackground(UIManager.getColor("Button.light"));
        pvp_btn.setBounds(98, 336, 253, 76);
        pvp_btn.setFocusable(false);
        pvp_btn.setBorder(new BevelBorder(BevelBorder.RAISED,
                                                new Color(0, 153,
255), new Color(0, 255, 255),
                                                new Color(51, 153,
255), new Color(51, 255, 255)));
        main_frm.getContentPane().add(pvp_btn);

        Howtoplay = new JButton("How To Play");
        Howtoplay.addActionListener(this);
        Howtoplay.addMouseListener(this);
        Howtoplay.addKeyListener(this);
        Howtoplay.setToolTipText("give a look at rules for better
understanding of gameplay");
        Howtoplay.setForeground(Color.BLACK);
        Howtoplay.setFont(new Font("Tahoma", Font.BOLD, 20));
        Howtoplay.setFocusable(false);
        Howtoplay.setBorder(new BevelBorder(BevelBorder.RAISED,
                                                new Color(0,
153, 255), new Color(0, 255, 255),
                                                new Color(51,
153, 255), new Color(51, 255, 255)));

    Howtoplay.setBackground(UIManager.getColor("Button.light"));
        Howtoplay.setBounds(98, 591, 253, 76);
        main_frm.getContentPane().add(Howtoplay);

        feedback = new JButton("Feedback");
        feedback.addActionListener(this);
        feedback.addMouseListener(this);
        feedback.addKeyListener(this);
        feedback.setForeground(Color.BLACK);
        feedback.setFont(new Font("Tahoma", Font.BOLD, 20));
        feedback.setFocusable(false);
        feedback.setBorder(new BevelBorder(BevelBorder.RAISED,
                                                new Color(0, 153,
255), new Color(0, 255, 255),
                                                new Color(51,
153, 255), new Color(51, 255, 255)));

    feedback.setBackground(UIManager.getColor("Button.light"));
```

```
        feedback.setBounds(878, 591, 253, 76);
        feedback.setToolTipText("let me know what i can
improve");
        main_frm.getContentPane().add(feedback);

        Settings = new JButton("Settings");
        Settings.addActionListener(this);
        Settings.addMouseListener(this);
        Settings.addKeyListener(this);
        Settings.setForeground(Color.BLACK);
        Settings.setFont(new Font("Tahoma", Font.BOLD, 20));
        Settings.setFocusable(false);
        Settings.setBorder(new BevelBorder(BevelBorder.RAISED,
                                                new Color(0, 153,
255), new Color(0, 255, 255),
                                                new Color(51,
153, 255), new Color(51, 255, 255)));

    Settings.setBackground(UIManager.getColor("Button.light"));
        Settings.setBounds(878, 76, 253, 76);
        main_frm.getContentPane().add(Settings);

        Aboutgame = new JButton("About Game");
        Aboutgame.addActionListener(this);
        Aboutgame.addMouseListener(this);
        Aboutgame.addKeyListener(this);
        Aboutgame.setToolTipText("let's get your hands dirty in
exploring this game");
        Aboutgame.setForeground(Color.BLACK);
        Aboutgame.setFont(new Font("Tahoma", Font.BOLD, 20));
        Aboutgame.setFocusable(false);
        Aboutgame.setBorder(new BevelBorder(BevelBorder.RAISED,
                                                new Color(0,
153, 255), new Color(0, 255, 255),
                                                new Color(51,
153, 255), new Color(51, 255, 255)));

    Aboutgame.setBackground(UIManager.getColor("Button.light"));
        Aboutgame.setBounds(877, 336, 253, 76);
        main_frm.getContentPane().add(Aboutgame);

        profile = new JButton("Profile");
        profile.addActionListener(this);
        profile.addMouseListener(this);
        profile.addKeyListener(this);
```

```
        profile.setForeground(Color.BLACK);
        profile.setFont(new Font("Tahoma", Font.BOLD, 20));
        profile.setFocusable(false);
        profile.setBorder(new BevelBorder(BevelBorder.RAISED,
                                                new Color(0, 153,
255), new Color(0, 255, 255),
                                                new Color(51, 153,
255), new Color(51, 255, 255)));

    profile.setBackground(UIManager.getColor("Button.light"));
        profile.setBounds(485, 444, 253, 76);
        profile.setToolTipText("let's see what you're up to");
        main_frm.getContentPane().add(profile);

        signout = new JButton("Sign Out");
        signout.addActionListener(this);
        signout.addMouseListener(this);
        signout.addKeyListener(this);
        signout.setToolTipText("");
        signout.setForeground(Color.BLACK);
        signout.setFont(new Font("Tahoma", Font.BOLD, 20));
        signout.setFocusable(false);
        signout.setBorder(new BevelBorder(BevelBorder.RAISED,
                                                new Color(0, 153,
255), new Color(0, 255, 255),
                                                new Color(51, 153,
255), new Color(51, 255, 255)));

    signout.setBackground(UIManager.getColor("Button.light"));
        signout.setBounds(485, 30, 253, 76);
        main_frm.getContentPane().add(signout);

        player=new BufferedReader(new FileReader("player.txt"));
        name = new JLabel(player.readLine());
        player.close();
        name.setFont(new Font("Tahoma", Font.BOLD, 25));
        name.setHorizontalAlignment(SwingConstants.CENTER);
        name.setBounds(354, 196, 502, 86);
        main_frm.getContentPane().add(name);

        playerpic = new JLabel();
        playerpic.setIcon(new
ImageIcon(getClass().getResource("user3.png")));
        playerpic.setHorizontalAlignment(SwingConstants.CENTER);
        playerpic.setBounds(502, 261, 212, 182);
```

```
        main_frm.getContentPane().add(playerpic);

        vm = new JLabel("Void_Matrix[:]");
        vm.setFont(new Font("Tahoma", Font.BOLD, 27));
        vm.setForeground(Color.BLACK);
        vm.setHorizontalAlignment(SwingConstants.CENTER);
        vm.setBounds(485, 659, 263, 61);
        main_frm.getContentPane().add(vm);

        main_lbl = new JLabel();
        main_lbl.setBorder(new
SoftBevelBorder(BevelBorder.RAISED, null, null, null, null));
        main_lbl.setForeground(new Color(153, 0, 102));
        main_lbl.setIcon(new
ImageIcon(getClass().getResource("chopstick_main_cs.jpg")));
        main_lbl.setBounds(0, 0, 1236, 750);
        main_lbl.addMouseListener(this);
        main_frm.getContentPane().add(main_lbl);


    }

    @Override
    public void actionPerformed(ActionEvent e) {

        if(e.getSource()==pwc_btn)
        {
            main_frm.dispose();
            try {
                new PWC();
            } catch (Exception e1) {
                e1.printStackTrace();
            }
        }

        if(e.getSource()==pvp_btn)
        {
            main_frm.dispose();
            try {
                new PVP();
            } catch (Exception e1) {
                e1.printStackTrace();
            }
        }
```

```
          if(e.getSource()==Howtoplay)
          {
               main_frm.dispose();
               new How_To_Play();
          }

          if(e.getSource()==Settings)
          {
               main_frm.dispose();
               new Settings();
          }

          if(e.getSource()==Aboutgame)
          {
               main_frm.dispose();
               new About_Game();
          }

          if(e.getSource()==feedback)
          {
               s=JOptionPane.showInputDialog(feedback,
                         "Tell us about your experience\n or any
suggetion", "Feedback",
                         JOptionPane.QUESTION_MESSAGE);
               fb=false;
          }

          if(e.getSource()==profile)
          {
               try {
                    won=new BufferedReader(new
FileReader("won.txt"));
                    lost=new BufferedReader(new
FileReader("lost.txt"));
                    int
iw=(int)won.read(),il=(int)lost.read(),it=iw+il;
                    String
w=String.valueOf(iw),l=String.valueOf(il),t=String.valueOf(it);
                    JOptionPane.showMessageDialog(profile,
name.getText()+
                              "\n(Against Computer)\nWon:
"+w+"\nLost: "+l+"\nTotal: "+t, "Profile",
                              JOptionPane.INFORMATION_MESSAGE);
                    won.close();
                    lost.close();
```

```
                } catch(Exception e1) {
                    e1.printStackTrace();
                }
            }

            if(e.getSource()==signout)
            {
                int x=JOptionPane.showConfirmDialog(main_frm,
                        "Are You Sure You wanna Sign Out!", "Sign
Out",
                        JOptionPane.YES_NO_OPTION,
JOptionPane.QUESTION_MESSAGE);
                if(x==0) {
                    main_frm.dispose();
                    new Void_matrix_account();
                }
            }
        }

        @Override
        public void mouseEntered(MouseEvent e) {

            if(e.getSource()==pwc_btn)
            {
                pwc_btn.setBackground(Color.black);
                pwc_btn.setForeground(Color.green);
            }
            if(e.getSource()==pvp_btn)
            {
                pvp_btn.setBackground(Color.black);
                pvp_btn.setForeground(Color.green);
            }
            if(e.getSource()==Howtoplay)
            {
                Howtoplay.setBackground(Color.black);
                Howtoplay.setForeground(Color.green);
            }
            if(e.getSource()==Settings)
            {
                Settings.setBackground(Color.black);
                Settings.setForeground(Color.green);
            }
            if(e.getSource()==Aboutgame)
            {
                Aboutgame.setBackground(Color.black);
```

```
            Aboutgame.setForeground(Color.green);
        }
        if(e.getSource()==profile)
        {
            profile.setBackground(Color.black);
            profile.setForeground(Color.green);
        }
        if(e.getSource()==feedback)
        {
            feedback.setBackground(Color.black);
            feedback.setForeground(Color.green);
        }
        if(e.getSource()==signout)
        {
            signout.setBackground(Color.black);
            signout.setForeground(Color.green);
        }
    }


    @Override
    public void mouseExited(MouseEvent e) {

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

    pwc_btn.setBackground(UIManager.getColor("Button.light"));
            pwc_btn.setForeground(Color.black);
        }
        if(e.getSource()==pvp_btn)
        {

    pvp_btn.setBackground(UIManager.getColor("Button.light"));
            pvp_btn.setForeground(Color.black);
        }
        if(e.getSource()==Howtoplay)
        {

    Howtoplay.setBackground(UIManager.getColor("Button.light"));
            Howtoplay.setForeground(Color.black);
        }
        if(e.getSource()==Settings)
        {

    Settings.setBackground(UIManager.getColor("Button.light"));
```

```
            Settings.setForeground(Color.black);
        }
        if(e.getSource()==Aboutgame)
        {

Aboutgame.setBackground(UIManager.getColor("Button.light"));
            Aboutgame.setForeground(Color.black);
        }
        if(e.getSource()==profile)
        {

profile.setBackground(UIManager.getColor("Button.light"));
            profile.setForeground(Color.black);
        }
        if(e.getSource()==feedback)
        {

feedback.setBackground(UIManager.getColor("Button.light"));
            feedback.setForeground(Color.black);
        }
        if(e.getSource()==signout)
        {

signout.setBackground(UIManager.getColor("Button.light"));
            signout.setForeground(Color.black);
        }
    }

    @Override
    public void mouseClicked(MouseEvent e) {

    }
    @Override
    public void mousePressed(MouseEvent e) {

    }
    @Override
    public void mouseReleased(MouseEvent e) {

    }


    @Override
    public void keyTyped(KeyEvent e) {
```

```
    }

    @Override
    public void keyPressed(KeyEvent e) {

    }

    @Override
    public void keyReleased(KeyEvent e) {

    }

}
```

PWC.java

```
package sahu.chopstick_game;

import javax.swing.JFrame;
import javax.swing.JButton;
import java.awt.Font;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.awt.event.ActionEvent;
import javax.swing.JPanel;
import java.awt.Toolkit;
import java.awt.Color;

import javax.swing.JLabel;
import javax.swing.JOptionPane;
```

```
import javax.swing.ImageIcon;
import javax.swing.border.CompoundBorder;

import org.apache.commons.mail.EmailException;

import javax.swing.border.BevelBorder;
import javax.swing.SwingConstants;

import java.util.Random;
import java.util.Vector;

import javax.swing.JTextArea;

@SuppressWarnings("unused")
public class PWC implements ActionListener,MouseListener,Runnable{

     private JFrame pwc_frm;

     private JButton quit_btn,
                     p_l_btn,
                     p_r_btn,
                     cp_r_btn,
                     cp_l_btn,
                     split_btn,
                     attack_btn,
                     done_btn;

     private JPanel l_pnl,
                     r_pnl,
                     main_pnl;

     private JLabel p_l_fist,
                     p_l_five,
                     p_r_fist,
                     p_l_one,
                     p_l_two,
                     p_l_three,
                     p_l_four,
                     p_r_five,
                     p_r_three,
                     p_r_four,
                     p_r_two,
                     p_r_one,
                     cp_r_fist,
                     cp_r_five,
```

```
                        cp_r_four,
                        cp_r_three,
                        cp_r_two,
                        cp_r_one,
                        cp_l_fist,
                        cp_l_five,
                        cp_l_four,
                        cp_l_three,
                        cp_l_one,
                        cp_l_two,
                        w_l;

    private JTextArea msg;

    Random r=new Random();

    private boolean running=true,turn,attack,split,p_l,p_r,t=true;

    private Vector<Integer> p_l_h=new Vector<Integer>(5),
                        p_r_h=new Vector<Integer>(5),
                        cp_l_h=new Vector<Integer>(5),
                        cp_r_h=new Vector<Integer>(5);

    public int cp_ch,
                    cp_ch_p_h,
                    cp_ch_c_h,
                    i,cp1,cp2,prpr,prpl,p,
                    cph,rc,prcpl,prcpr,x,wcount,lcount;

    public Connection c;
    public Statement st;

    BufferedReader br=null;
    BufferedWriter bw=null;

    public void delay(int n) {
    try {
              Thread.sleep(n);
         } catch (InterruptedException e) {
              e.printStackTrace();
         }
    }

    public PWC() throws Exception{
         br=new BufferedReader(new FileReader("settings.txt"));
```

```
            initialize();
            Thread t=new Thread(this);
            t.start();
        }

        @Override
        public void run() {
            delay(1000);
            try {
                Gameloop();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        private synchronized void initialize() throws Exception{
            pwc_frm = new JFrame();
            pwc_frm.setTitle("Playing With Computer");

        pwc_frm.setIconImage(Toolkit.getDefaultToolkit().getImage(getC
lass().getResource("logo_F.png")));
            pwc_frm.setResizable(false);
            pwc_frm.setBounds(150, 20, 1250, 785);
            pwc_frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            pwc_frm.setUndecorated(true);
            pwc_frm.getContentPane().setLayout(null);

            quit_btn = new JButton("Quit");
            quit_btn.addActionListener(this);
            quit_btn.setForeground(new Color(51, 0, 153));
            quit_btn.setFont(new Font("Tahoma", Font.BOLD, 30));
            quit_btn.setBackground(new Color(255, 204, 0));
            quit_btn.setBounds(10, 721, 138, 56);
            quit_btn.setFocusable(false);
            quit_btn.addMouseListener(this);
            pwc_frm.getContentPane().add(quit_btn);

            r_pnl = new JPanel();
            r_pnl.setBorder(new CompoundBorder(new
BevelBorder(BevelBorder.RAISED,
                                                    new
Color(0, 0, 128), new Color(0, 0, 205),
                                                    new
Color(0, 0, 139), new Color(0, 0, 255)),
```

```
                                                    new
BevelBorder(BevelBorder.LOWERED,
                                                        new
Color(0, 0, 205), new Color(0, 0, 255),
                                                        new
Color(0, 0, 128), new Color(0, 0, 139))));

          r_pnl.setBounds(670, 50, 516, 447);
          pwc_frm.getContentPane().add(r_pnl);
          r_pnl.setLayout(null);

          cp_r_btn = new JButton("Right");
          cp_r_btn.setForeground(new Color(0, 255, 0));
          cp_r_btn.addActionListener(this);
          cp_r_btn.setEnabled(false);
          cp_r_btn.setFont(new Font("Tahoma", Font.BOLD, 25));
          cp_r_btn.setBackground(new Color(0, 0, 204));
          cp_r_btn.setBounds(319, 100, 114, 45);
          cp_r_btn.setFocusable(false);
          cp_r_btn.addMouseListener(this);
          r_pnl.add(cp_r_btn);

          cp_l_btn = new JButton("Left");
          cp_l_btn.setForeground(new Color(0, 255, 0));
          cp_l_btn.addActionListener(this);
          cp_l_btn.setEnabled(false);
          cp_l_btn.setFont(new Font("Tahoma", Font.BOLD, 25));
          cp_l_btn.setBackground(new Color(51, 51, 153));
          cp_l_btn.setBounds(319, 291, 114, 45);
          cp_l_btn.setFocusable(false);
          cp_l_btn.addMouseListener(this);
          r_pnl.add(cp_l_btn);

          cp_r_fist = new JLabel("");
          cp_r_fist.setIcon(new
ImageIcon(getClass().getResource("cp_r_fist.png")));
          cp_r_fist.setHorizontalAlignment(SwingConstants.CENTER);
          cp_r_fist.setBounds(105, 61, 130, 128);
          cp_r_fist.setVisible(false);
          r_pnl.add(cp_r_fist);

          cp_r_five = new JLabel("");
          cp_r_five.setIcon(new
ImageIcon(getClass().getResource("cp_r_five.png")));
          cp_r_five.setHorizontalAlignment(SwingConstants.CENTER);
```

```
        cp_r_five.setBounds(105, 61, 130, 128);
        cp_r_five.setVisible(false);
        r_pnl.add(cp_r_five);

        cp_r_four = new JLabel("");
        cp_r_four.setIcon(new
ImageIcon(getClass().getResource("cp_r_four.png")));
        cp_r_four.setHorizontalAlignment(SwingConstants.CENTER);
        cp_r_four.setBounds(105, 61, 130, 128);
        cp_r_four.setVisible(false);
        r_pnl.add(cp_r_four);

        cp_r_three = new JLabel("");
        cp_r_three.setIcon(new
ImageIcon(getClass().getResource("cp_r_three.png")));
        cp_r_three.setHorizontalAlignment(SwingConstants.CENTER);
        cp_r_three.setBounds(105, 61, 130, 128);
        cp_r_three.setVisible(false);
        r_pnl.add(cp_r_three);

        cp_r_two = new JLabel("");
        cp_r_two.setIcon(new
ImageIcon(getClass().getResource("cp_r_two.png")));
        cp_r_two.setHorizontalAlignment(SwingConstants.CENTER);
        cp_r_two.setBounds(105, 61, 130, 128);
        cp_r_two.setVisible(false);
        r_pnl.add(cp_r_two);

        cp_r_one = new JLabel("");
        cp_r_one.setIcon(new
ImageIcon(getClass().getResource("cp_r_one.png")));
        cp_r_one.setHorizontalAlignment(SwingConstants.CENTER);
        cp_r_one.setBounds(105, 61, 130, 128);
        r_pnl.add(cp_r_one);

        cp_l_fist = new JLabel("");
        cp_l_fist.setIcon(new
ImageIcon(getClass().getResource("cp_l_fist.png")));
        cp_l_fist.setHorizontalAlignment(SwingConstants.CENTER);
        cp_l_fist.setBounds(105, 252, 130, 128);
        cp_l_fist.setVisible(false);
        r_pnl.add(cp_l_fist);

        cp_l_five = new JLabel("");
```

```
        cp_l_five.setIcon(new
ImageIcon(getClass().getResource("cp_l_five.png")));
        cp_l_five.setHorizontalAlignment(SwingConstants.CENTER);
        cp_l_five.setBounds(105, 252, 130, 128);
        cp_l_five.setVisible(false);
        r_pnl.add(cp_l_five);

        cp_l_four = new JLabel("");
        cp_l_four.setIcon(new
ImageIcon(getClass().getResource("cp_l_four.png")));
        cp_l_four.setHorizontalAlignment(SwingConstants.CENTER);
        cp_l_four.setBounds(105, 252, 130, 128);
        cp_l_four.setVisible(false);
        r_pnl.add(cp_l_four);

        cp_l_three = new JLabel("");
        cp_l_three.setIcon(new
ImageIcon(getClass().getResource("cp_l_three.png")));
        cp_l_three.setHorizontalAlignment(SwingConstants.CENTER);
        cp_l_three.setBounds(105, 252, 130, 128);
        cp_l_three.setVisible(false);
        r_pnl.add(cp_l_three);

        cp_l_two = new JLabel("");
        cp_l_two.setIcon(new
ImageIcon(getClass().getResource("cp_l_two.png")));
        cp_l_two.setHorizontalAlignment(SwingConstants.CENTER);
        cp_l_two.setBounds(105, 252, 130, 128);
        cp_l_two.setVisible(false);
        r_pnl.add(cp_l_two);

        cp_l_one = new JLabel("");
        cp_l_one.setIcon(new
ImageIcon(getClass().getResource("cp_l_one.png")));
        cp_l_one.setHorizontalAlignment(SwingConstants.CENTER);
        cp_l_one.setBounds(105, 252, 130, 128);
        r_pnl.add(cp_l_one);

        l_pnl = new JPanel();
        l_pnl.setLayout(null);
        l_pnl.setBorder(new CompoundBorder(new
BevelBorder(BevelBorder.RAISED,
                                                    new
Color(0, 0, 128), new Color(0, 0, 205),
```

```
                                                                new
Color(0, 0, 139), new Color(0, 0, 255)),
                                                            new
BevelBorder(BevelBorder.LOWERED,
                                                                new
Color(0, 0, 205), new Color(0, 0, 255),
                                                                new
Color(0, 0, 128), new Color(0, 0, 139))));
          l_pnl.setBounds(50, 50, 516, 447);
          pwc_frm.getContentPane().add(l_pnl);

          p_l_btn = new JButton("Left");
          p_l_btn.setForeground(new Color(0, 255, 0));
          p_l_btn.addActionListener(this);
          p_l_btn.setEnabled(false);
          p_l_btn.setBackground(new Color(51, 51, 153));
          p_l_btn.setFont(new Font("Tahoma", Font.BOLD, 25));
          p_l_btn.setBounds(85, 100, 114, 45);
          p_l_btn.setFocusable(false);
          p_l_btn.addMouseListener(this);
          l_pnl.add(p_l_btn);

          p_r_btn = new JButton("Right");
          p_r_btn.setForeground(new Color(0, 255, 0));
          p_r_btn.addActionListener(this);
          p_r_btn.setEnabled(false);
          p_r_btn.setFont(new Font("Tahoma", Font.BOLD, 25));
          p_r_btn.setBackground(new Color(0, 0, 204));
          p_r_btn.setBounds(85, 291, 114, 45);
          p_r_btn.setFocusable(false);
          p_r_btn.addMouseListener(this);
          l_pnl.add(p_r_btn);

          p_l_fist = new JLabel("");
          p_l_fist.setIcon(new
ImageIcon(getClass().getResource("p_l_fist.png")));
          p_l_fist.setHorizontalAlignment(SwingConstants.CENTER);
          p_l_fist.setBounds(299, 61, 130, 128);
          p_l_fist.setVisible(false);
          l_pnl.add(p_l_fist);

          p_l_five = new JLabel("");
          p_l_five.setIcon(new
ImageIcon(getClass().getResource("p_l_five.png")));
          p_l_five.setHorizontalAlignment(SwingConstants.CENTER);
```

```
        p_l_five.setBounds(299, 61, 130, 128);
        p_l_five.setVisible(false);
        l_pnl.add(p_l_five);

        p_l_four = new JLabel("");
        p_l_four.setIcon(new
ImageIcon(getClass().getResource("p_l_four.png")));
        p_l_four.setHorizontalAlignment(SwingConstants.CENTER);
        p_l_four.setBounds(299, 61, 130, 128);
        p_l_four.setVisible(false);
        l_pnl.add(p_l_four);

        p_l_three = new JLabel("");
        p_l_three.setIcon(new
ImageIcon(getClass().getResource("p_l_three.png")));
        p_l_three.setHorizontalAlignment(SwingConstants.CENTER);
        p_l_three.setBounds(299, 61, 130, 128);
        p_l_three.setVisible(false);
        l_pnl.add(p_l_three);

        p_l_two = new JLabel("");
        p_l_two.setIcon(new
ImageIcon(getClass().getResource("p_l_two.png")));
        p_l_two.setHorizontalAlignment(SwingConstants.CENTER);
        p_l_two.setBounds(299, 61, 130, 128);
        p_l_two.setVisible(false);
        l_pnl.add(p_l_two);

        p_l_one = new JLabel("");
        p_l_one.setIcon(new
ImageIcon(getClass().getResource("p_l_one.png")));
        p_l_one.setHorizontalAlignment(SwingConstants.CENTER);
        p_l_one.setBounds(299, 61, 130, 128);
        l_pnl.add(p_l_one);

        p_r_fist = new JLabel("");
        p_r_fist.setIcon(new
ImageIcon(getClass().getResource("p_r_fist.png")));
        p_r_fist.setHorizontalAlignment(SwingConstants.CENTER);
        p_r_fist.setBounds(299, 252, 130, 128);
        p_r_fist.setVisible(false);
        l_pnl.add(p_r_fist);

        p_r_five = new JLabel("");
```

```
        p_r_five.setIcon(new
ImageIcon(getClass().getResource("p_r_five.png")));
        p_r_five.setHorizontalAlignment(SwingConstants.CENTER);
        p_r_five.setBounds(299, 252, 130, 128);
        p_r_five.setVisible(false);
        l_pnl.add(p_r_five);

        p_r_four = new JLabel("");
        p_r_four.setIcon(new
ImageIcon(getClass().getResource("p_r_four.png")));
        p_r_four.setHorizontalAlignment(SwingConstants.CENTER);
        p_r_four.setBounds(299, 252, 130, 128);
        p_r_four.setVisible(false);
        l_pnl.add(p_r_four);

        p_r_three = new JLabel("");
        p_r_three.setIcon(new
ImageIcon(getClass().getResource("p_r_three.png")));
        p_r_three.setHorizontalAlignment(SwingConstants.CENTER);
        p_r_three.setBounds(299, 252, 130, 128);
        p_r_three.setVisible(false);
        l_pnl.add(p_r_three);

        p_r_two = new JLabel("");
        p_r_two.setIcon(new
ImageIcon(getClass().getResource("p_r_two.png")));
        p_r_two.setHorizontalAlignment(SwingConstants.CENTER);
        p_r_two.setBounds(299, 252, 130, 128);
        p_r_two.setVisible(false);
        l_pnl.add(p_r_two);

        p_r_one = new JLabel("");
        p_r_one.setIcon(new
ImageIcon(getClass().getResource("p_r_one.png")));
        p_r_one.setHorizontalAlignment(SwingConstants.CENTER);
        p_r_one.setBounds(299, 252, 130, 128);
        l_pnl.add(p_r_one);

        JLabel player_lbl = new JLabel("Player");
        player_lbl.setForeground(new Color(255, 215, 0));
        player_lbl.setFont(new Font("Tahoma", Font.BOLD, 35));
        player_lbl.setHorizontalAlignment(SwingConstants.CENTER);
        player_lbl.setBounds(197, 0, 182, 50);
        pwc_frm.getContentPane().add(player_lbl);
```

```
        JLabel player_lbl_1 = new JLabel("Computer");

    player_lbl_1.setHorizontalAlignment(SwingConstants.CENTER);
        player_lbl_1.setForeground(new Color(255, 215, 0));
        player_lbl_1.setFont(new Font("Tahoma", Font.BOLD, 35));
        player_lbl_1.setBounds(848, 0, 182, 50);
        pwc_frm.getContentPane().add(player_lbl_1);

        JLabel vs_lbl = new JLabel("Vs");
        vs_lbl.setHorizontalAlignment(SwingConstants.CENTER);
        vs_lbl.setForeground(new Color(255, 215, 0));
        vs_lbl.setFont(new Font("Tahoma", Font.BOLD, 30));
        vs_lbl.setBounds(576, 261, 84, 40);
        pwc_frm.getContentPane().add(vs_lbl);

        main_pnl = new JPanel();
        main_pnl.setLayout(null);
        main_pnl.setBorder(new CompoundBorder(new
BevelBorder(BevelBorder.RAISED,
                                                    new
Color(0, 0, 128), new Color(0, 0, 205),
                                                    new
Color(0, 0, 139), new Color(0, 0, 255)),
                                                new
BevelBorder(BevelBorder.LOWERED,
                                                    new
Color(0, 0, 205), new Color(0, 0, 255),
                                                    new
Color(0, 0, 128), new Color(0, 0, 139))));
        main_pnl.setBackground(new Color(30, 144, 255));
        main_pnl.setBounds(50, 521, 1136, 183);
        pwc_frm.getContentPane().add(main_pnl);

        msg = new JTextArea("Game Started");
        msg.setWrapStyleWord(true);
        msg.setForeground(Color.YELLOW);
        msg.setFont(new Font("Tahoma", Font.BOLD | Font.ITALIC,
32));
        msg.setBounds(10, 10, 1116, 163);
        msg.setVisible(true);
        msg.setEditable(false);
        msg.setOpaque(false);
        main_pnl.add(msg);

        attack_btn = new JButton("Attack");
```

```
        attack_btn.setBackground(new Color(51, 102, 102));
        attack_btn.setFocusable(false);
        attack_btn.addActionListener(this);
        attack_btn.setForeground(new Color(255, 255, 51));
        attack_btn.setFont(new Font("Tahoma", Font.BOLD, 30));
        attack_btn.setBounds(675, 90, 157, 59);
        attack_btn.setVisible(false);
        attack_btn.addMouseListener(this);
        msg.add(attack_btn);

        split_btn = new JButton("Split");
        split_btn.setBackground(new Color(51, 102, 102));
        split_btn.setFocusable(false);
        split_btn.addActionListener(this);
        split_btn.setFont(new Font("Tahoma", Font.BOLD, 30));
        split_btn.setForeground(new Color(255, 255, 51));
        split_btn.setBounds(291, 90, 157, 59);
        split_btn.setVisible(false);
        split_btn.addMouseListener(this);
        msg.add(split_btn);

        done_btn = new JButton("Done");
        done_btn.setForeground(new Color(255, 255, 51));
        done_btn.setBackground(new Color(51, 102, 102));
        done_btn.addActionListener(this);
        done_btn.setFont(new Font("Tahoma", Font.BOLD, 30));
        done_btn.setBounds(493, 90, 139, 59);
        done_btn.setFocusable(false);
        done_btn.setVisible(false);
        done_btn.addMouseListener(this);
        msg.add(done_btn);

        w_l = new JLabel("");
        w_l.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
45));
        w_l.setOpaque(true);
        w_l.setHorizontalAlignment(SwingConstants.CENTER);
        w_l.setForeground(Color.YELLOW);
        w_l.setBackground(Color.BLACK);
        w_l.setBounds(382, 10, 371, 86);
        w_l.setVisible(false);
        main_pnl.add(w_l);

        JPanel player_pnl = new JPanel();
        player_pnl.setBounds(208, 0, 161, 50);
```

```
        pwc_frm.getContentPane().add(player_pnl);
        player_pnl.setLayout(null);
        player_pnl.setBorder(new CompoundBorder(new
BevelBorder(BevelBorder.RAISED,
                                                    new
Color(0, 0, 128), new Color(0, 0, 205),
                                                    new
Color(0, 0, 139), new Color(0, 0, 255)),
                                                new
BevelBorder(BevelBorder.LOWERED,
                                                    new
Color(0, 0, 205), new Color(0, 0, 255),
                                                    new
Color(0, 0, 128), new Color(0, 0, 139))));
        player_pnl.setBackground(new Color(30, 144, 255));

        JPanel com_pnl = new JPanel();
        com_pnl.setLayout(null);
        com_pnl.setBorder(new CompoundBorder(new
BevelBorder(BevelBorder.RAISED,
                                                new
Color(0, 0, 128), new Color(0, 0, 205),
                                                new
Color(0, 0, 139), new Color(0, 0, 255)),
                                            new
BevelBorder(BevelBorder.LOWERED,
                                                new
Color(0, 0, 205), new Color(0, 0, 255),
                                                new
Color(0, 0, 128), new Color(0, 0, 139))));
        com_pnl.setBackground(new Color(30, 144, 255));
        com_pnl.setBounds(834, 0, 208, 50);
        pwc_frm.getContentPane().add(com_pnl);

        JPanel vs_pnl = new JPanel();
        vs_pnl.setLayout(null);
        vs_pnl.setBorder(new CompoundBorder(new
BevelBorder(BevelBorder.RAISED,
                                                new
Color(0, 0, 128), new Color(0, 0, 205),
                                                new
Color(0, 0, 139), new Color(0, 0, 255)),
                                            new
BevelBorder(BevelBorder.LOWERED,
```

```
                                                                new
Color(0, 0, 205), new Color(0, 0, 255),
                                                                new
Color(0, 0, 128), new Color(0, 0, 139))));
          vs_pnl.setBackground(new Color(30, 144, 255));
          vs_pnl.setBounds(564, 261, 108, 40);
          pwc_frm.getContentPane().add(vs_pnl);

          if(!br.ready()) {
               pwc_frm.getContentPane().setBackground(new Color(102,
0, 0));
               l_pnl.setBackground(new Color(30, 144, 255));
               r_pnl.setBackground(new Color(30, 144, 255));
               br.close();
          }
          else {
               pwc_frm.getContentPane().setBackground(new
Color(br.read(), br.read(), br.read()));
               br.skip(3);
               l_pnl.setBackground(new Color(br.read(), br.read(),
br.read()));
               r_pnl.setBackground(new Color(br.read(), br.read(),
br.read()));
               br.close();
          }
          pwc_frm.setVisible(true);
     }

     @Override
     public void mouseClicked(MouseEvent e) {


     }

     @Override
     public void mousePressed(MouseEvent e) {


     }

     @Override
     public void mouseReleased(MouseEvent e) {


     }
```

```
@Override
public void mouseEntered(MouseEvent e) {

     if(e.getSource()==quit_btn) {
          quit_btn.setBackground(Color.black);
     }
     if(e.getSource()==split_btn) {
          split_btn.setBackground(Color.black);
     }
     if(e.getSource()==attack_btn) {
          attack_btn.setBackground(Color.black);
     }
     if(e.getSource()==done_btn) {
          done_btn.setBackground(Color.black);
     }
     if(e.getSource()==p_l_btn) {
          p_l_btn.setBackground(Color.black);
     }
     if(e.getSource()==p_r_btn) {
          p_r_btn.setBackground(Color.black);
     }
     if(e.getSource()==cp_l_btn) {
          cp_l_btn.setBackground(Color.black);
     }
     if(e.getSource()==cp_r_btn) {
          cp_r_btn.setBackground(Color.black);
     }

}

@Override
public void mouseExited(MouseEvent e) {

     if(e.getSource()==quit_btn) {
          quit_btn.setBackground(new Color(255, 204, 0));
     }
     if(e.getSource()==split_btn) {
          split_btn.setBackground(new Color(51, 102, 102));
     }
     if(e.getSource()==attack_btn) {
          attack_btn.setBackground(new Color(51, 102, 102));
     }
     if(e.getSource()==done_btn) {
          done_btn.setBackground(new Color(51, 102, 102));
```

```
            }
            if(e.getSource()==p_l_btn) {
                 p_l_btn.setBackground(new Color(51, 51, 153));
            }
            if(e.getSource()==p_r_btn) {
                 p_r_btn.setBackground(new Color(0, 0, 204));
            }
            if(e.getSource()==cp_l_btn) {
                 cp_l_btn.setBackground(new Color(51, 51, 153));
            }
            if(e.getSource()==cp_r_btn) {
                 cp_r_btn.setBackground(new Color(0, 0, 204));
            }

       }


       public void actionPerformed(ActionEvent e) {

                 if(e.getSource()==quit_btn) {
                      int Q=JOptionPane.showConfirmDialog(pwc_frm,
"Sure want to quit?", "Quit",

       JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
                      if(Q==0) {
                           pwc_frm.dispose();
                           try {
                                new
ChopStick().main_frm.setVisible(true);
                           } catch (Exception e1) {
                                e1.printStackTrace();
                           }
                      }
                 }

                 if(e.getSource()==split_btn) {
                      split=true;
                      p_r_btn.setEnabled(true);
                      p_l_btn.setEnabled(true);
                      cp_l_btn.setEnabled(false);
                      cp_r_btn.setEnabled(false);
                      attack_btn.setVisible(false);
                      split_btn.setVisible(false);
                      msg.setText("Click on your either of the hand-
buttons[right or left]\n"
```

```
                              + "For splitting the fingers by one,\n"
                              + "(One finger will be increased on the
clicked hand-button,\n"
                              + "hence it will decrease one finger on
the opposite)");
                    prpr=p_l_h.size();
                    prpl=p_r_h.size();
                    p=prpr+prpl;
               }

               if(e.getSource()==attack_btn) {
                    attack=true;
                    if(p_l_h.size()==0) {
                         p_l_btn.setEnabled(false);
                         p_r_btn.setEnabled(true);
                    }
                    else if(p_r_h.size()==0) {
                         p_r_btn.setEnabled(false);
                         p_l_btn.setEnabled(true);
                    }
                    else {
                         p_r_btn.setEnabled(true);
                         p_l_btn.setEnabled(true);
                    }
                    attack_btn.setVisible(false);
                    split_btn.setVisible(false);
                    msg.setText("First click on either of your hand-
buttons[right or left]\n"
                                   + "and then computer's for the
attack");
               }


               if(e.getSource()==p_l_btn&&split) {
                    msg.setText(".....Splitting.....");

                    if(p_l_h.size()>=4)
                         JOptionPane.showMessageDialog(l_pnl,
                                   "You can't kill your own
hand!,\nIn splitting decrement of fingers isn't allowed",
                                   "Fingers' Decrement",
JOptionPane.WARNING_MESSAGE);
                    else if(p_r_h.size()==0) {
                         JOptionPane.showMessageDialog(l_pnl,
```

```
                                                    "No fingers
left to transfer\nYou got a fist on right hand",
                                                    "No fingers",
JOptionPane.WARNING_MESSAGE);
                    }
                    else {
                         p_l_h.add(1);
                         p_r_h.remove(p_r_h.lastElement());
                         if(p_l_h.size()==0) {
                         p_l_0();
                        }
                        else if(p_l_h.size()==1) {
                         p_l_1();
                        }
                        else if(p_l_h.size()==2) {
                         p_l_2();
                        }
                        else if(p_l_h.size()==3) {
                         p_l_3();
                        }
                        else if(p_l_h.size()==4) {
                         p_l_4();
                        }

                         if(p_r_h.size()==0) {
                              p_r_0();
                        }
                        else if(p_r_h.size()==1) {
                         p_r_1();
                        }
                        else if(p_r_h.size()==2) {
                         p_r_2();
                        }
                        else if(p_r_h.size()==3) {
                         p_r_3();
                        }
                        else if(p_r_h.size()==4) {
                         p_r_4();
                        }
                    }
                    done_btn.setVisible(true);
                }
                if(e.getSource()==p_r_btn&&split) {
                    msg.setText(".....Splitting.....");
```

```
                    if(p_r_h.size()>=4)
                         JOptionPane.showMessageDialog(l_pnl,
                                   "You can't kill your own
hand!,\nIn splitting decrement of fingers isn't allowed",
                                   "Fingers' Decrement",
JOptionPane.WARNING_MESSAGE);
                    else if(p_l_h.size()==0) {
                         JOptionPane.showMessageDialog(l_pnl,
                                   "you already got a fist on left
hand!", "Fist on left",
                                   JOptionPane.WARNING_MESSAGE);
                    }
                    else {
                         p_r_h.add(1);
                         p_l_h.remove(p_l_h.lastElement());

                         if(p_r_h.size()==0) {
                              p_r_0();
                        }
                        else if(p_r_h.size()==1) {
                         p_r_1();
                        }
                        else if(p_r_h.size()==2) {
                         p_r_2();
                        }
                        else if(p_r_h.size()==3) {
                         p_r_3();
                        }
                        else if(p_r_h.size()==4) {
                         p_r_4();
                        }

                         if(p_l_h.size()==0) {
                         p_l_0();
                        }
                        else if(p_l_h.size()==1) {
                         p_l_1();
                        }
                        else if(p_l_h.size()==2) {
                         p_l_2();
                        }
                        else if(p_l_h.size()==3) {
                         p_l_3();
                        }
                        else if(p_l_h.size()==4) {
```

```
                    p_l_4();
                   }


               }
               done_btn.setVisible(true);
          }

          if(e.getSource()==done_btn) {
               if(p_l_h.size()==prpr&&p_r_h.size()==prpl ||
p_l_h.size()==prpl&&p_r_h.size()==prpr) {
                    JOptionPane.showMessageDialog(l_pnl,
"Splitting is same as previos State!",

"Identical states", JOptionPane.WARNING_MESSAGE);
               }
               else {
                    p_r_btn.setEnabled(false);
                    p_l_btn.setEnabled(false);
                    cp_r_btn.setEnabled(false);
                    cp_l_btn.setEnabled(false);
                    done_btn.setVisible(false);
                    split=false;
                    t=false;
               }

          }


          if(e.getSource()==p_l_btn&&attack) {
               if(cp_r_h.size()==0) {
                    cp_r_btn.setEnabled(false);
                    cp_l_btn.setEnabled(true);
               }
               else if(cp_l_h.size()==0) {
                    cp_l_btn.setEnabled(false);
                    cp_r_btn.setEnabled(true);
               }
               else {
                    cp_r_btn.setEnabled(true);
                    cp_l_btn.setEnabled(true);
               }
               msg.setText(".....Your left hand.....");
               p_l=true;
               p_r=false;
```

```
                }
                if(e.getSource()==p_r_btn&&attack) {
                     if(cp_r_h.size()==0) {
                          cp_r_btn.setEnabled(false);
                          cp_l_btn.setEnabled(true);
                     }
                     else if(cp_l_h.size()==0) {
                          cp_l_btn.setEnabled(false);
                          cp_r_btn.setEnabled(true);
                     }
                     else {
                          cp_r_btn.setEnabled(true);
                          cp_l_btn.setEnabled(true);
                     }
                     msg.setText(".....Your right hand.....");
                     p_r=true;
                     p_l=false;
                }

                if(e.getSource()==cp_l_btn&&p_l) {
                     msg.setText(".....Your left hand -> Computer's
left hand.....");

                     p_l_btn.setEnabled(false);
                     p_r_btn.setEnabled(false);
                     cp_l_btn.setEnabled(false);
                     cp_r_btn.setEnabled(false);

                     x=p_l_h.size();
                    for(i=0;i<x;i++) {
                     if(cp_l_h.size()==5)
                          cp_l_h.removeAllElements();
                     cp_l_h.add(1);
                    }
                    if(cp_l_h.size()==5) {
                     cp_l_h.removeAllElements();
                    }

                     t=false;
                }
                if(e.getSource()==cp_r_btn&&p_l) {
                     msg.setText(".....Your left hand -> Computer's
right hand.....");

                     p_l_btn.setEnabled(false);
```

```
                    p_r_btn.setEnabled(false);
                    cp_l_btn.setEnabled(false);
                    cp_r_btn.setEnabled(false);

                    x=p_l_h.size();
                   for(i=0;i<x;i++) {
                    if(cp_r_h.size()==5)
                         cp_r_h.removeAllElements();
                    cp_r_h.add(1);
                   }
                   if(cp_r_h.size()==5) {
                    cp_r_h.removeAllElements();
                   }

                    t=false;
               }
               if(e.getSource()==cp_l_btn&&p_r) {
                    msg.setText(".....Your right hand -> Computer's
left hand.....");

                    p_l_btn.setEnabled(false);
                    p_r_btn.setEnabled(false);
                    cp_l_btn.setEnabled(false);
                    cp_r_btn.setEnabled(false);

                    x=p_r_h.size();
                   for(i=0;i<x;i++) {
                    if(cp_l_h.size()==5)
                         cp_l_h.removeAllElements();
                    cp_l_h.add(1);
                   }
                   if(cp_l_h.size()==5) {
                    cp_l_h.removeAllElements();
                   }

                    t=false;
               }
               if(e.getSource()==cp_r_btn&&p_r) {
                    msg.setText(".....Your right hand -> Computer's
right hand.....");

                    p_l_btn.setEnabled(false);
                    p_r_btn.setEnabled(false);
                    cp_l_btn.setEnabled(false);
                    cp_r_btn.setEnabled(false);
```

```
                    x=p_r_h.size();
                   for(i=0;i<x;i++) {
                    if(cp_r_h.size()==5)
                         cp_r_h.removeAllElements();
                    cp_r_h.add(1);
                   }
                   if(cp_r_h.size()==5) {
                    cp_r_h.removeAllElements();
                   }

                    t=false;
               }

     }

     public void loading() {
          String s=".";
          for(int i=0;i<5;i++) {
               msg.setText(s);
               s+=".";
               delay(150);
          }
     }

     public synchronized void Gameloop() throws
InterruptedException {
          p_l_h.add(1);
          p_r_h.add(1);
          cp_l_h.add(1);
          cp_r_h.add(1);
               delay(1000);
               loading();
               msg.setText("...Randomly Choosing the turn!...");
               delay(2000);
               loading();
               if(r.nextInt(2)==1) {
                    msg.setText("Computer's turn");
                    delay(1000);
                    turn=true;
               }
               else {
                    msg.setText("Player's turn");
                    delay(1000);
                    turn=false;
```

```
                }

                while(running) {
                     if(cp_l_h.isEmpty() && cp_r_h.isEmpty()) {
                          pla_won();
                     }
                     else if(p_l_h.isEmpty() && p_r_h.isEmpty())    {
                          com_won();
                     }
                     else {
                          if(turn) {
                               Com_turn();
                               if(cp_l_h.isEmpty() &&
cp_r_h.isEmpty()) {
                                    pla_won();
                               }
                               else if(p_l_h.isEmpty() &&
p_r_h.isEmpty())   {
                                    com_won();
                               }
                               else {
                                    turn=false;
                               }
                               repaint();
                          }
                          else {
                               Pla_turn();
                               while(t) {
                                    delay(1000);
                               }
                               Com_upd();
                               if(cp_l_h.isEmpty() &&
cp_r_h.isEmpty()) {
                                    pla_won();
                               }
                               else if(p_l_h.isEmpty() &&
p_r_h.isEmpty())   {
                                    com_won();
                               }
                               else {
                                    turn=true;
                                    t=true;
                               }
                               repaint();
```

```
                    }
                }
            }

        }

    public void Com_upd() {

        loading();

        if(cp_l_h.size()==0) {
        cp_l_5_0();
       }
       else if(cp_l_h.size()==1) {
        cp_l_1();
       }
       else if(cp_l_h.size()==2) {
        cp_l_2();
       }
       else if(cp_l_h.size()==3) {
        cp_l_3();
       }
       else if(cp_l_h.size()==4) {
        cp_l_4();
       }


        if(cp_r_h.size()==0) {
            cp_r_5_0();
       }
       else if(cp_r_h.size()==1) {
        cp_r_1();
       }
       else if(cp_r_h.size()==2) {
        cp_r_2();
       }
       else if(cp_r_h.size()==3) {
        cp_r_3();
       }
       else if(cp_r_h.size()==4) {
        cp_r_4();
       }
    }

    public void com_won() {
```

```
            loading();

            cp_l_btn.setVisible(false);
            cp_r_btn.setVisible(false);
            p_l_btn.setVisible(false);
            p_r_btn.setVisible(false);
            msg.setVisible(false);
            w_l.setVisible(true);
            w_l.setText("(: You Lost :)");
            delay(1000);

            int O=JOptionPane.showConfirmDialog(pwc_frm, "want to
play again ?", "play with computer",

JOptionPane.YES_NO_OPTION ,JOptionPane.QUESTION_MESSAGE , null);
            if(JOptionPane.YES_OPTION==O) {
                 running=false;
                 pwc_frm.dispose();
                 try {
                      new PWC();
                 } catch (Exception e) {
                      e.printStackTrace();
                 }
            }
            else if(JOptionPane.NO_OPTION==O) {
                 running=false;
                 pwc_frm.dispose();
                 try {
                      new ChopStick().main_frm.setVisible(true);
                 } catch (Exception e) {
                      e.printStackTrace();
                 }
            }
            else {
                 running=false;
            }

            if(Void_matrix_account.pag) {
                 lcount=0;
                 lcount++;
                 try {
                      bw=new BufferedWriter(new
FileWriter("lost.txt"));
                      bw.write(lcount);
                      bw.close();
```

```
                    c.close();
               } catch (Exception e) {
                    e.printStackTrace();
               }
          }
          else {
               String
url="jdbc:mysql://db4free.net:3306/stickchop?useSSL=false";
               String usr="voidmatrix";
               String s="voidmatrix2002";
               try {
                    Class.forName("com.mysql.cj.jdbc.Driver");
                    c= DriverManager.getConnection(url,usr,s);
                    st=c.createStatement();
               } catch (Exception e) {
                    e.printStackTrace();
               }
               int l=0;
               String name=null;
               try {
                    br=new BufferedReader(new
FileReader("player.txt"));
                    name=br.readLine();
                    br.close();
                    ResultSet rs=st.executeQuery("select *from
signup where uname='"+name+"'");
                    rs.next();
                    l=rs.getInt(6);
               } catch (SQLException e1) {
                    e1.printStackTrace();
               } catch (Exception e1) {
                    e1.printStackTrace();
               }
               lcount=l;
               lcount++;
               try {
                    PreparedStatement p=c.prepareStatement("update
signup set="+lcount+" where uname='"+name+"'");
                    bw=new BufferedWriter(new
FileWriter("lost.txt"));
                    bw.write(lcount);
                    bw.close();
                    c.close();
               } catch (Exception e) {
                    e.printStackTrace();
```

```
                }
            }

        }

        public void pla_won()  {
            loading();

            cp_l_btn.setVisible(false);
            cp_r_btn.setVisible(false);
            p_l_btn.setVisible(false);
            p_r_btn.setVisible(false);
            msg.setVisible(false);
            w_l.setVisible(true);
            w_l.setText("(: You Won :)");
            delay(1000);
            int O=JOptionPane.showConfirmDialog(pwc_frm, "want to
play again ?", "play with computer",

JOptionPane.YES_NO_OPTION ,JOptionPane.QUESTION_MESSAGE , null);
            if(JOptionPane.YES_OPTION==0) {
                running=false;
                pwc_frm.dispose();
                try {
                    new PWC();
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
            else if(JOptionPane.NO_OPTION==0) {
                running=false;
                pwc_frm.dispose();
                try {
                    new ChopStick().main_frm.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
            else {
                running=false;
            }
            if(Void_matrix_account.pag) {
                wcount=0;
                wcount++;
                try {
```

```
                    bw=new BufferedWriter(new
FileWriter("won.txt"));
                    bw.write(wcount);
                    bw.close();
                    c.close();
               } catch (Exception e) {
                    e.printStackTrace();
               }
          }
          else {
               String
url="jdbc:mysql://db4free.net:3306/matrixvoid02?useSSL=false";
               String usr="chopstick02";
               String s="15112002Chop";
               try {
                    Class.forName("com.mysql.cj.jdbc.Driver");
                    c= DriverManager.getConnection(url,usr,s);
                    st=c.createStatement();
               } catch (Exception e) {
                    e.printStackTrace();
               }
               int w=0;
               String name=null;
               try {
                    br=new BufferedReader(new
FileReader("player.txt"));
                    name=br.readLine();
                    br.close();
                    ResultSet rs=st.executeQuery("select *from
ChopStick_Players where pname='"+name+"'");
                    rs.next();
                    w=rs.getInt(5);
               } catch (SQLException e1) {
                    e1.printStackTrace();
               } catch (Exception e1) {
                    e1.printStackTrace();
               }
               wcount=w;
               wcount++;
               try {
                    PreparedStatement p=c.prepareStatement("update
ChopStick_Players set="+wcount+" where pname='"+name+"'");
                    bw=new BufferedWriter(new
FileWriter("won.txt"));
                    bw.write(wcount);
```

```
                    bw.close();
                    c.close();
               } catch (Exception e) {
                    e.printStackTrace();
               }
          }

     }

     public void Com_turn() {
          loading();

          r_pnl.setBackground(Color.white);
          delay(500);
          r_pnl.setBackground(new Color(30, 144, 255));

          msg.setText("Computer is making the move");
          delay(1000);

          if(cp_l_h.size()==1&&cp_r_h.size()==0)
          {
               loading();
               msg.setText("Computer chose to attack");
               delay(1000);
               if(p_l_h.isEmpty()) {
                    loading();
                    msg.setText("Computer's left hand -> Your right
hand");
                    delay(3000);
                    C_L_R_HIT();
               }
               else if(p_r_h.isEmpty()) {
                    loading();
                    msg.setText("Computer's left hand -> Your left
hand");
                    delay(3000);
                    C_L_L_HIT();
               }
               else {
                    cp_ch_p_h=r.nextInt(2);//p : 0=l , 1=r
                    if(cp_ch_p_h==0)
                    {
                         loading();
                         msg.setText("Computer's left hand -> Your
left hand");
```

```
                    delay(3000);
                    C_L_L_HIT();
               }
               else
               {
                    loading();
                    msg.setText("Computer's left hand -> Your
right hand");
                    delay(3000);
                    C_L_R_HIT();
               }
          }
     }

     else if(cp_l_h.size()==0&&cp_r_h.size()==1)
     {
          loading();
          msg.setText("Computer chose to attack");
          delay(1000);
          if(p_l_h.isEmpty()) {
               loading();
               msg.setText("Computer's right hand -> Your right
hand");
               delay(3000);
               C_R_R_HIT();
          }
          else if(p_r_h.isEmpty()) {
               loading();
               msg.setText("Computer's right hand -> Your left
hand");
               delay(3000);
               C_R_L_HIT();
          }
          else {
               cp_ch_p_h=r.nextInt(2);//p : 0=l , 1=r
               if(cp_ch_p_h==0)
               {
                    loading();
                    msg.setText("Computer's right hand -> Your
left hand");
                    delay(3000);
                    C_R_L_HIT();
               }
               else
               {
```

```
                    loading();
                    msg.setText("Computer's right hand -> Your
right hand");
                    delay(3000);
                    C_R_R_HIT();
                }
            }
        }

        else if(cp_l_h.size()+cp_r_h.size()>4) {
            cp_ch_c_h=r.nextInt(2);//c : 0=l , 1=r
            cp_ch_p_h=r.nextInt(2);//p : 0=l , 1=r
            loading();
            msg.setText("Computer chose to attack");
            delay(1000);

            if(p_l_h.isEmpty()) {
                if(cp_ch_c_h==0)
                {
                    loading();
                    msg.setText("Computer's left hand -> Your
right hand");
                    delay(3000);
                    C_L_R_HIT();
                }
                else {
                    loading();
                    msg.setText("Computer's right hand -> Your
right hand");
                    delay(3000);
                    C_R_R_HIT();
                }
            }
            else if(p_r_h.isEmpty()) {
                if(cp_ch_c_h==0)
                {
                    loading();
                    msg.setText("Computer's left hand -> Your
left hand");
                    delay(3000);
                    C_L_L_HIT();
                }
                else {
                    loading();
```

```
                    msg.setText("Computer's right hand -> Your
left hand");
                    delay(3000);
                    C_R_L_HIT();
                }
            }
            else {
                if(cp_ch_c_h==0 && cp_ch_p_h==0)
                {
                    loading();
                    msg.setText("Computer's left hand -> Your
left hand");
                    delay(3000);
                    C_L_L_HIT();
                }
                else if(cp_ch_c_h==0 && cp_ch_p_h==1)
                {
                    loading();
                    msg.setText("Computer's left hand -> Your
right hand");
                    delay(3000);
                    C_L_R_HIT();
                }
                else if(cp_ch_c_h==1 && cp_ch_p_h==0)
                {
                    loading();
                    msg.setText("Computer's right hand -> Your
left hand");
                    delay(3000);
                    C_R_L_HIT();
                }
                else if(cp_ch_c_h==1 && cp_ch_p_h==1)
                {
                    loading();
                    msg.setText("Computer's right hand -> Your
right hand");
                    delay(3000);
                    C_R_R_HIT();
                }

            }
        }
        else {

            cp_ch=r.nextInt(4);//0=a , 1=s , 2=a, 3=s
```

```
                if(cp_ch==0 || cp_ch==2)
                {
                     cp_ch_c_h=r.nextInt(2);//c : 0=l , 1=r
                     cp_ch_p_h=r.nextInt(2);//p : 0=l , 1=r
                     loading();
                     msg.setText("Computer chose to attack");
                     delay(1000);

                     if(cp_l_h.isEmpty()) {
                          if(p_l_h.isEmpty()) {
                               loading();
                               msg.setText("Computer's right hand ->
Your right hand");
                               delay(3000);
                               C_R_R_HIT();
                          }
                          else if(p_r_h.isEmpty()) {
                               loading();
                               msg.setText("Computer's right hand ->
Your left hand");
                               delay(3000);
                               C_R_L_HIT();
                          }
                          else {
                               cp_ch_p_h=r.nextInt(2);//p : 0=l , 1=r
                               if(cp_ch_p_h==0)
                               {
                                    loading();
                                    msg.setText("Computer's right hand
-> Your left hand");
                                    delay(3000);
                                    C_R_L_HIT();
                               }
                               else
                               {
                                    loading();
                                    msg.setText("Computer's right hand
-> Your right hand");
                                    delay(3000);
                                    C_R_R_HIT();
                               }
                          }
                     }
                     else if(cp_r_h.isEmpty()) {
```

```
                    if(p_l_h.isEmpty()) {
                         loading();
                         msg.setText("Computer's left hand ->
Your right hand");
                         delay(3000);
                         C_L_R_HIT();
                    }
                    else if(p_r_h.isEmpty()) {
                         loading();
                         msg.setText("Computer's left hand ->
Your left hand");
                         delay(3000);
                         C_L_L_HIT();
                    }
                    else {
                         cp_ch_p_h=r.nextInt(2);//p : 0=l , 1=r
                         if(cp_ch_p_h==0)
                         {
                              loading();
                              msg.setText("Computer's left hand
-> Your left hand");
                              delay(3000);
                              C_L_L_HIT();
                         }
                         else
                         {
                              loading();
                              msg.setText("Computer's left hand
-> Your right hand");
                              delay(3000);
                              C_L_R_HIT();
                         }
                    }
               }
               else if(p_l_h.isEmpty()) {
                    if(cp_ch_c_h==0)
                    {
                         loading();
                         msg.setText("Computer's left hand ->
Your right hand");
                         delay(3000);
                         C_L_R_HIT();
                    }
                    else {
                         loading();
```

```
                    msg.setText("Computer's right hand ->
Your right hand");
                    delay(3000);
                    C_R_R_HIT();
               }
          }
          else if(p_r_h.isEmpty()) {
               if(cp_ch_c_h==0)
               {
                    loading();
                    msg.setText("Computer's left hand ->
Your left hand");
                    delay(3000);
                    C_L_L_HIT();
               }
               else {
                    loading();
                    msg.setText("Computer's right hand ->
Your left hand");
                    delay(3000);
                    C_R_L_HIT();
               }
          }
          else {
               if(cp_ch_c_h==0 && cp_ch_p_h==0)
               {
                    loading();
                    msg.setText("Computer's left hand ->
Your left hand");
                    delay(3000);
                    C_L_L_HIT();
               }
               else if(cp_ch_c_h==0 && cp_ch_p_h==1)
               {
                    loading();
                    msg.setText("Computer's left hand ->
Your right hand");
                    delay(3000);
                    C_L_R_HIT();
               }
               else if(cp_ch_c_h==1 && cp_ch_p_h==0)
               {
                    loading();
                    msg.setText("Computer's right hand ->
Your left hand");
```

```
                    delay(3000);
                    C_R_L_HIT();
               }
               else if(cp_ch_c_h==1 && cp_ch_p_h==1)
               {
                    loading();
                    msg.setText("Computer's right hand ->
Your right hand");
                    delay(3000);
                    C_R_R_HIT();
               }
          }

     }

     else
     {
          loading();
          msg.setText("Computer chose to split");
          delay(1000);
          do {
               prcpl=cp_l_h.size();
               prcpr=cp_r_h.size();
               cp1=cp_l_h.size()+cp_r_h.size();
               cph=r.nextInt(2);//0=L,1=R
               rc=r.nextInt(cp1+1);
               cp2=cp1-rc;
               if(cph==0)
               {
                    cp_l_h.removeAllElements();
                    cp_r_h.removeAllElements();
                    for(i=0;i<rc;i++)
                    {
                         cp_l_h.add(1);
                    }
                    for(i=0;i<cp2;i++)
                    {
                         cp_r_h.add(1);
                    }
                    loading();
               }
               else if(cph==1)
               {
                    cp_l_h.removeAllElements();
                    cp_r_h.removeAllElements();
```

```
                    for(i=0;i<rc;i++)
                    {
                         cp_r_h.add(1);
                    }
                    for(i=0;i<cp2;i++)
                    {
                         cp_l_h.add(1);
                    }
                    loading();
               }
          }
          while(cp_l_h.size()==cp_r_h.size() ||
cp_l_h.size()==prcpr&&cp_r_h.size()==prcpl || cp_l_h.size()==prcpl
|| cp_r_h.size()==prcpr);

          if(cp_l_h.size()==0) {
          cp_l_0();
         }
         else if(cp_l_h.size()==1) {
          cp_l_1();
         }
         else if(cp_l_h.size()==2) {
          cp_l_2();
         }
         else if(cp_l_h.size()==3) {
          cp_l_3();
         }
         else if(cp_l_h.size()==4) {
          cp_l_4();
         }


          if(cp_r_h.size()==0) {
               cp_r_0();
         }
         else if(cp_r_h.size()==1) {
          cp_r_1();
         }
         else if(cp_r_h.size()==2) {
          cp_r_2();
         }
         else if(cp_r_h.size()==3) {
          cp_r_3();
         }
         else if(cp_l_h.size()==4) {
```

```
                cp_r_4();
               }

           }
      }
 }


  //computer-hit
      public void C_L_L_HIT()
      {
           x=cp_l_h.size();
          for(i=0;i<x;i++) {
           if(p_l_h.size()==5) {
                p_l_h.removeAllElements();
                p_l_5_0();
           }
           p_l_h.add(1);
          }
          if(p_l_h.size()==5) {
           p_l_h.removeAllElements();
          }

          if(p_l_h.size()==0) {
           p_l_5_0();
          }
          else if(p_l_h.size()==1) {
           p_l_1();
          }
          else if(p_l_h.size()==2) {
           p_l_2();
          }
          else if(p_l_h.size()==3) {
           p_l_3();
          }
          else if(p_l_h.size()==4) {
           p_l_4();
          }
      }

      public void C_L_R_HIT()
      {
           x=cp_l_h.size();
          for(i=0;i<x;i++) {
           if(p_r_h.size()==5) {
```

```
                p_r_h.removeAllElements();
                p_r_5_0();
            }
            p_r_h.add(1);
           }
           if(p_r_h.size()==5) {
            p_r_h.removeAllElements();
           }

           if(p_r_h.size()==0) {
            p_r_5_0();
           }
           else if(p_r_h.size()==1) {
            p_r_1();
           }
           else if(p_r_h.size()==2) {
            p_r_2();
           }
           else if(p_r_h.size()==3) {
            p_r_3();
           }
           else if(p_r_h.size()==4) {
            p_r_4();
           }
       }

       public void C_R_L_HIT()
       {
            x=cp_r_h.size();
           for(i=0;i<x;i++) {
            if(p_l_h.size()==5) {
                p_l_h.removeAllElements();
                p_l_5_0();
            }
            p_l_h.add(1);
           }
           if(p_l_h.size()==5) {
            p_l_h.removeAllElements();
           }

           if(p_l_h.size()==0) {
            p_l_5_0();
           }
           else if(p_l_h.size()==1) {
            p_l_1();
```

```
        }
        else if(p_l_h.size()==2) {
         p_l_2();
        }
        else if(p_l_h.size()==3) {
         p_l_3();
        }
        else if(p_l_h.size()==4) {
         p_l_4();
        }
    }

    public void C_R_R_HIT()
    {
         x=cp_r_h.size();

        for(i=0;i<x;i++) {
         if(p_r_h.size()==5) {
              p_r_h.removeAllElements();
              p_r_5_0();
         }
         p_r_h.add(1);
        }
        if(p_r_h.size()==5) {
         p_r_h.removeAllElements();
        }

        if(p_r_h.size()==0) {
         p_r_5_0();
        }
        else if(p_r_h.size()==1) {
         p_r_1();
        }
        else if(p_r_h.size()==2) {
         p_r_2();
        }
        else if(p_r_h.size()==3) {
         p_r_3();
        }
        else if(p_r_h.size()==4) {
         p_r_4();

        }
    }
```

```
public void Pla_turn() {

     loading();
     l_pnl.setBackground(Color.white);
     delay(500);
     l_pnl.setBackground(new Color(30, 144, 255));

     if(p_l_h.size()==0&&p_r_h.size()==1) {
          msg.setText(".....You can only attack.....");
          attack_btn.setVisible(true);
          split_btn.setVisible(true);
          split_btn.setEnabled(false);
     }
     else if(p_r_h.size()==0&&p_l_h.size()==1) {
          msg.setText(".....You can only attack.....");
          attack_btn.setVisible(true);
          split_btn.setVisible(true);
          split_btn.setEnabled(false);
     }
     else {
          msg.setText("What do you wanna do ?");
          attack_btn.setVisible(true);
          split_btn.setVisible(true);
          split_btn.setEnabled(true);
     }

}

public void P_L_L_H() {

}

public void P_L_R_H() {

}

public void P_R_L_H() {

}

public void P_R_R_H() {

}
```

```
//cp_l_fingers
public void cp_l_0() {
     cp_l_one.setVisible(false);
     cp_l_two.setVisible(false);
     cp_l_three.setVisible(false);
     cp_l_four.setVisible(false);
     cp_l_five.setVisible(false);
     cp_l_fist.setVisible(true);
}

public void cp_l_1() {
     cp_l_one.setVisible(true);
     cp_l_two.setVisible(false);
     cp_l_three.setVisible(false);
     cp_l_four.setVisible(false);
     cp_l_five.setVisible(false);
     cp_l_fist.setVisible(false);
}

public void cp_l_2() {
     cp_l_one.setVisible(false);
     cp_l_two.setVisible(true);
     cp_l_three.setVisible(false);
     cp_l_four.setVisible(false);
     cp_l_five.setVisible(false);
     cp_l_fist.setVisible(false);
}

public void cp_l_3() {
     cp_l_one.setVisible(false);
     cp_l_two.setVisible(false);
     cp_l_three.setVisible(true);
     cp_l_four.setVisible(false);
     cp_l_five.setVisible(false);
     cp_l_fist.setVisible(false);
}

public void cp_l_4() {
     cp_l_one.setVisible(false);
     cp_l_two.setVisible(false);
     cp_l_three.setVisible(false);
     cp_l_four.setVisible(true);
     cp_l_five.setVisible(false);
     cp_l_fist.setVisible(false);
```

```
}

public void cp_l_5_0() {
     cp_l_one.setVisible(false);
     cp_l_two.setVisible(false);
     cp_l_three.setVisible(false);
     cp_l_four.setVisible(false);
     cp_l_five.setVisible(true);
     delay(1000);
     cp_l_five.setVisible(false);
     cp_l_fist.setVisible(true);
}


//cp_r_fingers
public void cp_r_0() {
     cp_r_one.setVisible(false);
     cp_r_two.setVisible(false);
     cp_r_three.setVisible(false);
     cp_r_four.setVisible(false);
     cp_r_five.setVisible(false);
     cp_r_fist.setVisible(true);
}

public void cp_r_1() {
     cp_r_one.setVisible(true);
     cp_r_two.setVisible(false);
     cp_r_three.setVisible(false);
     cp_r_four.setVisible(false);
     cp_r_five.setVisible(false);
     cp_r_fist.setVisible(false);
}

public void cp_r_2() {
     cp_r_one.setVisible(false);
     cp_r_two.setVisible(true);
     cp_r_three.setVisible(false);
     cp_r_four.setVisible(false);
     cp_r_five.setVisible(false);
     cp_r_fist.setVisible(false);
}

public void cp_r_3() {
     cp_r_one.setVisible(false);
     cp_r_two.setVisible(false);
```

```
        cp_r_three.setVisible(true);
        cp_r_four.setVisible(false);
        cp_r_five.setVisible(false);
        cp_r_fist.setVisible(false);
    }

    public void cp_r_4() {
        cp_r_one.setVisible(false);
        cp_r_two.setVisible(false);
        cp_r_three.setVisible(false);
        cp_r_four.setVisible(true);
        cp_r_five.setVisible(false);
        cp_r_fist.setVisible(false);
    }

    public void cp_r_5_0() {
        cp_r_one.setVisible(false);
        cp_r_two.setVisible(false);
        cp_r_three.setVisible(false);
        cp_r_four.setVisible(false);
        cp_r_five.setVisible(true);
        delay(1000);
        cp_r_five.setVisible(false);
        cp_r_fist.setVisible(true);
    }


    //p_l_fingers
    public void p_l_0() {
        p_l_one.setVisible(false);
        p_l_two.setVisible(false);
        p_l_three.setVisible(false);
        p_l_four.setVisible(false);
        p_l_five.setVisible(false);
        p_l_fist.setVisible(true);
    }

    public void p_l_1() {
        p_l_one.setVisible(true);
        p_l_two.setVisible(false);
        p_l_three.setVisible(false);
        p_l_four.setVisible(false);
        p_l_five.setVisible(false);
        p_l_fist.setVisible(false);
    }
```

```
public void p_l_2() {
     p_l_one.setVisible(false);
     p_l_two.setVisible(true);
     p_l_three.setVisible(false);
     p_l_four.setVisible(false);
     p_l_five.setVisible(false);
     p_l_fist.setVisible(false);
}

public void p_l_3() {
     p_l_one.setVisible(false);
     p_l_two.setVisible(false);
     p_l_three.setVisible(true);
     p_l_four.setVisible(false);
     p_l_five.setVisible(false);
     p_l_fist.setVisible(false);
}

public void p_l_4() {
     p_l_one.setVisible(false);
     p_l_two.setVisible(false);
     p_l_three.setVisible(false);
     p_l_four.setVisible(true);
     p_l_five.setVisible(false);
     p_l_fist.setVisible(false);
}

public void p_l_5_0() {
     p_l_one.setVisible(false);
     p_l_two.setVisible(false);
     p_l_three.setVisible(false);
     p_l_four.setVisible(false);
     p_l_five.setVisible(true);
     delay(1000);
     p_l_five.setVisible(false);
     p_l_fist.setVisible(true);
}


//p_r_fingers
public void p_r_0() {
     p_r_one.setVisible(false);
     p_r_two.setVisible(false);
     p_r_three.setVisible(false);
```

```
        p_r_four.setVisible(false);
        p_r_five.setVisible(false);
        p_r_fist.setVisible(true);
    }

    public void p_r_1() {
        p_r_one.setVisible(true);
        p_r_two.setVisible(false);
        p_r_three.setVisible(false);
        p_r_four.setVisible(false);
        p_r_five.setVisible(false);
        p_r_fist.setVisible(false);
    }

    public void p_r_2() {
        p_r_one.setVisible(false);
        p_r_two.setVisible(true);
        p_r_three.setVisible(false);
        p_r_four.setVisible(false);
        p_r_five.setVisible(false);
        p_r_fist.setVisible(false);
    }

    public void p_r_3() {
        p_r_one.setVisible(false);
        p_r_two.setVisible(false);
        p_r_three.setVisible(true);
        p_r_four.setVisible(false);
        p_r_five.setVisible(false);
        p_r_fist.setVisible(false);
    }

    public void p_r_4() {
        p_r_one.setVisible(false);
        p_r_two.setVisible(false);
        p_r_three.setVisible(false);
        p_r_four.setVisible(true);
        p_r_five.setVisible(false);
        p_r_fist.setVisible(false);
    }

    public void p_r_5_0() {
        p_r_one.setVisible(false);
        p_r_two.setVisible(false);
        p_r_three.setVisible(false);
```

```
        p_r_four.setVisible(false);
        p_r_five.setVisible(true);
        delay(1000);
        p_r_five.setVisible(false);
        p_r_fist.setVisible(true);
    }

    public void repaint() {
        loading();

        if(cp_l_h.size()==0) {
        cp_l_5_0();
       }
       else if(cp_l_h.size()==1) {
        cp_l_1();
       }
       else if(cp_l_h.size()==2) {
        cp_l_2();
       }
       else if(cp_l_h.size()==3) {
        cp_l_3();
       }
       else if(cp_l_h.size()==4) {
        cp_l_4();
       }


        if(cp_r_h.size()==0) {
             cp_r_5_0();
       }
       else if(cp_r_h.size()==1) {
        cp_r_1();
       }
       else if(cp_r_h.size()==2) {
        cp_r_2();
       }
       else if(cp_r_h.size()==3) {
        cp_r_3();
       }
       else if(cp_r_h.size()==4) {
        cp_r_4();
       }


        if(p_l_h.size()==0) {
```

```
            p_l_5_0();
           }
           else if(p_l_h.size()==1) {
            p_l_1();
           }
           else if(p_l_h.size()==2) {
            p_l_2();
           }
           else if(p_l_h.size()==3) {
            p_l_3();
           }
           else if(p_l_h.size()==4) {
            p_l_4();
           }


            if(p_r_h.size()==0) {
                 p_r_5_0();
           }
           else if(p_r_h.size()==1) {
            p_r_1();
           }
           else if(p_r_h.size()==2) {
            p_r_2();
           }
           else if(p_r_h.size()==3) {
            p_r_3();
           }
           else if(p_r_h.size()==4) {
            p_r_4();
           }
     }
}
```

PVP.java

```
package sahu.chopstick_game;

import javax.swing.JFrame;
import javax.swing.JButton;
import java.awt.Font;
import java.awt.event.ActionListener;
```

```
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.io.BufferedReader;
import java.io.FileReader;
import java.awt.event.ActionEvent;
import javax.swing.JPanel;
import java.awt.Toolkit;
import java.awt.Color;

import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.ImageIcon;
import javax.swing.border.CompoundBorder;

import org.apache.commons.mail.EmailException;

import javax.swing.border.BevelBorder;
import javax.swing.SwingConstants;

import java.util.Random;
import java.util.Vector;

import javax.swing.JTextArea;

@SuppressWarnings("unused")
public class PVP implements ActionListener,MouseListener,Runnable{

     private JFrame pvp_frm;

     private JButton quit_btn,
                     p1_l_btn,
                     p1_r_btn,
                     p2_r_btn,
                     p2_l_btn,
                     split_btn,
                     attack_btn,
                     done_btn;

     private JPanel l_pnl,
                     r_pnl,
                     main_pnl;

     private JLabel p_l_fist,
                     p_l_five,
                     p_r_fist,
```

```
                        p_l_one,
                        p_l_two,
                        p_l_three,
                        p_l_four,
                        p_r_five,
                        p_r_three,
                        p_r_four,
                        p_r_two,
                        p_r_one,
                        cp_r_fist,
                        cp_r_five,
                        cp_r_four,
                        cp_r_three,
                        cp_r_two,
                        cp_r_one,
                        cp_l_fist,
                        cp_l_five,
                        cp_l_four,
                        cp_l_three,
                        cp_l_one,
                        cp_l_two,
                        w_l;

    private JTextArea msg;

    Random r=new Random();

    private boolean
running=true,turn,p1_attack,p1_split,p2_attack,p2_split,attack,spli
t,p1_l,p1_r,p2_l,p2_r,t;

    private Vector<Integer> p1_l_h=new Vector<Integer>(5),
                        p1_r_h=new Vector<Integer>(5),
                        p2_l_h=new Vector<Integer>(5),
                        p2_r_h=new Vector<Integer>(5);

    @SuppressWarnings("unused")
    public int i,cp1,cp2,prp1l,prp1r,
                  cph,rc,prp2l,prp2r,x;


    BufferedReader br=null;

    public void delay(int n) {
    try {
```

```
                Thread.sleep(n);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        public PVP() throws Exception{
            br=new BufferedReader(new FileReader("settings.txt"));
            initialize();
            Thread t=new Thread(this);
            t.start();
        }

        @Override
        public void run() {
            delay(1000);
            try {
                Gameloop();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        private synchronized void initialize() throws Exception{
            pvp_frm= new JFrame();
            pvp_frm.setTitle("Player Vs Player");

        pvp_frm.setIconImage(Toolkit.getDefaultToolkit().getImage(getC
lass().getResource("logo_F.png")));
            pvp_frm.setResizable(false);
            pvp_frm.setBounds(150, 20, 1250, 785);
            pvp_frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            pvp_frm.setUndecorated(true);
            pvp_frm.getContentPane().setLayout(null);

            quit_btn = new JButton("Quit");
            quit_btn.addActionListener(this);
            quit_btn.setForeground(new Color(51, 0, 153));
            quit_btn.setFont(new Font("Tahoma", Font.BOLD, 30));
            quit_btn.setBackground(new Color(255, 204, 0));
            quit_btn.setBounds(10, 721, 138, 56);
            quit_btn.setFocusable(false);
            quit_btn.addMouseListener(this);
            pvp_frm.getContentPane().add(quit_btn);
```

```
        r_pnl = new JPanel();
        r_pnl.setBorder(new CompoundBorder(new
BevelBorder(BevelBorder.RAISED,
                                                    new
Color(0, 0, 128), new Color(0, 0, 205),
                                                    new
Color(0, 0, 139), new Color(0, 0, 255)),
                                                new
BevelBorder(BevelBorder.LOWERED,
                                                    new
Color(0, 0, 205), new Color(0, 0, 255),
                                                    new
Color(0, 0, 128), new Color(0, 0, 139))));
        r_pnl.setBounds(670, 50, 516, 447);
        pvp_frm.getContentPane().add(r_pnl);
        r_pnl.setLayout(null);

        p2_r_btn = new JButton("Right");
        p2_r_btn.setForeground(new Color(0, 255, 0));
        p2_r_btn.addActionListener(this);
        p2_r_btn.setEnabled(false);
        p2_r_btn.setFont(new Font("Tahoma", Font.BOLD, 25));
        p2_r_btn.setBackground(new Color(0, 0, 204));
        p2_r_btn.setBounds(319, 100, 114, 45);
        p2_r_btn.setFocusable(false);
        p2_r_btn.addMouseListener(this);
        r_pnl.add(p2_r_btn);

        p2_l_btn = new JButton("Left");
        p2_l_btn.setForeground(new Color(0, 255, 0));
        p2_l_btn.addActionListener(this);
        p2_l_btn.setEnabled(false);
        p2_l_btn.setFont(new Font("Tahoma", Font.BOLD, 25));
        p2_l_btn.setBackground(new Color(51, 51, 153));
        p2_l_btn.setBounds(319, 291, 114, 45);
        p2_l_btn.setFocusable(false);
        p2_l_btn.addMouseListener(this);
        r_pnl.add(p2_l_btn);

        cp_r_fist = new JLabel("");
        cp_r_fist.setIcon(new
ImageIcon(getClass().getResource("cp_r_fist.png")));
        cp_r_fist.setHorizontalAlignment(SwingConstants.CENTER);
        cp_r_fist.setBounds(105, 61, 130, 128);
        cp_r_fist.setVisible(false);
```

```
        r_pnl.add(cp_r_fist);

        cp_r_five = new JLabel("");
        cp_r_five.setIcon(new
ImageIcon(getClass().getResource("cp_r_five.png")));
        cp_r_five.setHorizontalAlignment(SwingConstants.CENTER);
        cp_r_five.setBounds(105, 61, 130, 128);
        cp_r_five.setVisible(false);
        r_pnl.add(cp_r_five);

        cp_r_four = new JLabel("");
        cp_r_four.setIcon(new
ImageIcon(getClass().getResource("cp_r_four.png")));
        cp_r_four.setHorizontalAlignment(SwingConstants.CENTER);
        cp_r_four.setBounds(105, 61, 130, 128);
        cp_r_four.setVisible(false);
        r_pnl.add(cp_r_four);

        cp_r_three = new JLabel("");
        cp_r_three.setIcon(new
ImageIcon(getClass().getResource("cp_r_three.png")));
        cp_r_three.setHorizontalAlignment(SwingConstants.CENTER);
        cp_r_three.setBounds(105, 61, 130, 128);
        cp_r_three.setVisible(false);
        r_pnl.add(cp_r_three);

        cp_r_two = new JLabel("");
        cp_r_two.setIcon(new
ImageIcon(getClass().getResource("cp_r_two.png")));
        cp_r_two.setHorizontalAlignment(SwingConstants.CENTER);
        cp_r_two.setBounds(105, 61, 130, 128);
        cp_r_two.setVisible(false);
        r_pnl.add(cp_r_two);

        cp_r_one = new JLabel("");
        cp_r_one.setIcon(new
ImageIcon(getClass().getResource("cp_r_one.png")));
        cp_r_one.setHorizontalAlignment(SwingConstants.CENTER);
        cp_r_one.setBounds(105, 61, 130, 128);
        r_pnl.add(cp_r_one);

        cp_l_fist = new JLabel("");
        cp_l_fist.setIcon(new
ImageIcon(getClass().getResource("cp_l_fist.png")));
        cp_l_fist.setHorizontalAlignment(SwingConstants.CENTER);
```

```
        cp_l_fist.setBounds(105, 252, 130, 128);
        cp_l_fist.setVisible(false);
        r_pnl.add(cp_l_fist);

        cp_l_five = new JLabel("");
        cp_l_five.setIcon(new
ImageIcon(getClass().getResource("cp_l_five.png")));
        cp_l_five.setHorizontalAlignment(SwingConstants.CENTER);
        cp_l_five.setBounds(105, 252, 130, 128);
        cp_l_five.setVisible(false);
        r_pnl.add(cp_l_five);

        cp_l_four = new JLabel("");
        cp_l_four.setIcon(new
ImageIcon(getClass().getResource("cp_l_four.png")));
        cp_l_four.setHorizontalAlignment(SwingConstants.CENTER);
        cp_l_four.setBounds(105, 252, 130, 128);
        cp_l_four.setVisible(false);
        r_pnl.add(cp_l_four);

        cp_l_three = new JLabel("");
        cp_l_three.setIcon(new
ImageIcon(getClass().getResource("cp_l_three.png")));
        cp_l_three.setHorizontalAlignment(SwingConstants.CENTER);
        cp_l_three.setBounds(105, 252, 130, 128);
        cp_l_three.setVisible(false);
        r_pnl.add(cp_l_three);

        cp_l_two = new JLabel("");
        cp_l_two.setIcon(new
ImageIcon(getClass().getResource("cp_l_two.png")));
        cp_l_two.setHorizontalAlignment(SwingConstants.CENTER);
        cp_l_two.setBounds(105, 252, 130, 128);
        cp_l_two.setVisible(false);
        r_pnl.add(cp_l_two);

        cp_l_one = new JLabel("");
        cp_l_one.setIcon(new
ImageIcon(getClass().getResource("cp_l_one.png")));
        cp_l_one.setHorizontalAlignment(SwingConstants.CENTER);
        cp_l_one.setBounds(105, 252, 130, 128);
        r_pnl.add(cp_l_one);

        l_pnl = new JPanel();
        l_pnl.setLayout(null);
```

```
        l_pnl.setBorder(new CompoundBorder(new
BevelBorder(BevelBorder.RAISED,
                                                    new
Color(0, 0, 128), new Color(0, 0, 205),
                                                    new
Color(0, 0, 139), new Color(0, 0, 255)),
                                                new
BevelBorder(BevelBorder.LOWERED,
                                                    new
Color(0, 0, 205), new Color(0, 0, 255),
                                                    new
Color(0, 0, 128), new Color(0, 0, 139))));
        l_pnl.setBounds(50, 50, 516, 447);
        pvp_frm.getContentPane().add(l_pnl);

        p1_l_btn = new JButton("Left");
        p1_l_btn.setForeground(new Color(0, 255, 0));
        p1_l_btn.addActionListener(this);
        p1_l_btn.setEnabled(false);
        p1_l_btn.setBackground(new Color(51, 51, 153));
        p1_l_btn.setFont(new Font("Tahoma", Font.BOLD, 25));
        p1_l_btn.setBounds(85, 100, 114, 45);
        p1_l_btn.setFocusable(false);
        p1_l_btn.addMouseListener(this);
        l_pnl.add(p1_l_btn);

        p1_r_btn = new JButton("Right");
        p1_r_btn.setForeground(new Color(0, 255, 0));
        p1_r_btn.addActionListener(this);
        p1_r_btn.setEnabled(false);
        p1_r_btn.setFont(new Font("Tahoma", Font.BOLD, 25));
        p1_r_btn.setBackground(new Color(0, 0, 204));
        p1_r_btn.setBounds(85, 291, 114, 45);
        p1_r_btn.setFocusable(false);
        p1_r_btn.addMouseListener(this);
        l_pnl.add(p1_r_btn);

        p_l_fist = new JLabel("");
        p_l_fist.setIcon(new
ImageIcon(getClass().getResource("p_l_fist.png")));
        p_l_fist.setHorizontalAlignment(SwingConstants.CENTER);
        p_l_fist.setBounds(299, 61, 130, 128);
        p_l_fist.setVisible(false);
        l_pnl.add(p_l_fist);
```

```
        p_l_five = new JLabel("");
        p_l_five.setIcon(new
ImageIcon(getClass().getResource("p_l_five.png")));
        p_l_five.setHorizontalAlignment(SwingConstants.CENTER);
        p_l_five.setBounds(299, 61, 130, 128);
        p_l_five.setVisible(false);
        l_pnl.add(p_l_five);

        p_l_four = new JLabel("");
        p_l_four.setIcon(new
ImageIcon(getClass().getResource("p_l_four.png")));
        p_l_four.setHorizontalAlignment(SwingConstants.CENTER);
        p_l_four.setBounds(299, 61, 130, 128);
        p_l_four.setVisible(false);
        l_pnl.add(p_l_four);

        p_l_three = new JLabel("");
        p_l_three.setIcon(new
ImageIcon(getClass().getResource("p_l_three.png")));
        p_l_three.setHorizontalAlignment(SwingConstants.CENTER);
        p_l_three.setBounds(299, 61, 130, 128);
        p_l_three.setVisible(false);
        l_pnl.add(p_l_three);

        p_l_two = new JLabel("");
        p_l_two.setIcon(new
ImageIcon(getClass().getResource("p_l_two.png")));
        p_l_two.setHorizontalAlignment(SwingConstants.CENTER);
        p_l_two.setBounds(299, 61, 130, 128);
        p_l_two.setVisible(false);
        l_pnl.add(p_l_two);

        p_l_one = new JLabel("");
        p_l_one.setIcon(new
ImageIcon(getClass().getResource("p_l_one.png")));
        p_l_one.setHorizontalAlignment(SwingConstants.CENTER);
        p_l_one.setBounds(299, 61, 130, 128);
        l_pnl.add(p_l_one);

        p_r_fist = new JLabel("");
        p_r_fist.setIcon(new
ImageIcon(getClass().getResource("p_r_fist.png")));
        p_r_fist.setHorizontalAlignment(SwingConstants.CENTER);
        p_r_fist.setBounds(299, 252, 130, 128);
        p_r_fist.setVisible(false);
```

```
        l_pnl.add(p_r_fist);

        p_r_five = new JLabel("");
        p_r_five.setIcon(new
ImageIcon(getClass().getResource("p_r_five.png")));
        p_r_five.setHorizontalAlignment(SwingConstants.CENTER);
        p_r_five.setBounds(299, 252, 130, 128);
        p_r_five.setVisible(false);
        l_pnl.add(p_r_five);

        p_r_four = new JLabel("");
        p_r_four.setIcon(new
ImageIcon(getClass().getResource("p_r_four.png")));
        p_r_four.setHorizontalAlignment(SwingConstants.CENTER);
        p_r_four.setBounds(299, 252, 130, 128);
        p_r_four.setVisible(false);
        l_pnl.add(p_r_four);

        p_r_three = new JLabel("");
        p_r_three.setIcon(new
ImageIcon(getClass().getResource("p_r_three.png")));
        p_r_three.setHorizontalAlignment(SwingConstants.CENTER);
        p_r_three.setBounds(299, 252, 130, 128);
        p_r_three.setVisible(false);
        l_pnl.add(p_r_three);

        p_r_two = new JLabel("");
        p_r_two.setIcon(new
ImageIcon(getClass().getResource("p_r_two.png")));
        p_r_two.setHorizontalAlignment(SwingConstants.CENTER);
        p_r_two.setBounds(299, 252, 130, 128);
        p_r_two.setVisible(false);
        l_pnl.add(p_r_two);

        p_r_one = new JLabel("");
        p_r_one.setIcon(new
ImageIcon(getClass().getResource("p_r_one.png")));
        p_r_one.setHorizontalAlignment(SwingConstants.CENTER);
        p_r_one.setBounds(299, 252, 130, 128);
        l_pnl.add(p_r_one);

        JLabel lplayer_lbl = new JLabel("Player1");
        lplayer_lbl.setForeground(new Color(255, 215, 0));
        lplayer_lbl.setFont(new Font("Tahoma", Font.BOLD, 35));
```

```
        lplayer_lbl.setHorizontalAlignment(SwingConstants.CENTER);
        lplayer_lbl.setBounds(197, 0, 193, 50);
        pvp_frm.getContentPane().add(lplayer_lbl);

        JLabel rplayer_lbl = new JLabel("Player2");

        rplayer_lbl.setHorizontalAlignment(SwingConstants.CENTER);
        rplayer_lbl.setForeground(new Color(255, 215, 0));
        rplayer_lbl.setFont(new Font("Tahoma", Font.BOLD, 35));
        rplayer_lbl.setBounds(848, 0, 182, 50);
        pvp_frm.getContentPane().add(rplayer_lbl);

        JLabel vs_lbl = new JLabel("Vs");
        vs_lbl.setHorizontalAlignment(SwingConstants.CENTER);
        vs_lbl.setForeground(new Color(255, 215, 0));
        vs_lbl.setFont(new Font("Tahoma", Font.BOLD, 30));
        vs_lbl.setBounds(576, 261, 84, 40);
        pvp_frm.getContentPane().add(vs_lbl);

        main_pnl = new JPanel();
        main_pnl.setLayout(null);
        main_pnl.setBorder(new CompoundBorder(new
BevelBorder(BevelBorder.RAISED,
                                                    new
Color(0, 0, 128), new Color(0, 0, 205),
                                                    new
Color(0, 0, 139), new Color(0, 0, 255)),
                                                new
BevelBorder(BevelBorder.LOWERED,
                                                    new
Color(0, 0, 205), new Color(0, 0, 255),
                                                    new
Color(0, 0, 128), new Color(0, 0, 139))));
        main_pnl.setBackground(new Color(30, 144, 255));
        main_pnl.setBounds(50, 521, 1136, 183);
        pvp_frm.getContentPane().add(main_pnl);

        msg = new JTextArea("Game Started");
        msg.setWrapStyleWord(true);
        msg.setForeground(Color.YELLOW);
        msg.setFont(new Font("Tahoma", Font.BOLD | Font.ITALIC,
32));
        msg.setBounds(10, 10, 1116, 163);
        msg.setVisible(true);
```

```
        msg.setEditable(false);
        msg.setOpaque(false);
        main_pnl.add(msg);

        attack_btn = new JButton("Attack");
        attack_btn.setBackground(new Color(51, 102, 102));
        attack_btn.setFocusable(false);
        attack_btn.addActionListener(this);
        attack_btn.setForeground(new Color(255, 255, 51));
        attack_btn.setFont(new Font("Tahoma", Font.BOLD, 30));
        attack_btn.setBounds(675, 90, 157, 59);
        attack_btn.setVisible(false);
        attack_btn.addMouseListener(this);
        msg.add(attack_btn);

        split_btn = new JButton("Split");
        split_btn.setBackground(new Color(51, 102, 102));
        split_btn.setFocusable(false);
        split_btn.addActionListener(this);
        split_btn.setFont(new Font("Tahoma", Font.BOLD, 30));
        split_btn.setForeground(new Color(255, 255, 51));
        split_btn.setBounds(291, 90, 157, 59);
        split_btn.setVisible(false);
        split_btn.addMouseListener(this);
        msg.add(split_btn);

        done_btn = new JButton("Done");
        done_btn.setForeground(new Color(255, 255, 51));
        done_btn.setBackground(new Color(51, 102, 102));
        done_btn.addActionListener(this);
        done_btn.setFont(new Font("Tahoma", Font.BOLD, 30));
        done_btn.setBounds(493, 90, 139, 59);
        done_btn.setFocusable(false);
        done_btn.setVisible(false);
        done_btn.addMouseListener(this);
        msg.add(done_btn);

        w_l = new JLabel("");
        w_l.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
45));
        w_l.setOpaque(true);
        w_l.setHorizontalAlignment(SwingConstants.CENTER);
        w_l.setForeground(Color.YELLOW);
        w_l.setBackground(Color.BLACK);
        w_l.setBounds(382, 10, 371, 86);
```

```
        w_l.setVisible(false);
        main_pnl.add(w_l);

        JPanel lplayer_pnl = new JPanel();
        lplayer_pnl.setBounds(197, 0, 193, 50);
        pvp_frm.getContentPane().add(lplayer_pnl);
        lplayer_pnl.setLayout(null);
        lplayer_pnl.setBorder(new CompoundBorder(new
BevelBorder(BevelBorder.RAISED,
                                                 new
Color(0, 0, 128), new Color(0, 0, 205),
                                                 new
Color(0, 0, 139), new Color(0, 0, 255)),
                                               new
BevelBorder(BevelBorder.LOWERED,
                                                 new
Color(0, 0, 205), new Color(0, 0, 255),
                                                 new
Color(0, 0, 128), new Color(0, 0, 139))));
        lplayer_pnl.setBackground(new Color(30, 144, 255));

        JPanel player_r_pnl = new JPanel();
        player_r_pnl.setLayout(null);
        player_r_pnl.setBorder(new CompoundBorder(new
BevelBorder(BevelBorder.RAISED,
                                                 new
Color(0, 0, 128), new Color(0, 0, 205),
                                                 new
Color(0, 0, 139), new Color(0, 0, 255)),
                                               new
BevelBorder(BevelBorder.LOWERED,
                                                 new
Color(0, 0, 205), new Color(0, 0, 255),
                                                 new
Color(0, 0, 128), new Color(0, 0, 139))));
        player_r_pnl.setBackground(new Color(30, 144, 255));
        player_r_pnl.setBounds(834, 0, 208, 50);
        pvp_frm.getContentPane().add(player_r_pnl);

        JPanel vs_pnl = new JPanel();
        vs_pnl.setLayout(null);
        vs_pnl.setBorder(new CompoundBorder(new
BevelBorder(BevelBorder.RAISED,
                                               new
Color(0, 0, 128), new Color(0, 0, 205),
```

```
                                                                new
Color(0, 0, 139), new Color(0, 0, 255)),
                                                            new
BevelBorder(BevelBorder.LOWERED,
                                                                new Color(0, 0,
205), new Color(0, 0, 255),
                                                                new Color(0, 0,
128), new Color(0, 0, 139))));
          vs_pnl.setBackground(new Color(30, 144, 255));
          vs_pnl.setBounds(564, 261, 108, 40);
          pvp_frm.getContentPane().add(vs_pnl);

          if(!br.ready()) {
               pvp_frm.getContentPane().setBackground(new Color(102,
0, 0));
               r_pnl.setBackground(new Color(30, 144, 255));
               l_pnl.setBackground(new Color(30, 144, 255));
               br.close();
          }
          else {
               br.skip(3);
               pvp_frm.getContentPane().setBackground(new
Color(br.read(), br.read(), br.read()));
               br.skip(6);
               r_pnl.setBackground(new Color(br.read(), br.read(),
br.read()));
               l_pnl.setBackground(new Color(br.read(), br.read(),
br.read()));
          }

          pvp_frm.setVisible(true);
     }

     @Override
     public void mouseClicked(MouseEvent e) {


     }

     @Override
     public void mousePressed(MouseEvent e) {


     }
```

```
@Override
public void mouseReleased(MouseEvent e) {


}

@Override
public void mouseEntered(MouseEvent e) {

     if(e.getSource()==quit_btn) {
          quit_btn.setBackground(Color.black);
     }
     if(e.getSource()==split_btn) {
          split_btn.setBackground(Color.black);
     }
     if(e.getSource()==attack_btn) {
          attack_btn.setBackground(Color.black);
     }
     if(e.getSource()==done_btn) {
          done_btn.setBackground(Color.black);
     }
     if(e.getSource()==p1_l_btn) {
          p1_l_btn.setBackground(Color.black);
     }
     if(e.getSource()==p1_r_btn) {
          p1_r_btn.setBackground(Color.black);
     }
     if(e.getSource()==p2_l_btn) {
          p2_l_btn.setBackground(Color.black);
     }
     if(e.getSource()==p2_r_btn) {
          p2_r_btn.setBackground(Color.black);
     }

}

@Override
public void mouseExited(MouseEvent e) {

     if(e.getSource()==quit_btn) {
          quit_btn.setBackground(new Color(255, 204, 0));
     }
     if(e.getSource()==split_btn) {
          split_btn.setBackground(new Color(51, 102, 102));
     }
```

```
            if(e.getSource()==attack_btn) {
                attack_btn.setBackground(new Color(51, 102, 102));
            }
            if(e.getSource()==done_btn) {
                done_btn.setBackground(new Color(51, 102, 102));
            }
            if(e.getSource()==p1_l_btn) {
                p1_l_btn.setBackground(new Color(51, 51, 153));
            }
            if(e.getSource()==p1_r_btn) {
                p1_r_btn.setBackground(new Color(0, 0, 204));
            }
            if(e.getSource()==p2_l_btn) {
                p2_l_btn.setBackground(new Color(51, 51, 153));
            }
            if(e.getSource()==p2_r_btn) {
                p2_r_btn.setBackground(new Color(0, 0, 204));
            }

        }


        public void actionPerformed(ActionEvent e) {

                    if(e.getSource()==quit_btn) {
                        int Q=JOptionPane.showConfirmDialog(pvp_frm,
"Sure want to quit?", "Quit",

        JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
                        if(Q==0) {
                            pvp_frm.dispose();
                            try {
                                new
ChopStick().main_frm.setVisible(true);
                            } catch (Exception e1) {
                                e1.printStackTrace();
                            }
                        }
                    }

                    if(e.getSource()==split_btn) {
                        split=true;
                        if(p1_split) {
                            p1_r_btn.setEnabled(true);
                            p1_l_btn.setEnabled(true);
```

```
                    p2_l_btn.setEnabled(false);
                    p2_r_btn.setEnabled(false);
                    p1_attack=false;
                    p1_l=false;
                    p1_r=false;
                    p2_l=false;
                    p2_r=false;
                    prp1l=p1_l_h.size();
                    prp1r=p1_r_h.size();
               }
               else if(p2_split) {
                    p2_r_btn.setEnabled(true);
                    p2_l_btn.setEnabled(true);
                    p1_l_btn.setEnabled(false);
                    p1_r_btn.setEnabled(false);
                    p2_attack=false;
                    p1_l=false;
                    p1_r=false;
                    p2_l=false;
                    p2_r=false;
                    prp2l=p1_l_h.size();
                    prp2r=p1_r_h.size();
               }
               attack_btn.setVisible(false);
               split_btn.setVisible(false);
               msg.setText("Click on your either of the hand-
buttons[right or left]\n"
                         + "For splitting the fingers by one,\n"
                         + "(One finger will be increased on the
clicked hand-button,\n"
                         + "hence it will decrease one finger on
the opposite)");

          }

          if(e.getSource()==attack_btn) {
               attack=true;
               p1_split=false;
               p2_split=false;
               if(p1_attack) {
                    if(p1_l_h.size()==0) {
                         p1_l_btn.setEnabled(false);
                         p1_r_btn.setEnabled(true);
                    }
                    else if(p1_r_h.size()==0) {
```

```
                    p1_r_btn.setEnabled(false);
                    p1_l_btn.setEnabled(true);
               }
               else {
                    p1_r_btn.setEnabled(true);
                    p1_l_btn.setEnabled(true);
               }
          }
          else if(p2_attack) {
               if(p2_l_h.size()==0) {
                    p2_l_btn.setEnabled(false);
                    p2_r_btn.setEnabled(true);
               }
               else if(p2_r_h.size()==0) {
                    p2_r_btn.setEnabled(false);
                    p2_l_btn.setEnabled(true);
               }
               else {
                    p2_r_btn.setEnabled(true);
                    p2_l_btn.setEnabled(true);
               }
          }

          attack_btn.setVisible(false);
          split_btn.setVisible(false);
          msg.setText("First click on either of your hand-
buttons[right or left]\n"
                         + "and then opponent's for the
attack");
     }


     if(e.getSource()==p1_l_btn&&p1_split) {
          msg.setText(".....Splitting.....");

          if(p1_l_h.size()>=4)
               JOptionPane.showMessageDialog(l_pnl,
                         "You can't kill your own
hand!,\nIn splitting decrement of fingers isn't allowed",
                         "Fingers' Decrement",
JOptionPane.WARNING_MESSAGE);
          else if(p1_r_h.size()==0) {
               JOptionPane.showMessageDialog(l_pnl,
                         "No fingers left to transfer\nYou
got a fist on right hand",
```

```
                                        "No more fingers",
JOptionPane.WARNING_MESSAGE);
                    }
                    else {
                         p1_l_h.add(1);
                         p1_r_h.removeElement(p1_r_h.lastElement());
                         p1_split();
                    }
                    done_btn.setVisible(true);
               }
               if(e.getSource()==p1_r_btn&&p1_split) {
                    msg.setText(".....Splitting.....");

                    if(p1_r_h.size()>=4)
                         JOptionPane.showMessageDialog(l_pnl,
                                        "You can't kill your own
hand!,\nIn splitting decrement of fingers isn't allowed",
                                        "Fingers' Decrement",
JOptionPane.WARNING_MESSAGE);
                    else if(p1_l_h.size()==0) {
                         JOptionPane.showMessageDialog(l_pnl,
                                        "No fingers left to transfer\nYou
got a fist on left hand",
                                        "No more fingers",
JOptionPane.WARNING_MESSAGE);
                    }
                    else {
                         p1_r_h.add(1);
                         p1_l_h.removeElement(p1_l_h.lastElement());
                         p1_split();
                    }
                    done_btn.setVisible(true);
               }

               if(e.getSource()==p2_l_btn&&p2_split) {
                    msg.setText(".....Splitting.....");

                    if(p2_l_h.size()>=4)
                         JOptionPane.showMessageDialog(r_pnl,
                                        "You can't kill your own
hand!,\nIn splitting decrement of fingers isn't allowed",
                                        "Fingers' Decrement",
JOptionPane.WARNING_MESSAGE);
                    else if(p2_r_h.size()==0) {
                         JOptionPane.showMessageDialog(r_pnl,
```

```
                                        "No fingers left to transfer\nYou
got a fist on right hand",
                                        "No more fingers",
JOptionPane.WARNING_MESSAGE);
                    }
                    else {
                         p2_l_h.add(1);
                         p2_r_h.removeElement(p2_r_h.lastElement());
                         p2_split();
                    }
                    done_btn.setVisible(true);
               }
               if(e.getSource()==p2_r_btn&&p2_split) {
                    msg.setText(".....Splitting.....");

                    if(p2_r_h.size()>=4)
                         JOptionPane.showMessageDialog(r_pnl,
                                        "You can't kill your own
hand!,\nIn splitting decrement of fingers isn't allowed",
                                        "Fingers' Decrement",
JOptionPane.WARNING_MESSAGE);
                    else if(p2_l_h.size()==0) {
                         JOptionPane.showMessageDialog(r_pnl,
                                        "No fingers left to transfer\nYou
got a fist on left hand", "No more fingers",
                                        JOptionPane.WARNING_MESSAGE);
                    }
                    else {
                         p2_r_h.add(1);
                         p2_l_h.removeElement(p2_l_h.lastElement());
                         p2_split();
                    }
                    done_btn.setVisible(true);
               }


               if(e.getSource()==done_btn) {
                    if(p1_split) {

     if(p1_l_h.size()==prp1r&&p1_r_h.size()==prp1l ||
p1_l_h.size()==prp1l&&p1_r_h.size()==prp1r) {
                              JOptionPane.showMessageDialog(l_pnl,
"Splitting is same as previous State!",
                                        "Identical states",
JOptionPane.WARNING_MESSAGE);
```

```
                    }
                    else {
                         p1_r_btn.setEnabled(false);
                         p1_l_btn.setEnabled(false);
                         p2_r_btn.setEnabled(false);
                         p2_l_btn.setEnabled(false);
                         done_btn.setVisible(false);
                         p1_split=false;
                         p2_split=false;
                         p1_attack=false;
                         p2_attack=false;

                         split=false;
                         if(t)
                              t=false;
                         else
                              t=true;
                    }
               }
               else if(p2_split) {

     if(p2_l_h.size()==prp2r&&p2_r_h.size()==prp2l ||
p2_l_h.size()==prp2l&&p2_r_h.size()==prp2r) {
                         JOptionPane.showMessageDialog(r_pnl,
"Splitting is same as previous State!",
                                   "Identical states",
JOptionPane.WARNING_MESSAGE);
                    }
                    else {
                         p1_r_btn.setEnabled(false);
                         p1_l_btn.setEnabled(false);
                         p2_r_btn.setEnabled(false);
                         p2_l_btn.setEnabled(false);
                         done_btn.setVisible(false);
                         p1_split=false;
                         p2_split=false;
                         p1_attack=false;
                         p2_attack=false;

                         split=false;
                         if(t)
                              t=false;
                         else
                              t=true;
                    }
```

```
            }
        }


        if(e.getSource()==p1_l_btn&&p1_attack) {
            if(p2_r_h.size()==0) {
                p2_r_btn.setEnabled(false);
                p2_l_btn.setEnabled(true);
            }
            else if(p2_l_h.size()==0) {
                p2_l_btn.setEnabled(false);
                p2_r_btn.setEnabled(true);
            }
            else {
                p2_r_btn.setEnabled(true);
                p2_l_btn.setEnabled(true);
            }
            msg.setText(".....Player1's left hand.....");
            p1_l=true;
            p1_r=false;
        }
        if(e.getSource()==p1_r_btn&&p1_attack) {
            if(p2_r_h.size()==0) {
                p2_r_btn.setEnabled(false);
                p2_l_btn.setEnabled(true);
            }
            else if(p2_l_h.size()==0) {
                p2_l_btn.setEnabled(false);
                p2_r_btn.setEnabled(true);
            }
            else {
                p2_r_btn.setEnabled(true);
                p2_l_btn.setEnabled(true);
            }
            msg.setText(".....Player1's right hand.....");
            p1_r=true;
            p1_l=false;
        }

        if(e.getSource()==p2_l_btn&&p2_attack) {
            if(p1_r_h.size()==0) {
                p1_r_btn.setEnabled(false);
                p1_l_btn.setEnabled(true);
            }
            else if(p1_l_h.size()==0) {
```

```
                    p1_l_btn.setEnabled(false);
                    p1_r_btn.setEnabled(true);
               }
               else {
                    p1_r_btn.setEnabled(true);
                    p1_l_btn.setEnabled(true);
               }
               msg.setText(".....Player2's left hand.....");
               p2_l=true;
               p2_r=false;
          }
          if(e.getSource()==p2_r_btn&&p2_attack) {
               if(p1_r_h.size()==0) {
                    p1_r_btn.setEnabled(false);
                    p1_l_btn.setEnabled(true);
               }
               else if(p1_l_h.size()==0) {
                    p1_l_btn.setEnabled(false);
                    p1_r_btn.setEnabled(true);
               }
               else {
                    p1_r_btn.setEnabled(true);
                    p1_l_btn.setEnabled(true);
               }
               msg.setText(".....Player2's right hand.....");
               p2_r=true;
               p2_l=false;
          }

          //player1 hit
          if(e.getSource()==p2_l_btn&&p1_l) {
               if(p1_attack) {
                    msg.setText(".....Player1's left hand ->
Player2's left hand.....");

                    p1_l_btn.setEnabled(false);
                    p1_r_btn.setEnabled(false);
                    p2_l_btn.setEnabled(false);
                    p2_r_btn.setEnabled(false);

                    P1_L_L_HIT();

                    p1_split=false;
                    p2_split=false;
                    p1_attack=false;
```

```
                    p2_attack=false;
                    attack=false;
                    t=false;
               }

          }
          if(e.getSource()==p2_r_btn&&p1_l) {
               if(p1_attack) {
                    msg.setText(".....Player1's left hand ->
Player2's right hand.....");

                    p1_l_btn.setEnabled(false);
                    p1_r_btn.setEnabled(false);
                    p2_l_btn.setEnabled(false);
                    p2_r_btn.setEnabled(false);

                    P1_L_R_HIT();

                    p1_split=false;
                    p2_split=false;
                    p1_attack=false;
                    p2_attack=false;
                    attack=false;
                    t=false;
               }

          }
          if(e.getSource()==p2_l_btn&&p1_r) {
               if(p1_attack) {
                    msg.setText(".....Player1's right hand ->
Player2's left hand.....");

                    p1_l_btn.setEnabled(false);
                    p1_r_btn.setEnabled(false);
                    p2_l_btn.setEnabled(false);
                    p2_r_btn.setEnabled(false);

                    P1_R_L_HIT();

                    p1_split=false;
                    p2_split=false;
                    p1_attack=false;
                    p2_attack=false;
                    attack=false;
                    t=false;
```

```
                    }

               }
               if(e.getSource()==p2_r_btn&&p1_r) {
                    if(p1_attack) {
                         msg.setText(".....Player1's right hand ->
Player2's right hand.....");

                         p1_l_btn.setEnabled(false);
                         p1_r_btn.setEnabled(false);
                         p2_l_btn.setEnabled(false);
                         p2_r_btn.setEnabled(false);

                         P1_R_R_HIT();

                         p1_split=false;
                         p2_split=false;
                         p1_attack=false;
                         p2_attack=false;
                         attack=false;
                         t=false;
                    }

               }

               //player2 hit
               if(e.getSource()==p1_l_btn&&p2_l) {
                    if(p2_attack) {
                         msg.setText(".....Player2's left hand ->
Player1's left hand.....");

                         p1_l_btn.setEnabled(false);
                         p1_r_btn.setEnabled(false);
                         p2_l_btn.setEnabled(false);
                         p2_r_btn.setEnabled(false);

                         P2_L_L_HIT();

                         p1_split=false;
                         p2_split=false;
                         p1_attack=false;
                         p2_attack=false;
                         attack=false;
                         t=true;
                    }
```

```
                }
                if(e.getSource()==p1_r_btn&&p2_l) {
                     if(p2_attack) {
                          msg.setText(".....Player2's left hand ->
Player1's right hand.....");

                          p1_l_btn.setEnabled(false);
                          p1_r_btn.setEnabled(false);
                          p2_l_btn.setEnabled(false);
                          p2_r_btn.setEnabled(false);

                          P2_L_R_HIT();

                          p1_split=false;
                          p2_split=false;
                          p1_attack=false;
                          p2_attack=false;
                          attack=false;
                          t=true;
                     }

                }
                if(e.getSource()==p1_l_btn&&p2_r) {
                     if(p2_attack) {
                          msg.setText(".....Player2's right hand ->
Player2's left hand.....");

                          p1_l_btn.setEnabled(false);
                          p1_r_btn.setEnabled(false);
                          p2_l_btn.setEnabled(false);
                          p2_r_btn.setEnabled(false);

                          P2_R_L_HIT();

                          p1_split=false;
                          p2_split=false;
                          p1_attack=false;
                          p2_attack=false;
                          attack=false;
                          t=true;
                     }

                }
                if(e.getSource()==p1_r_btn&&p2_r) {
```

```
                    if(p2_attack) {
                         msg.setText(".....Player2's right hand ->
Player1's right hand.....");

                         p1_l_btn.setEnabled(false);
                         p1_r_btn.setEnabled(false);
                         p2_l_btn.setEnabled(false);
                         p2_r_btn.setEnabled(false);

                         P2_R_R_HIT();

                         p1_split=false;
                         p2_split=false;
                         p1_attack=false;
                         p2_attack=false;
                         attack=false;
                         t=true;
                    }

               }

     }

     public void loading() {
          String s=".";
          for(int i=0;i<5;i++) {
               msg.setText(s);
               s+=".";
               delay(150);
          }
     }

     public synchronized void Gameloop() throws
InterruptedException {
          p1_l_h.add(1);
          p1_r_h.add(1);
          p2_l_h.add(1);
          p2_r_h.add(1);
               delay(1000);
               loading();
               msg.setText("...Randomly Choosing the turn!...");
               delay(2000);
               loading();
               if(r.nextInt(2)==0) {
                    loading();
```

```
                    turn=true;
               }
               else {
                    loading();
                    turn=false;
               }

               while(running) {
                    if(p1_l_h.isEmpty() && p1_r_h.isEmpty()) {
                         p2_won();
                    }
                    else if(p2_l_h.isEmpty() && p2_r_h.isEmpty())  {
                         p1_won();
                    }
                    else {
                         if(turn) {
                              p1_turn();
                              t=true;
                              while(t) {
                                   delay(1000);
                              }
                              repaint();
                              if(p1_l_h.isEmpty() &&
p1_r_h.isEmpty()) {
                                   p2_won();
                              }
                              else if(p2_l_h.isEmpty() &&
p2_r_h.isEmpty())  {
                                   p1_won();
                              }
                              else {
                                   turn=false;
                              }

                         }
                         else {
                              p2_turn();
                              t=false;
                              while(!t) {
                                   delay(1000);
                              }
                              repaint();
                              if(p1_l_h.isEmpty() &&
p1_r_h.isEmpty()) {
                                   p2_won();
```

```
                              }
                              else if(p2_l_h.isEmpty() &&
p2_r_h.isEmpty())  {
                                   p1_won();
                              }
                              else {
                                   turn=true;
                              }


                         }
                    }
               }
          }


     public void p1_won() {
          loading();
          p2_l_btn.setVisible(false);
          p2_r_btn.setVisible(false);
          p1_l_btn.setVisible(false);
          p1_r_btn.setVisible(false);
          msg.setVisible(false);
          w_l.setVisible(true);
          w_l.setText("(: Player1 won! :)");
          delay(1000);
          int O=JOptionPane.showConfirmDialog(pvp_frm, "want to
play again ?", "Two players",

     JOptionPane.YES_NO_OPTION ,JOptionPane.QUESTION_MESSAGE);
          if(JOptionPane.YES_OPTION==O) {
               running=false;
               pvp_frm.dispose();
               try {
                    new PWC();
               } catch (Exception e) {
                    e.printStackTrace();
               }
          }
          else if(JOptionPane.NO_OPTION==O) {
               running=false;
               pvp_frm.dispose();
               try {
                    new ChopStick().main_frm.setVisible(true);
               } catch (Exception e) {
```

```
                    e.printStackTrace();
               }
          }
          else {
               running=false;
          }
     }

     public void p2_won() {
          loading();
          p2_l_btn.setVisible(false);
          p2_r_btn.setVisible(false);
          p1_l_btn.setVisible(false);
          p1_r_btn.setVisible(false);
          msg.setVisible(false);
          w_l.setVisible(true);
          w_l.setText("(: Player2 Won! :)");
          delay(1000);
          int O=JOptionPane.showConfirmDialog(pvp_frm, "want to
play again ?", "Two players",

     JOptionPane.YES_NO_OPTION ,JOptionPane.QUESTION_MESSAGE ,
null);
          if(JOptionPane.YES_OPTION==O) {
               running=false;
               pvp_frm.dispose();
               try {
                    new PWC();
               } catch (Exception e) {
                    e.printStackTrace();
               }
          }
          else if(JOptionPane.NO_OPTION==O) {
               running=false;
               pvp_frm.dispose();
               try {
                    new ChopStick().main_frm.setVisible(true);
               } catch (Exception e) {
                    e.printStackTrace();
               }
          }
          else {
               running=false;
          }
     }
```

```
//player1
public void P1_L_L_HIT() {
     x=p1_l_h.size();
    for(i=0;i<x;i++) {
     if(p2_l_h.size()==5) {
          p2_l_h.removeAllElements();
          cp_l_5_0();
     }
     p2_l_h.add(1);
    }
    if(p2_l_h.size()==5) {
     p2_l_h.removeAllElements();
    }

    if(p2_l_h.size()==1) {
     cp_l_1();
    }
    else if(p2_l_h.size()==2) {
     cp_l_2();
    }
    else if(p2_l_h.size()==3) {
     cp_l_3();
    }
    else if(p2_l_h.size()==4) {
     cp_l_4();
    }
}

public void P1_L_R_HIT() {
     x=p1_l_h.size();
    for(i=0;i<x;i++) {
     if(p2_r_h.size()==5) {
          p2_r_h.removeAllElements();
          cp_r_5_0();
     }
     p2_r_h.add(1);
    }
    if(p2_r_h.size()==5) {
     p2_r_h.removeAllElements();
    }

    if(p2_r_h.size()==1) {
     cp_r_1();
```

```
        }
        else if(p2_r_h.size()==2) {
         cp_r_2();
        }
        else if(p2_r_h.size()==3) {
         cp_r_3();
        }
        else if(p2_r_h.size()==4) {
         cp_r_4();
        }
    }

    public void P1_R_L_HIT() {
         x=p1_r_h.size();
        for(i=0;i<x;i++) {
         if(p2_l_h.size()==5) {
              p2_l_h.removeAllElements();
              cp_l_5_0();
         }
         p2_l_h.add(1);
        }
        if(p2_l_h.size()==5) {
         p2_l_h.removeAllElements();
        }

        if(p2_l_h.size()==1) {
         cp_l_1();
        }
        else if(p2_l_h.size()==2) {
         cp_l_2();
        }
        else if(p2_l_h.size()==3) {
         cp_l_3();
        }
        else if(p2_l_h.size()==4) {
         cp_l_4();
        }
    }

    public void P1_R_R_HIT() {
         x=p1_r_h.size();
        for(i=0;i<x;i++) {
         if(p2_r_h.size()==5) {
              p2_r_h.removeAllElements();
              cp_r_5_0();
```

```
        }
        p2_r_h.add(1);
       }
       if(p2_r_h.size()==5) {
        p2_r_h.removeAllElements();
       }

       if(p2_r_h.size()==1) {
        cp_r_1();
       }
       else if(p2_r_h.size()==2) {
        cp_r_2();
       }
       else if(p2_r_h.size()==3) {
        cp_r_3();
       }
       else if(p2_r_h.size()==4) {
        cp_r_4();
       }
}

//player2
public void P2_L_L_HIT()
{
     x=p2_l_h.size();
    for(i=0;i<x;i++) {
     if(p1_l_h.size()==5) {
          p1_l_h.removeAllElements();
          p_l_5_0();
     }
     p1_l_h.add(1);
    }
    if(p1_l_h.size()==5) {
     p1_l_h.removeAllElements();
    }

    if(p1_l_h.size()==1) {
     p_l_1();
    }
    else if(p1_l_h.size()==2) {
     p_l_2();
    }
    else if(p1_l_h.size()==3) {
     p_l_3();
    }
```

```
        else if(p1_l_h.size()==4) {
         p_l_4();
        }
    }

    public void P2_L_R_HIT()
    {
         x=p2_l_h.size();
        for(i=0;i<x;i++) {
         if(p1_r_h.size()==5) {
              p1_r_h.removeAllElements();
              p_r_5_0();
         }
         p1_r_h.add(1);
        }
        if(p1_r_h.size()==5) {
         p1_r_h.removeAllElements();
        }


        if(p1_r_h.size()==1) {
         p_r_1();
        }
        else if(p1_r_h.size()==2) {
         p_r_2();
        }
        else if(p1_r_h.size()==3) {
         p_r_3();
        }
        else if(p1_r_h.size()==4) {
         p_r_4();
        }
    }

    public void P2_R_L_HIT()
    {
         x=p2_r_h.size();
        for(i=0;i<x;i++) {
         if(p1_l_h.size()==5) {
              p1_l_h.removeAllElements();
              p_l_5_0();
         }
         p1_l_h.add(1);
        }
        if(p1_l_h.size()==5) {
```

```
        p1_l_h.removeAllElements();
       }


       if(p1_l_h.size()==1) {
        p_l_1();
       }
       else if(p1_l_h.size()==2) {
        p_l_2();
       }
       else if(p1_l_h.size()==3) {
        p_l_3();
       }
       else if(p1_l_h.size()==4) {
        p_l_4();
       }
}

public void P2_R_R_HIT()
{
        x=p2_r_h.size();

       for(i=0;i<x;i++) {
        if(p1_r_h.size()==5) {
             p1_r_h.removeAllElements();
             p_r_5_0();
        }
        p1_r_h.add(1);
       }
       if(p1_r_h.size()==5) {
        p1_r_h.removeAllElements();
       }

       if(p1_r_h.size()==1) {
        p_r_1();
       }
       else if(p1_r_h.size()==2) {
        p_r_2();
       }
       else if(p1_r_h.size()==3) {
        p_r_3();
       }
       else if(p1_r_h.size()==4) {
        p_r_4();
```

```
            }
        }

    public void p1_turn() {
        p1_attack=true;
        p1_split=true;

        loading();
        msg.setText(".....Player1's turn.....");
        delay(1000);

        l_pnl.setBackground(Color.white);
        delay(500);
        l_pnl.setBackground(new Color(30, 144, 255));

        if(p1_l_h.size()==0&&p1_r_h.size()==1) {
            msg.setText(".....You can only attack.....");
            attack_btn.setVisible(true);
            split_btn.setVisible(true);
            split_btn.setEnabled(false);
        }
        else if(p1_r_h.size()==0&&p1_l_h.size()==1) {
            msg.setText(".....You can only attack.....");
            attack_btn.setVisible(true);
            split_btn.setVisible(true);
            split_btn.setEnabled(false);
        }
        else {
            msg.setText("What do you wanna do ?");
            attack_btn.setVisible(true);
            split_btn.setVisible(true);
            split_btn.setEnabled(true);
        }

    }

    public void p2_turn() {
        p2_attack=true;
        p2_split=true;

        loading();
        msg.setText(".....Player2's turn.....");
        delay(1000);

        r_pnl.setBackground(Color.white);
```

```
        delay(500);
        r_pnl.setBackground(new Color(30, 144, 255));

        if(p2_l_h.size()==0&&p2_r_h.size()==1) {
             msg.setText(".....You can only attack.....");
             attack_btn.setVisible(true);
             split_btn.setVisible(true);
             split_btn.setEnabled(false);
        }
        else if(p2_r_h.size()==0&&p2_l_h.size()==1) {
             msg.setText(".....You can only attack.....");
             attack_btn.setVisible(true);
             split_btn.setVisible(true);
             split_btn.setEnabled(false);
        }
        else {
             msg.setText("What do you wanna do ?");
             attack_btn.setVisible(true);
             split_btn.setVisible(true);
             split_btn.setEnabled(true);
        }
    }

    public void p1_split() {

         if(p1_l_h.size()==0) {
         p_l_0();
        }
        else if(p1_l_h.size()==1) {
         p_l_1();
        }
        else if(p1_l_h.size()==2) {
         p_l_2();
        }
        else if(p1_l_h.size()==3) {
         p_l_3();
        }
        else if(p1_l_h.size()==4) {
         p_l_4();
        }

         if(p1_r_h.size()==0) {
             p_r_0();
        }
        else if(p1_r_h.size()==1) {
```

```
        p_r_1();
       }
       else if(p1_r_h.size()==2) {
        p_r_2();
       }
       else if(p1_r_h.size()==3) {
        p_r_3();
       }
       else if(p1_r_h.size()==4) {
        p_r_4();
       }
  }

  public void p2_split() {
       if(p2_l_h.size()==0) {
       cp_l_0();
       }
       else if(p2_l_h.size()==1) {
        cp_l_1();
       }
       else if(p2_l_h.size()==2) {
        cp_l_2();
       }
       else if(p2_l_h.size()==3) {
        cp_l_3();
       }
       else if(p2_l_h.size()==4) {
        cp_l_4();
       }

        if(p2_r_h.size()==0) {
             cp_r_0();
       }
       else if(p2_r_h.size()==1) {
        cp_r_1();
       }
       else if(p2_r_h.size()==2) {
        cp_r_2();
       }
       else if(p2_r_h.size()==3) {
        cp_r_3();
       }
       else if(p2_r_h.size()==4) {
        cp_r_4();
       }
```

```
    }
    //p2_l_fingers
    public void cp_l_0() {
         cp_l_one.setVisible(false);
         cp_l_two.setVisible(false);
         cp_l_three.setVisible(false);
         cp_l_four.setVisible(false);
         cp_l_five.setVisible(false);
         cp_l_fist.setVisible(true);
    }

    public void cp_l_1() {
         cp_l_one.setVisible(true);
         cp_l_two.setVisible(false);
         cp_l_three.setVisible(false);
         cp_l_four.setVisible(false);
         cp_l_five.setVisible(false);
         cp_l_fist.setVisible(false);
    }

    public void cp_l_2() {
         cp_l_one.setVisible(false);
         cp_l_two.setVisible(true);
         cp_l_three.setVisible(false);
         cp_l_four.setVisible(false);
         cp_l_five.setVisible(false);
         cp_l_fist.setVisible(false);
    }

    public void cp_l_3() {
         cp_l_one.setVisible(false);
         cp_l_two.setVisible(false);
         cp_l_three.setVisible(true);
         cp_l_four.setVisible(false);
         cp_l_five.setVisible(false);
         cp_l_fist.setVisible(false);
    }

    public void cp_l_4() {
         cp_l_one.setVisible(false);
         cp_l_two.setVisible(false);
         cp_l_three.setVisible(false);
         cp_l_four.setVisible(true);
         cp_l_five.setVisible(false);
         cp_l_fist.setVisible(false);
```

```
}

public void cp_l_5_0() {
     cp_l_one.setVisible(false);
     cp_l_two.setVisible(false);
     cp_l_three.setVisible(false);
     cp_l_four.setVisible(false);
     cp_l_five.setVisible(true);
     delay(1000);
     cp_l_five.setVisible(false);
     cp_l_fist.setVisible(true);
}


//p2_r_fingers
public void cp_r_0() {
     cp_r_one.setVisible(false);
     cp_r_two.setVisible(false);
     cp_r_three.setVisible(false);
     cp_r_four.setVisible(false);
     cp_r_five.setVisible(false);
     cp_r_fist.setVisible(true);
}

public void cp_r_1() {
     cp_r_one.setVisible(true);
     cp_r_two.setVisible(false);
     cp_r_three.setVisible(false);
     cp_r_four.setVisible(false);
     cp_r_five.setVisible(false);
     cp_r_fist.setVisible(false);
}

public void cp_r_2() {
     cp_r_one.setVisible(false);
     cp_r_two.setVisible(true);
     cp_r_three.setVisible(false);
     cp_r_four.setVisible(false);
     cp_r_five.setVisible(false);
     cp_r_fist.setVisible(false);
}

public void cp_r_3() {
     cp_r_one.setVisible(false);
     cp_r_two.setVisible(false);
```

```
        cp_r_three.setVisible(true);
        cp_r_four.setVisible(false);
        cp_r_five.setVisible(false);
        cp_r_fist.setVisible(false);
    }

    public void cp_r_4() {
        cp_r_one.setVisible(false);
        cp_r_two.setVisible(false);
        cp_r_three.setVisible(false);
        cp_r_four.setVisible(true);
        cp_r_five.setVisible(false);
        cp_r_fist.setVisible(false);
    }

    public void cp_r_5_0() {
        cp_r_one.setVisible(false);
        cp_r_two.setVisible(false);
        cp_r_three.setVisible(false);
        cp_r_four.setVisible(false);
        cp_r_five.setVisible(true);
        delay(1000);
        cp_r_five.setVisible(false);
        cp_r_fist.setVisible(true);
    }


    //p_l_fingers
    public void p_l_0() {
        p_l_one.setVisible(false);
        p_l_two.setVisible(false);
        p_l_three.setVisible(false);
        p_l_four.setVisible(false);
        p_l_five.setVisible(false);
        p_l_fist.setVisible(true);
    }

    public void p_l_1() {
        p_l_one.setVisible(true);
        p_l_two.setVisible(false);
        p_l_three.setVisible(false);
        p_l_four.setVisible(false);
        p_l_five.setVisible(false);
        p_l_fist.setVisible(false);
    }
```

```
public void p_l_2() {
     p_l_one.setVisible(false);
     p_l_two.setVisible(true);
     p_l_three.setVisible(false);
     p_l_four.setVisible(false);
     p_l_five.setVisible(false);
     p_l_fist.setVisible(false);
}

public void p_l_3() {
     p_l_one.setVisible(false);
     p_l_two.setVisible(false);
     p_l_three.setVisible(true);
     p_l_four.setVisible(false);
     p_l_five.setVisible(false);
     p_l_fist.setVisible(false);
}

public void p_l_4() {
     p_l_one.setVisible(false);
     p_l_two.setVisible(false);
     p_l_three.setVisible(false);
     p_l_four.setVisible(true);
     p_l_five.setVisible(false);
     p_l_fist.setVisible(false);
}

public void p_l_5_0() {
     p_l_one.setVisible(false);
     p_l_two.setVisible(false);
     p_l_three.setVisible(false);
     p_l_four.setVisible(false);
     p_l_five.setVisible(true);
     delay(1000);
     p_l_five.setVisible(false);
     p_l_fist.setVisible(true);
}


//p_r_fingers
public void p_r_0() {
     p_r_one.setVisible(false);
     p_r_two.setVisible(false);
     p_r_three.setVisible(false);
```

```
        p_r_four.setVisible(false);
        p_r_five.setVisible(false);
        p_r_fist.setVisible(true);
    }

    public void p_r_1() {
        p_r_one.setVisible(true);
        p_r_two.setVisible(false);
        p_r_three.setVisible(false);
        p_r_four.setVisible(false);
        p_r_five.setVisible(false);
        p_r_fist.setVisible(false);
    }

    public void p_r_2() {
        p_r_one.setVisible(false);
        p_r_two.setVisible(true);
        p_r_three.setVisible(false);
        p_r_four.setVisible(false);
        p_r_five.setVisible(false);
        p_r_fist.setVisible(false);
    }

    public void p_r_3() {
        p_r_one.setVisible(false);
        p_r_two.setVisible(false);
        p_r_three.setVisible(true);
        p_r_four.setVisible(false);
        p_r_five.setVisible(false);
        p_r_fist.setVisible(false);
    }

    public void p_r_4() {
        p_r_one.setVisible(false);
        p_r_two.setVisible(false);
        p_r_three.setVisible(false);
        p_r_four.setVisible(true);
        p_r_five.setVisible(false);
        p_r_fist.setVisible(false);
    }

    public void p_r_5_0() {
        p_r_one.setVisible(false);
        p_r_two.setVisible(false);
        p_r_three.setVisible(false);
```

```
        p_r_four.setVisible(false);
        p_r_five.setVisible(true);
        delay(1000);
        p_r_five.setVisible(false);
        p_r_fist.setVisible(true);
    }

    public void repaint() {
        loading();

        if(p2_l_h.size()==0) {
        cp_l_5_0();
       }
       else if(p2_l_h.size()==1) {
        cp_l_1();
       }
       else if(p2_l_h.size()==2) {
        cp_l_2();
       }
       else if(p2_l_h.size()==3) {
        cp_l_3();
       }
       else if(p2_l_h.size()==4) {
        cp_l_4();
       }


        if(p2_r_h.size()==0) {
             cp_r_5_0();
       }
       else if(p2_r_h.size()==1) {
        cp_r_1();
       }
       else if(p2_r_h.size()==2) {
        cp_r_2();
       }
       else if(p2_r_h.size()==3) {
        cp_r_3();
       }
       else if(p2_r_h.size()==4) {
        cp_r_4();
       }


        if(p1_l_h.size()==0) {
```

```
            p_l_5_0();
           }
           else if(p1_l_h.size()==1) {
            p_l_1();
           }
           else if(p1_l_h.size()==2) {
            p_l_2();
           }
           else if(p1_l_h.size()==3) {
            p_l_3();
           }
           else if(p1_l_h.size()==4) {
            p_l_4();
           }


            if(p1_r_h.size()==0) {
                 p_r_5_0();
           }
           else if(p1_r_h.size()==1) {
            p_r_1();
           }
           else if(p1_r_h.size()==2) {
            p_r_2();
           }
           else if(p1_r_h.size()==3) {
            p_r_3();
           }
           else if(p1_r_h.size()==4) {
            p_r_4();
           }
     }
}
```

How_To_Play.java

```
package sahu.chopstick_game;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
```

```
import java.awt.Toolkit;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JScrollBar;
import javax.swing.JScrollPane;
import javax.swing.border.LineBorder;
import javax.swing.border.TitledBorder;

import org.apache.commons.mail.EmailException;

import javax.swing.JButton;
import javax.swing.ScrollPaneConstants;
import java.awt.FlowLayout;
import javax.swing.JTextField;
import javax.swing.BoxLayout;
import java.awt.GridLayout;
import java.awt.CardLayout;
import com.jgoodies.forms.layout.FormLayout;
import com.jgoodies.forms.layout.ColumnSpec;
import com.jgoodies.forms.layout.FormSpecs;
import com.jgoodies.forms.layout.RowSpec;
import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JTextPane;
import javax.swing.JTextArea;
import javax.swing.border.BevelBorder;
import javax.swing.JLabel;
import javax.swing.SwingConstants;
import javax.swing.ImageIcon;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.border.EtchedBorder;
import javax.swing.UIManager;
import java.awt.SystemColor;
import javax.swing.border.MatteBorder;

@SuppressWarnings({ "serial", "unused" })
public class How_To_Play extends JFrame implements
MouseListener,ActionListener{
     private JPanel p1,p2,p3,p4,p5;
```

```
    private JButton n1,n2,n3,n4,
                    back,back2,
                    pr1,pr2,pr3,pr4;

    private JLabel l1,l2,l3,l4,l5,
                   cnt1,cnt2,cnt3,cnt4,cnt5;

    private JTextPane t1,t2,t3,t4,t5;

    public How_To_Play() {
        setResizable(false);
        setForeground(Color.BLACK);

    setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().g
etResource("logo_F.png")));
        setBackground(Color.BLACK);
        setFont(new Font("Dialog", Font.BOLD, 20));
        setTitle("How To Play");
        setBounds(150, 20, 1250, 785);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().setLayout(null);

        p1 = new JPanel();
        p1.setBackground(new Color(255, 204, 255));
        p1.setBorder(new BevelBorder(BevelBorder.RAISED, null,
null, null, null));
        p1.setBounds(0, 0, 1236, 748);
        getContentPane().add(p1);
        p1.setLayout(null);

        n1 = new JButton("Next");
        n1.setBorder(new MatteBorder(3, 3, 3, 3, (Color)
Color.BLACK));
        n1.addMouseListener(this);
        n1.addActionListener(this);
        n1.setFocusable(false);
        n1.setFont(new Font("Dialog", Font.BOLD, 30));
        n1.setForeground(Color.black);
        n1.setBackground(UIManager.getColor("Button.light"));
        n1.setBounds(1098, 694, 128, 49);
        p1.add(n1);

        l1 = new JLabel();
        l1.setBorder(new MatteBorder(2, 2, 2, 2, (Color) new
Color(0, 0, 0)));
```

```
        l1.setIcon(new
ImageIcon(getClass().getResource("rule1.jpg")));
        l1.setHorizontalAlignment(SwingConstants.CENTER);
        l1.setBounds(234, 0, 732, 455);
        p1.add(l1);

        t1 = new JTextPane();
        t1.setBorder(new EtchedBorder(EtchedBorder.RAISED,

UIManager.getColor("Button.focus"),

UIManager.getColor("Button.foreground")));
        t1.setEditable(false);
        t1.setText("This official set of rules is called rollover
where five fingers are "
                + "subtracted should a hand's sum exceeds 5 as
described further.\r\n"
                + "Each player begins with one finger raised on
each hand. "
                + "After the first player turns proceed
clockwise.\r\n"
                + "On a player's turn, they must either attack
or split. "
                + "There are two types of splits, transfers and
divisions.");
        t1.setBackground(UIManager.getColor("Button.highlight"));
        t1.setForeground(SystemColor.textText);
        t1.setFont(new Font("Tahoma", Font.BOLD | Font.ITALIC,
30));
        t1.setBounds(10, 456, 1216, 234);
        p1.add(t1);

        back = new JButton("Back");
        back.setBorder(new MatteBorder(3, 3, 3, 3, (Color)
Color.BLACK));
        back.addMouseListener(this);
        back.addActionListener(this);
        back.setForeground(Color.black);
        back.setFont(new Font("Dialog", Font.BOLD, 30));
        back.setFocusable(false);
        back.setBackground(UIManager.getColor("Button.light"));
        back.setBounds(10, 694, 128, 49);
        p1.add(back);

        cnt1 = new JLabel("1");
```

```
        cnt1.setBorder(new LineBorder(new Color(255, 255, 255),
4, true));
        cnt1.setOpaque(true);
        cnt1.setFont(new Font("Tahoma", Font.BOLD, 30));
        cnt1.setForeground(Color.white);
        cnt1.setBackground(Color.BLACK);
        cnt1.setHorizontalAlignment(SwingConstants.CENTER);
        cnt1.setBounds(556, 692, 98, 54);
        p1.add(cnt1);

        p2 = new JPanel();
        p2.setBorder(new BevelBorder(BevelBorder.RAISED, null,
null, null, null));
        p2.setBackground(new Color(255, 204, 255));
        p2.setBounds(0, 0, 1236, 748);
        getContentPane().add(p2);
        p2.setLayout(null);

        cnt2 = new JLabel("2");
        cnt2.setOpaque(true);
        cnt2.setBorder(new LineBorder(new Color(255, 255, 255),
4, true));
        cnt2.setFont(new Font("Tahoma", Font.BOLD, 30));
        cnt2.setForeground(Color.WHITE);
        cnt2.setBackground(Color.black);
        cnt2.setHorizontalAlignment(SwingConstants.CENTER);
        cnt2.setBounds(556, 692, 98, 54);
        p2.add(cnt2);

        pr1 = new JButton("Previous");
        pr1.setBorder(new MatteBorder(3, 3, 3, 3, (Color)
Color.BLACK));
        pr1.addMouseListener(this);
        pr1.addActionListener(this);
        pr1.setFocusable(false);
        pr1.setFont(new Font("Dialog", Font.BOLD, 30));
        pr1.setForeground(Color.black);
        pr1.setBackground(UIManager.getColor("Button.light"));
        pr1.setBounds(10, 694, 159, 49);
        p2.add(pr1);

        n2 = new JButton("Next");
        n2.setBorder(new MatteBorder(3, 3, 3, 3, (Color)
Color.BLACK));
        n2.addMouseListener(this);
```

```
        n2.addActionListener(this);
        n2.setFocusable(false);
        n2.setFont(new Font("Dialog", Font.BOLD, 30));
        n2.setForeground(Color.black);
        n2.setBackground(UIManager.getColor("Button.light"));
        n2.setBounds(1098, 694, 128, 49);
        p2.add(n2);

        l2 = new JLabel();
        l2.setBorder(new MatteBorder(2, 2, 2, 2, (Color) new
Color(0, 0, 0)));
        l2.setHorizontalAlignment(SwingConstants.CENTER);
        l2.setBounds(234, 0, 732, 472);
        l2.setIcon(new
ImageIcon(getClass().getResource("rule2.jpg")));
        p2.add(l2);

        t2 = new JTextPane();
        t2.setBorder(new EtchedBorder(EtchedBorder.RAISED,
                    UIManager.getColor("Button.focus"),
                    UIManager.getColor("Button.foreground")));
        t2.setFont(new Font("Tahoma", Font.BOLD | Font.ITALIC,
30));
        t2.setText("To attack, a player uses one of their live
hands to strike an opponent's live hand."
                    + " The number of fingers on the opponent's
struck hand will increase by the number of fingers on the hand used
to strike.\r\n"
                    + "You will then take "
                    + "turns going back and forth. On each turn, one
player will use one hand to tap one of their opponent's
hand.\r\n");
        t2.setForeground(SystemColor.textText);
        t2.setBackground(UIManager.getColor("Button.highlight"));
        t2.setEditable(false);
        t2.setBounds(10, 475, 1216, 214);
        p2.add(t2);

        p3 = new JPanel();
        p3.setBorder(new BevelBorder(BevelBorder.RAISED, null,
null, null, null));
        p3.setBackground(new Color(255, 204, 255));
        p3.setBounds(0, 0, 1236, 748);
        getContentPane().add(p3);
        p3.setLayout(null);
```

```
        cnt3 = new JLabel("3");
        cnt3.setOpaque(true);
        cnt3.setBorder(new LineBorder(new Color(255, 255, 255),
4, true));
        cnt3.setFont(new Font("Tahoma", Font.BOLD, 30));
        cnt3.setForeground(Color.WHITE);
        cnt3.setBackground(new Color(0, 0, 0));
        cnt3.setHorizontalAlignment(SwingConstants.CENTER);
        cnt3.setBounds(556, 692, 98, 54);
        p3.add(cnt3);

        pr2 = new JButton("Previous");
        pr2.setBorder(new MatteBorder(3, 3, 3, 3, (Color)
Color.BLACK));
        pr2.addMouseListener(this);
        pr2.addActionListener(this);
        pr2.setFocusable(false);
        pr2.setFont(new Font("Dialog", Font.BOLD, 30));
        pr2.setForeground(Color.black);
        pr2.setBackground(UIManager.getColor("Button.light"));
        pr2.setBounds(10, 694, 149, 49);
        p3.add(pr2);

        n3 = new JButton("Next");
        n3.setBorder(new MatteBorder(3, 3, 3, 3, (Color)
Color.BLACK));
        n3.addMouseListener(this);
        n3.addActionListener(this);
        n3.setFocusable(false);
        n3.setFont(new Font("Dialog", Font.BOLD, 30));
        n3.setForeground(Color.black);
        n3.setBackground(UIManager.getColor("Button.light"));
        n3.setBounds(1098, 694, 128, 49);
        p3.add(n3);

        l3 = new JLabel();
        l3.setBorder(new MatteBorder(2, 2, 2, 2, (Color) new
Color(0, 0, 0)));
        l3.setHorizontalAlignment(SwingConstants.CENTER);
        l3.setBounds(234, 0, 732, 469);
        l3.setIcon(new
ImageIcon(getClass().getResource("rule3.jpg")));
        p3.add(l3);
```

```
        t3 = new JTextPane();
        t3.setBorder(new EtchedBorder(EtchedBorder.RAISED,
                    UIManager.getColor("Button.focus"),
                    UIManager.getColor("Button.foreground")));
        t3.setText("A hand is live if it has at least one finger,
and this is indicated by raising at least one finger. "
                + "\r\nIf any hand of any player reaches exactly
five fingers, then the hand is dead.\r\n"
                + "If you go over 5 you subtract the sum of all
of the numbers by 5\r\n"
                + "Continue playing until one player has lost
both of their hands. \r\n");
        t3.setFont(new Font("Tahoma", Font.BOLD | Font.ITALIC,
30));
        t3.setForeground(SystemColor.textText);
        t3.setBackground(UIManager.getColor("Button.highlight"));
        t3.setEditable(false);
        t3.setBounds(10, 475, 1216, 213);
        p3.add(t3);

        p4 = new JPanel();
        p4.setBorder(new BevelBorder(BevelBorder.RAISED, null,
null, null, null));
        p4.setBackground(new Color(255, 204, 255));
        p4.setBounds(0, 0, 1236, 748);
        getContentPane().add(p4);
        p4.setLayout(null);

        cnt4 = new JLabel("4");
        cnt4.setOpaque(true);
        cnt4.setBorder(new LineBorder(new Color(255, 255, 255),
4, true));
        cnt4.setFont(new Font("Tahoma", Font.BOLD, 30));
        cnt4.setForeground(Color.WHITE);
        cnt4.setBackground(new Color(0, 0, 0));
        cnt4.setHorizontalAlignment(SwingConstants.CENTER);
        cnt4.setBounds(556, 692, 98, 54);
        p4.add(cnt4);

        pr3 = new JButton("Previous");
        pr3.setBorder(new MatteBorder(3, 3, 3, 3, (Color)
Color.BLACK));
        pr3.addMouseListener(this);
        pr3.addActionListener(this);
        pr3.setFocusable(false);
```

```
        pr3.setFont(new Font("Dialog", Font.BOLD, 30));
        pr3.setForeground(Color.black);
        pr3.setBackground(UIManager.getColor("Button.light"));
        pr3.setBounds(10, 694, 149, 49);
        p4.add(pr3);

        n4 = new JButton("Next");
        n4.setBorder(new MatteBorder(3, 3, 3, 3, (Color)
Color.BLACK));
        n4.addMouseListener(this);
        n4.addActionListener(this);
        n4.setFocusable(false);
        n4.setFont(new Font("Dialog", Font.BOLD, 30));
        n4.setForeground(Color.black);
        n4.setBackground(UIManager.getColor("Button.light"));
        n4.setBounds(1098, 694, 128, 49);
        p4.add(n4);

        l4 = new JLabel();
        l4.setBorder(new MatteBorder(2, 2, 2, 2, (Color) new
Color(0, 0, 0)));
        l4.setHorizontalAlignment(SwingConstants.CENTER);
        l4.setBounds(234, 0, 732, 473);
        l4.setIcon(new
ImageIcon(getClass().getResource("rule4.jpg")));
        p4.add(l4);

        t4 = new JTextPane();
        t4.setBorder(new EtchedBorder(EtchedBorder.RAISED,
                    UIManager.getColor("Button.focus"),
                    UIManager.getColor("Button.foreground")));
        t4.setText("If a hand has zero fingers, the hand is dead,
and this is indicated by "
                + "raising zero fingers (i.e. a closed
fist).\r\n"
                + "A player with two dead hands is eliminated
from the game.\r\n"
                + "The goal is to be the last one standing with
at least one hand left still alive.\r\n"
                + "A player wins once all opponents are
eliminated.");
        t4.setFont(new Font("Tahoma", Font.BOLD | Font.ITALIC,
30));
        t4.setForeground(SystemColor.textText);
        t4.setBackground(UIManager.getColor("Button.highlight"));
```

```
        t4.setEditable(false);
        t4.setBounds(10, 475, 1216, 214);
        p4.add(t4);

        p5 = new JPanel();
        p5.setBorder(new BevelBorder(BevelBorder.RAISED, null,
null, null, null));
        p5.setBackground(new Color(255, 204, 255));
        p5.setBounds(0, 0, 1236, 748);
        getContentPane().add(p5);
        p5.setLayout(null);

        cnt5 = new JLabel("5");
        cnt5.setOpaque(true);
        cnt5.setBorder(new LineBorder(new Color(255, 255, 255),
4, true));
        cnt5.setFont(new Font("Tahoma", Font.BOLD, 30));
        cnt5.setForeground(Color.WHITE);
        cnt5.setBackground(new Color(0, 0, 0));
        cnt5.setHorizontalAlignment(SwingConstants.CENTER);
        cnt5.setBounds(556, 692, 98, 54);
        p5.add(cnt5);

        pr4 = new JButton("Previous");
        pr4.setBorder(new MatteBorder(3, 3, 3, 3, (Color)
Color.BLACK));
        pr4.addMouseListener(this);
        pr4.addActionListener(this);
        pr4.setFocusable(false);
        pr4.setFont(new Font("Dialog", Font.BOLD, 30));
        pr4.setForeground(Color.black);
        pr4.setBackground(UIManager.getColor("Button.light"));
        pr4.setBounds(10, 694, 149, 49);
        p5.add(pr4);

        l5 = new JLabel();
        l5.setBorder(new MatteBorder(2, 2, 2, 2, (Color) new
Color(0, 0, 0)));
        l5.setHorizontalAlignment(SwingConstants.CENTER);
        l5.setBounds(234, 0, 732, 414);
        l5.setIcon(new
ImageIcon(getClass().getResource("rule5.jpg")));
        p5.add(l5);

        t5 = new JTextPane();
```

```
            t5.setBorder(new EtchedBorder(EtchedBorder.RAISED,
                        UIManager.getColor("Button.focus"),
                        UIManager.getColor("Button.foreground")));
            t5.setText("To transfer, a player strikes their own two
hands together, "
                    + "and transfers raised fingers from one hand to
the other as desired.\r\n "
                    + "However, a player cannot transfer fingers to
make a hand have more than 4 fingers.\r\n"
                    + "If a player has a dead hand, the player can
divide the fingers between the other hand "
                    + "and the dead hand by transferring fingers
from the other hand to the dead hand.");
            t5.setFont(new Font("Tahoma", Font.BOLD | Font.ITALIC,
30));
            t5.setForeground(SystemColor.textText);
            t5.setBackground(UIManager.getColor("Button.highlight"));
            t5.setEditable(false);
            t5.setBounds(10, 416, 1216, 275);
            p5.add(t5);

            back2 = new JButton("Back");
            back2.setBorder(new MatteBorder(3, 3, 3, 3, (Color)
Color.BLACK));
            back2.addMouseListener(this);
            back2.addActionListener(this);
            back2.setForeground(Color.black);
            back2.setFont(new Font("Dialog", Font.BOLD, 30));
            back2.setFocusable(false);
            back2.setBackground(UIManager.getColor("Button.light"));
            back2.setBounds(1098, 694, 128, 49);
            p5.add(back2);

            p1.setVisible(true);
            p2.setVisible(false);
            p3.setVisible(false);
            p4.setVisible(false);
            p5.setVisible(false);

            setVisible(true);
        }

        @Override
        public void actionPerformed(ActionEvent e) {
            if(e.getSource()==back || e.getSource()==back2) {
```

```
                this.dispose();
                try {
                    new ChopStick().main_frm.setVisible(true);
                } catch (Exception e1) {
                    e1.printStackTrace();
                }
            }

            if(e.getSource()==n1 || e.getSource()==pr2) {
                p1.setVisible(false);
                p2.setVisible(true);
                p3.setVisible(false);
                p4.setVisible(false);
                p5.setVisible(false);
            }
            if(e.getSource()==pr1) {
                p1.setVisible(true);
                p2.setVisible(false);
                p3.setVisible(false);
                p4.setVisible(false);
                p5.setVisible(false);
            }
            if(e.getSource()==n2 || e.getSource()==pr3) {
                p1.setVisible(false);
                p2.setVisible(false);
                p3.setVisible(true);
                p4.setVisible(false);
                p5.setVisible(false);
            }
            if(e.getSource()==n3 || e.getSource()==pr4) {
                p1.setVisible(false);
                p2.setVisible(false);
                p3.setVisible(false);
                p4.setVisible(true);
                p5.setVisible(false);
            }
            if(e.getSource()==n4) {
                p1.setVisible(false);
                p2.setVisible(false);
                p3.setVisible(false);
                p4.setVisible(false);
                p5.setVisible(true);
            }
        }
```

```
    @Override
    public void mouseEntered(MouseEvent e) {

        if(e.getSource()==n1) {
            n1.setBackground(Color.lightGray);
        }
        if(e.getSource()==n2) {
            n2.setBackground(Color.lightGray);
        }
        if(e.getSource()==n3) {
            n3.setBackground(Color.lightGray);
        }
        if(e.getSource()==n4) {
            n4.setBackground(Color.lightGray);
        }

        if(e.getSource()==pr1) {
            pr1.setBackground(Color.lightGray);
        }
        if(e.getSource()==pr2) {
            pr2.setBackground(Color.lightGray);
        }
        if(e.getSource()==pr3) {
            pr3.setBackground(Color.lightGray);
        }
        if(e.getSource()==pr4) {
            pr4.setBackground(Color.lightGray);
        }

        if(e.getSource()==back) {
            back.setBackground(Color.lightGray);
        }
        if(e.getSource()==back2) {
            back2.setBackground(Color.lightGray);
        }
    }

    @Override
    public void mouseExited(MouseEvent e) {

        if(e.getSource()==n1) {
            n1.setBackground(UIManager.getColor("Button.light"));
        }
        if(e.getSource()==n2) {
            n2.setBackground(UIManager.getColor("Button.light"));
```

```
        }
        if(e.getSource()==n3) {
            n3.setBackground(UIManager.getColor("Button.light"));
        }
        if(e.getSource()==n4) {
            n4.setBackground(UIManager.getColor("Button.light"));
        }

        if(e.getSource()==pr1) {

    pr1.setBackground(UIManager.getColor("Button.light"));
        }
        if(e.getSource()==pr2) {

    pr2.setBackground(UIManager.getColor("Button.light"));
        }
        if(e.getSource()==pr3) {

    pr3.setBackground(UIManager.getColor("Button.light"));
        }
        if(e.getSource()==pr4) {

    pr4.setBackground(UIManager.getColor("Button.light"));
        }

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

    back.setBackground(UIManager.getColor("Button.light"));
        }
        if(e.getSource()==back2) {

    back2.setBackground(UIManager.getColor("Button.light"));
        }
    }

    @Override
    public void mouseClicked(MouseEvent e) {

    }

    @Override
    public void mousePressed(MouseEvent e) {

    }
```

```
    @Override
    public void mouseReleased(MouseEvent e) {

    }
}
```

Settings.java

```
package sahu.chopstick_game;

import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.Toolkit;

import javax.swing.JColorChooser;
import javax.swing.JFrame;
import java.awt.Window.Type;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.SwingConstants;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.awt.event.ActionEvent;
import javax.swing.border.BevelBorder;
import javax.swing.border.CompoundBorder;
import javax.swing.border.EtchedBorder;
import javax.swing.border.LineBorder;
import javax.swing.border.MatteBorder;
import javax.swing.border.SoftBevelBorder;
import javax.swing.border.TitledBorder;

import org.apache.commons.mail.EmailException;
import javax.swing.JToggleButton;
import java.awt.event.ItemListener;
```

```
import java.awt.event.ItemEvent;
import javax.swing.UIManager;
import java.awt.SystemColor;

import java.sql.*;

@SuppressWarnings("unused")
public class Settings implements
MouseListener,ActionListener,Runnable{

     private JFrame set_frm;

     private JLabel bg_lbl,
                    pwc_lbl_ch,
                    comc_lbl,
                    pvp_lbl_ch,
                    pvpc_lbl,
                    pwc_lbl_ch_1,
                    pwc_lbl_ch_2,
                    pwc_lbl_ch_3,
                    pwc_lbl_ch_4,
                    pla_pnl_lbl_c,
                    com_pnl_lbl_c,
                    pla1_pnl_lbl_c,
                    pla2_pnl_lbl_c,
                    bg_pnl,
                    bg_pnl_1,
                    bg_pnl_2,
                    audio;

     private JButton    combgch_btn,
                        cpvp_btn,
                        btn_pla_pnl_ch,
                        btn_com_pnl_ch,
                        btn_pnl_pla1_ch,
                        btn_pla2_pnl_ch,
                        back,
                        res_def,
                        deletacc_btn;

     private JToggleButton oo;

     public boolean
bgpwc,bgpvp,bgpnlpla,bgpnlcom,bgpnlpla1,bgpnlpla2,del;
     public Color b1,b2,pl,cr,p1,p2;
```

```
     static final char c1[]={102,0,0},c2[]={30,144,255};

     BufferedWriter w=null;
     BufferedReader r=null;

     public Connection c;
     public Statement st;

     public void deleteac() throws Exception{
          del=true;
          while(del) {
               Thread.sleep(1000);
          }
          String
url="jdbc:mysql://db4free.net:3306/stickchop?useSSL=false";
          String usr="voidmatrix";
          String s="voidmatrix2002";
          try {
               r=new BufferedReader(new FileReader("player.txt"));
               w=new BufferedWriter(new FileWriter("player.txt"));
               if(Void_matrix_account.pag) {
                    w.flush();
                    r.close();
                    w.close();
               }
               else {
                    Class.forName("com.mysql.cj.jdbc.Driver");
                    c= DriverManager.getConnection(url,usr,s);
                    st=c.createStatement();
                    st.executeUpdate("delete from signup where
uname='"+r.readLine()+"'");
                    r.close();
                    w.close();
               }

          } catch (Exception e) {
               e.printStackTrace();
               JOptionPane.showMessageDialog(set_frm,
                         "So Sorry, some errors occurred\nCheck your
Internet Connection\nErr: "+e.getMessage(),
                         "Account wasn't deleted",
JOptionPane.ERROR_MESSAGE);
          }
```

```
    }
    @Override
    public void run() {
        try {
            deleteac();
        }catch(Exception e) {
            e.printStackTrace();
        }
    }

    public Settings() {
        try {
            w=new BufferedWriter(new FileWriter("settings.txt"));
            r=new BufferedReader(new FileReader("settings.txt"));
            initialize();
            new Thread(this).start();
        }catch(Exception e) {
            e.printStackTrace();
        }

    }

    private void initialize() throws Exception{
        set_frm= new JFrame();
        set_frm.getContentPane().setFocusable(false);
        set_frm.getContentPane().setBackground(Color.LIGHT_GRAY);
        set_frm.setResizable(false);
        set_frm.setForeground(Color.BLACK);

    set_frm.setIconImage(Toolkit.getDefaultToolkit().getImage(getC
lass().getResource("logo_F.png")));
        set_frm.setBackground(Color.BLACK);
        set_frm.setFont(new Font("Dialog", Font.BOLD, 20));
        set_frm.setTitle("Settings");
        set_frm.setBounds(150, 20, 1250, 785);
        set_frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        set_frm.getContentPane().setLayout(null);

        bg_lbl = new JLabel("Background Color");
        bg_lbl.setBorder(new BevelBorder(BevelBorder.RAISED,
null, null, null, null));
        bg_lbl.setFont(new Font("Tahoma", Font.BOLD, 35));
        bg_lbl.setForeground(Color.BLACK);
        bg_lbl.setHorizontalAlignment(SwingConstants.CENTER);
```

```
        bg_lbl.setBounds(385, 10, 343, 68);
        set_frm.getContentPane().add(bg_lbl);

        pwc_lbl_ch = new JLabel("While Playing Against Computer
:");
        pwc_lbl_ch.setBorder(new
EtchedBorder(EtchedBorder.RAISED, null, null));
        pwc_lbl_ch.setBackground(null);
        pwc_lbl_ch.setHorizontalAlignment(SwingConstants.CENTER);
        pwc_lbl_ch.setForeground(Color.BLACK);
        pwc_lbl_ch.setFont(new Font("Tahoma", Font.BOLD, 25));
        pwc_lbl_ch.setBounds(126, 90, 478, 41);
        set_frm.getContentPane().add(pwc_lbl_ch);

        pvp_lbl_ch = new JLabel("While Playing Against Player
:");
        pvp_lbl_ch.setBorder(new
EtchedBorder(EtchedBorder.RAISED, null, null));
        pvp_lbl_ch.setHorizontalAlignment(SwingConstants.CENTER);
        pvp_lbl_ch.setForeground(Color.BLACK);
        pvp_lbl_ch.setFont(new Font("Tahoma", Font.BOLD, 25));
        pvp_lbl_ch.setBackground((Color) null);
        pvp_lbl_ch.setBounds(126, 174, 478, 41);
        set_frm.getContentPane().add(pvp_lbl_ch);

        bg_pnl = new JLabel("Panel Background Color");
        bg_pnl.setHorizontalAlignment(SwingConstants.CENTER);
        bg_pnl.setForeground(Color.BLACK);
        bg_pnl.setFont(new Font("Tahoma", Font.BOLD, 35));
        bg_pnl.setBorder(new BevelBorder(BevelBorder.RAISED,
null, null, null, null));
        bg_pnl.setBounds(351, 266, 445, 68);
        set_frm.getContentPane().add(bg_pnl);

        bg_pnl_1 = new JLabel("While Playing Against Computer");

    bg_pnl_1.setHorizontalTextPosition(SwingConstants.CENTER);
        bg_pnl_1.setHorizontalAlignment(SwingConstants.CENTER);
        bg_pnl_1.setForeground(Color.BLACK);
        bg_pnl_1.setFont(new Font("Tahoma", Font.BOLD, 30));
        bg_pnl_1.setBorder(new
SoftBevelBorder(BevelBorder.LOWERED, null, null, null, null));
        bg_pnl_1.setBounds(52, 364, 496, 48);
        set_frm.getContentPane().add(bg_pnl_1);
```

```
        bg_pnl_2 = new JLabel("While Playing Against Player");

    bg_pnl_2.setHorizontalTextPosition(SwingConstants.CENTER);
        bg_pnl_2.setHorizontalAlignment(SwingConstants.CENTER);
        bg_pnl_2.setForeground(Color.BLACK);
        bg_pnl_2.setFont(new Font("Tahoma", Font.BOLD, 30));
        bg_pnl_2.setBorder(new
SoftBevelBorder(BevelBorder.LOWERED, null, null, null, null));
        bg_pnl_2.setBounds(666, 364, 470, 48);
        set_frm.getContentPane().add(bg_pnl_2);

        pwc_lbl_ch_1 = new JLabel("Player Panel  :");

    pwc_lbl_ch_1.setHorizontalAlignment(SwingConstants.CENTER);
        pwc_lbl_ch_1.setForeground(Color.BLACK);
        pwc_lbl_ch_1.setFont(new Font("Tahoma", Font.BOLD, 25));
        pwc_lbl_ch_1.setBorder(new
EtchedBorder(EtchedBorder.RAISED, null, null));
        pwc_lbl_ch_1.setBackground((Color) null);
        pwc_lbl_ch_1.setBounds(52, 433, 226, 41);
        set_frm.getContentPane().add(pwc_lbl_ch_1);

        pwc_lbl_ch_2 = new JLabel("Computer Panel  :");

    pwc_lbl_ch_2.setHorizontalAlignment(SwingConstants.CENTER);
        pwc_lbl_ch_2.setForeground(Color.BLACK);
        pwc_lbl_ch_2.setFont(new Font("Tahoma", Font.BOLD, 25));
        pwc_lbl_ch_2.setBorder(new
EtchedBorder(EtchedBorder.RAISED, null, null));
        pwc_lbl_ch_2.setBackground((Color) null);
        pwc_lbl_ch_2.setBounds(52, 500, 226, 41);
        set_frm.getContentPane().add(pwc_lbl_ch_2);

        pwc_lbl_ch_3 = new JLabel("Player1 Panel  :");

    pwc_lbl_ch_3.setHorizontalAlignment(SwingConstants.CENTER);
        pwc_lbl_ch_3.setForeground(Color.BLACK);
        pwc_lbl_ch_3.setFont(new Font("Tahoma", Font.BOLD, 25));
        pwc_lbl_ch_3.setBorder(new
EtchedBorder(EtchedBorder.RAISED, null, null));
        pwc_lbl_ch_3.setBackground((Color) null);
        pwc_lbl_ch_3.setBounds(666, 433, 219, 41);
        set_frm.getContentPane().add(pwc_lbl_ch_3);

        pwc_lbl_ch_4 = new JLabel("Player2 Panel  :");
```

```
        pwc_lbl_ch_4.setHorizontalAlignment(SwingConstants.CENTER);
        pwc_lbl_ch_4.setForeground(Color.BLACK);
        pwc_lbl_ch_4.setFont(new Font("Tahoma", Font.BOLD, 25));
        pwc_lbl_ch_4.setBorder(new
EtchedBorder(EtchedBorder.RAISED, null, null));
        pwc_lbl_ch_4.setBackground((Color) null);
        pwc_lbl_ch_4.setBounds(666, 500, 219, 41);
        set_frm.getContentPane().add(pwc_lbl_ch_4);

        comc_lbl = new JLabel();
        comc_lbl.setBorder(new MatteBorder(2, 2, 2, 2, (Color)
new Color(0, 0, 0)));
        comc_lbl.setOpaque(true);
        comc_lbl.setBounds(821, 83, 78, 48);
        set_frm.getContentPane().add(comc_lbl);

        pvpc_lbl = new JLabel();
        pvpc_lbl.setBorder(new MatteBorder(2, 2, 2, 2, (Color)
new Color(0, 0, 0)));
        pvpc_lbl.setOpaque(true);
        pvpc_lbl.setBounds(821, 166, 78, 49);
        set_frm.getContentPane().add(pvpc_lbl);

        pla_pnl_lbl_c = new JLabel();
        pla_pnl_lbl_c.setOpaque(true);
        pla_pnl_lbl_c.setBorder(new MatteBorder(2, 2, 2, 2,
(Color) new Color(0, 0, 0)));
        pla_pnl_lbl_c.setBounds(480, 433, 44, 41);
        set_frm.getContentPane().add(pla_pnl_lbl_c);

        com_pnl_lbl_c = new JLabel();
        com_pnl_lbl_c.setOpaque(true);
        com_pnl_lbl_c.setBorder(new MatteBorder(2, 2, 2, 2,
(Color) new Color(0, 0, 0)));
        com_pnl_lbl_c.setBounds(480, 500, 44, 41);
        set_frm.getContentPane().add(com_pnl_lbl_c);

        pla1_pnl_lbl_c = new JLabel();
        pla1_pnl_lbl_c.setOpaque(true);
        pla1_pnl_lbl_c.setBorder(new MatteBorder(2, 2, 2, 2,
(Color) new Color(0, 0, 0)));
        pla1_pnl_lbl_c.setBounds(1071, 433, 44, 41);
        set_frm.getContentPane().add(pla1_pnl_lbl_c);
```

```
          pla2_pnl_lbl_c = new JLabel();
          pla2_pnl_lbl_c.setOpaque(true);
          pla2_pnl_lbl_c.setBorder(new MatteBorder(2, 2, 2, 2,
(Color) new Color(0, 0, 0)));
          pla2_pnl_lbl_c.setBounds(1071, 500, 44, 41);
          set_frm.getContentPane().add(pla2_pnl_lbl_c);

          if(!r.ready()) {
               w.write(c1); w.write(c1);
               for(int i=0;i<4;i++) {
                    w.write(c2);
               }
               w.close();
               r.close();
               comc_lbl.setBackground(new Color(102, 0, 0));
               pvpc_lbl.setBackground(new Color(102, 0, 0));
               pla_pnl_lbl_c.setBackground(new Color(30,144,255));
               com_pnl_lbl_c.setBackground(new Color(30,144,255));
               pla1_pnl_lbl_c.setBackground(new Color(30,144,255));
               pla2_pnl_lbl_c.setBackground(new Color(30,144,255));
          }
          else {
               w.close();
               comc_lbl.setBackground(new Color(r.read(), r.read(),
r.read()));
               pvpc_lbl.setBackground(new Color(r.read(), r.read(),
r.read()));
               pla_pnl_lbl_c.setBackground(new
Color(r.read(),r.read(),r.read()));
               com_pnl_lbl_c.setBackground(new
Color(r.read(),r.read(),r.read()));
               pla1_pnl_lbl_c.setBackground(new
Color(r.read(),r.read(),r.read()));
               pla2_pnl_lbl_c.setBackground(new
Color(r.read(),r.read(),r.read()));
               r.close();
          }

          res_def = new JButton("Restore Defaults");
          res_def.setBorder(new BevelBorder(BevelBorder.RAISED,
null, null, null, null));
          res_def.addActionListener(this);
          res_def.addMouseListener(this);
          res_def.setForeground(new Color(0, 0, 51));
          res_def.setFont(new Font("Tahoma", Font.BOLD, 30));
```

```
        res_def.setFocusable(false);

    res_def.setBackground(UIManager.getColor("Button.light"));
        res_def.setBounds(433, 690, 296, 48);
        set_frm.getContentPane().add(res_def);

        audio = new JLabel("Sound  :");
        audio.setHorizontalAlignment(SwingConstants.CENTER);
        audio.setForeground(Color.BLACK);
        audio.setFont(new Font("Tahoma", Font.BOLD, 35));
        audio.setBorder(new BevelBorder(BevelBorder.RAISED, null,
null, null, null));
        audio.setBounds(408, 591, 211, 68);
        set_frm.getContentPane().add(audio);

        combgch_btn = new JButton("Choose");
        combgch_btn.setBorder(new LineBorder(new Color(0, 0, 0),
3, true));
        combgch_btn.addActionListener(this);
        combgch_btn.addMouseListener(this);
        combgch_btn.setFocusable(false);
        combgch_btn.setForeground(new Color(0, 0, 51));

    combgch_btn.setBackground(UIManager.getColor("Button.light"));
        combgch_btn.setFont(new Font("Tahoma", Font.BOLD, 30));
        combgch_btn.setBounds(633, 88, 146, 41);
        set_frm.getContentPane().add(combgch_btn);

        cpvp_btn = new JButton("Choose");
        cpvp_btn.addActionListener(this);
        cpvp_btn.addMouseListener(this);
        cpvp_btn.setBorder(new LineBorder(new Color(0, 0, 0), 3,
true));
        cpvp_btn.setForeground(new Color(0, 0, 51));
        cpvp_btn.setFont(new Font("Tahoma", Font.BOLD, 30));
        cpvp_btn.setFocusable(false);

    cpvp_btn.setBackground(UIManager.getColor("Button.light"));
        cpvp_btn.setBounds(633, 172, 146, 41);
        set_frm.getContentPane().add(cpvp_btn);

        btn_pla_pnl_ch = new JButton("Choose");
        btn_pla_pnl_ch.addActionListener(this);
        btn_pla_pnl_ch.addMouseListener(this);
        btn_pla_pnl_ch.setForeground(new Color(0, 0, 51));
```

```
        btn_pla_pnl_ch.setFont(new Font("Tahoma", Font.BOLD,
30));
        btn_pla_pnl_ch.setFocusable(false);
        btn_pla_pnl_ch.setBorder(new LineBorder(new Color(0, 0,
0), 3, true));

    btn_pla_pnl_ch.setBackground(UIManager.getColor("Button.light"
));
        btn_pla_pnl_ch.setBounds(304, 431, 137, 41);
        set_frm.getContentPane().add(btn_pla_pnl_ch);

        btn_com_pnl_ch = new JButton("Choose");
        btn_com_pnl_ch.addActionListener(this);
        btn_com_pnl_ch.addMouseListener(this);
        btn_com_pnl_ch.setForeground(new Color(0, 0, 51));
        btn_com_pnl_ch.setFont(new Font("Tahoma", Font.BOLD,
30));
        btn_com_pnl_ch.setFocusable(false);
        btn_com_pnl_ch.setBorder(new LineBorder(new Color(0, 0,
0), 3, true));

    btn_com_pnl_ch.setBackground(UIManager.getColor("Button.light"
));
        btn_com_pnl_ch.setBounds(304, 500, 137, 41);
        set_frm.getContentPane().add(btn_com_pnl_ch);

        btn_pnl_pla1_ch = new JButton("Choose");
        btn_pnl_pla1_ch.addActionListener(this);
        btn_pnl_pla1_ch.addMouseListener(this);
        btn_pnl_pla1_ch.setForeground(new Color(0, 0, 51));
        btn_pnl_pla1_ch.setFont(new Font("Tahoma", Font.BOLD,
30));
        btn_pnl_pla1_ch.setFocusable(false);
        btn_pnl_pla1_ch.setBorder(new LineBorder(new Color(0, 0,
0), 3, true));

    btn_pnl_pla1_ch.setBackground(UIManager.getColor("Button.light
"));
        btn_pnl_pla1_ch.setBounds(907, 433, 137, 41);
        set_frm.getContentPane().add(btn_pnl_pla1_ch);

        btn_pla2_pnl_ch = new JButton("Choose");
        btn_pla2_pnl_ch.addActionListener(this);
        btn_pla2_pnl_ch.addMouseListener(this);
        btn_pla2_pnl_ch.setForeground(new Color(0, 0, 51));
```

```
        btn_pla2_pnl_ch.setFont(new Font("Tahoma", Font.BOLD,
30));
        btn_pla2_pnl_ch.setFocusable(false);
        btn_pla2_pnl_ch.setBorder(new LineBorder(new Color(0, 0,
0), 3, true));

    btn_pla2_pnl_ch.setBackground(UIManager.getColor("Button.light
"));
        btn_pla2_pnl_ch.setBounds(907, 498, 137, 41);
        set_frm.getContentPane().add(btn_pla2_pnl_ch);

        back = new JButton("Back");
        back.setBorder(new BevelBorder(BevelBorder.RAISED, null,
null, null, null));
        back.addActionListener(this);
        back.addMouseListener(this);
        back.setFocusable(false);
        back.setFont(new Font("Tahoma", Font.BOLD, 30));
        back.setForeground(new Color(0, 0, 51));
        back.setBackground(UIManager.getColor("Button.light"));
        back.setBounds(10, 690, 137, 48);
        set_frm.getContentPane().add(back);

        oo = new JToggleButton("On");
        oo.setBackground(UIManager.getColor("Button.light"));
        oo.setForeground(Color.black);
        oo.setBorder(new MatteBorder(3, 3, 3, 3, (Color) new
Color(0, 0, 0)));
        oo.setSelected(true);
        oo.setFocusable(false);
        oo.addMouseListener(this);
        oo.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent e) {
                if(oo.isSelected()) {
                    oo.setText("On");
                    ChopStick.c.start();
                    ChopStick.c.loop(100);

                }
                else {
                    oo.setText("Off");
                    ChopStick.c.stop();
                }
            }
        });
```

```
            oo.setFont(new Font("Tahoma", Font.BOLD, 25));
            oo.setBounds(629, 591, 115, 68);
            set_frm.getContentPane().add(oo);

            deletacc_btn = new JButton("Delete This Account");
            deletacc_btn.setForeground(new Color(0, 0, 51));
            deletacc_btn.setFont(new Font("Tahoma", Font.BOLD, 30));
            deletacc_btn.setFocusable(false);
            deletacc_btn.setBorder(new
BevelBorder(BevelBorder.RAISED, null, null, null, null));
            deletacc_btn.setBackground(SystemColor.controlHighlight);
            deletacc_btn.setBounds(907, 690, 319, 48);
            set_frm.getContentPane().add(deletacc_btn);

            set_frm.setVisible(true);
        }

        @Override
        public void actionPerformed(ActionEvent e) {

            if(e.getSource()==combgch_btn) {
                bgpwc=true;
                Color c=JColorChooser.showDialog(null, "Pick A
Color", new Color(102,0,0));
                comc_lbl.setBackground(c);
            }

            if(e.getSource()==cpvp_btn) {
                bgpvp=true;
                Color c=JColorChooser.showDialog(null, "Pick A
Color", new Color(102,0,0));
                pvpc_lbl.setBackground(c);
            }

            if(e.getSource()==btn_pla_pnl_ch) {
                bgpnlpla=true;
                Color c=JColorChooser.showDialog(null, "Pick A
Color", new Color(30,144,255));
                pla_pnl_lbl_c.setBackground(c);
            }

            if(e.getSource()==btn_com_pnl_ch) {
                bgpnlcom=true;
                Color c=JColorChooser.showDialog(null, "Pick A
Color", new Color(30,144,255));
```

```
            com_pnl_lbl_c.setBackground(c);
        }

        if(e.getSource()==btn_pnl_pla1_ch) {
            bgpnlpla1=true;
            Color c=JColorChooser.showDialog(null, "Pick A
Color", new Color(30,144,255));
            pla1_pnl_lbl_c.setBackground(c);
        }

        if(e.getSource()==btn_pla2_pnl_ch) {
            bgpnlpla2=true;
            Color c=JColorChooser.showDialog(null, "Pick A
Color", new Color(30,144,255));
            pla2_pnl_lbl_c.setBackground(c);
        }

        if(e.getSource()==back) {
            set_frm.dispose();
            try {
                new ChopStick().main_frm.setVisible(true);
            } catch (Exception e1) {
                e1.printStackTrace();
            }
        }

        if(e.getSource()==res_def) {
            bgpwc=false;
            bgpvp=false;
            bgpnlpla=false;
            bgpnlcom=false;
            bgpnlpla1=false;
            bgpnlpla2=false;
            comc_lbl.setBackground(new Color(102, 0, 0));
            pvpc_lbl.setBackground(new Color(102, 0, 0));
            pla_pnl_lbl_c.setBackground(new Color(30,144,255));
            com_pnl_lbl_c.setBackground(new Color(30,144,255));
            pla1_pnl_lbl_c.setBackground(new Color(30,144,255));
            pla2_pnl_lbl_c.setBackground(new Color(30,144,255));
        }

        try{
            w=new BufferedWriter(new FileWriter("settings.txt"));
            char C1[]={(char)comc_lbl.getBackground().getRed(),
```

```
(char)comc_lbl.getBackground().getGreen(),

(char)comc_lbl.getBackground().getBlue()};
                char C2[]={(char)pvpc_lbl.getBackground().getRed(),

(char)pvpc_lbl.getBackground().getGreen(),

(char)pvpc_lbl.getBackground().getBlue()};
                char
C3[]={(char)pla_pnl_lbl_c.getBackground().getRed(),

(char)pla_pnl_lbl_c.getBackground().getGreen(),

(char)pla_pnl_lbl_c.getBackground().getBlue()};
                char
C4[]={(char)com_pnl_lbl_c.getBackground().getRed(),

(char)com_pnl_lbl_c.getBackground().getGreen(),

(char)com_pnl_lbl_c.getBackground().getBlue()};
                char
C5[]={(char)pla1_pnl_lbl_c.getBackground().getRed(),

(char)pla1_pnl_lbl_c.getBackground().getGreen(),

(char)pla1_pnl_lbl_c.getBackground().getBlue()};
                char
C6[]={(char)pla2_pnl_lbl_c.getBackground().getRed(),

(char)pla2_pnl_lbl_c.getBackground().getGreen(),

(char)pla2_pnl_lbl_c.getBackground().getBlue()};
                w.write(C1);
                w.write(C2);
                w.write(C3);
                w.write(C4);
                w.write(C5);
                w.write(C6);
                w.close();
            }catch(Exception e1) {
                e1.printStackTrace();
            }

            if(e.getSource()==deletacc_btn) {
```

```
                del=false;
                JOptionPane.showConfirmDialog(set_frm, "Deleted
Successfully");
                set_frm.dispose();
                new Void_matrix_account();
           }
      }

      @Override
      public void mouseEntered(MouseEvent e) {

           if(e.getSource()==combgch_btn) {
                combgch_btn.setBackground(Color.black);
                combgch_btn.setForeground(Color.white);
           }

           if(e.getSource()==cpvp_btn) {
                cpvp_btn.setBackground(Color.black);
                cpvp_btn.setForeground(Color.white);
           }

           if(e.getSource()==btn_pla_pnl_ch) {
                btn_pla_pnl_ch.setBackground(Color.black);
                btn_pla_pnl_ch.setForeground(Color.white);
           }

           if(e.getSource()==btn_com_pnl_ch) {
                btn_com_pnl_ch.setBackground(Color.black);
                btn_com_pnl_ch.setForeground(Color.white);
           }

           if(e.getSource()==btn_pnl_pla1_ch) {
                btn_pnl_pla1_ch.setBackground(Color.black);
                btn_pnl_pla1_ch.setForeground(Color.white);
           }

           if(e.getSource()==btn_pla2_pnl_ch) {
                btn_pla2_pnl_ch.setBackground(Color.black);
                btn_pla2_pnl_ch.setForeground(Color.white);
           }

           if(e.getSource()==back) {
                back.setBackground(Color.black);
                back.setForeground(Color.white);
           }
```

```
            if(e.getSource()==res_def) {
                res_def.setBackground(Color.black);
                res_def.setForeground(Color.white);
            }

            if(e.getSource()==oo) {
                oo.setBackground(Color.black);
                oo.setForeground(Color.white);
            }

            if(e.getSource()==deletacc_btn) {
                deletacc_btn.setBackground(Color.black);
                deletacc_btn.setForeground(Color.white);
            }

        }

        @Override
        public void mouseExited(MouseEvent e) {

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

        combgch_btn.setBackground(UIManager.getColor("Button.light"));
                combgch_btn.setForeground(Color.black);
            }

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

        cpvp_btn.setBackground(UIManager.getColor("Button.light"));
                cpvp_btn.setForeground(Color.black);
            }

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

        btn_pla_pnl_ch.setBackground(UIManager.getColor("Button.light"
));
                btn_pla_pnl_ch.setForeground(Color.black);
            }

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

        btn_com_pnl_ch.setBackground(UIManager.getColor("Button.light"
));
                btn_com_pnl_ch.setForeground(Color.black);
```

```
            }

            if(e.getSource()==btn_pnl_pla1_ch) {

        btn_pnl_pla1_ch.setBackground(UIManager.getColor("Button.light
"));
                    btn_pnl_pla1_ch.setForeground(Color.black);
            }

            if(e.getSource()==btn_pla2_pnl_ch) {

        btn_pla2_pnl_ch.setBackground(UIManager.getColor("Button.light
"));
                    btn_pla2_pnl_ch.setForeground(Color.black);
            }

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

        back.setBackground(UIManager.getColor("Button.light"));
                    back.setForeground(Color.black);
            }

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

        res_def.setBackground(UIManager.getColor("Button.light"));
                    res_def.setForeground(Color.black);
            }

            if(e.getSource()==oo) {
                    oo.setBackground(UIManager.getColor("Button.light"));
                    oo.setForeground(Color.black);
            }

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

        deletacc_btn.setBackground(UIManager.getColor("Button.light"))
;
                    deletacc_btn.setForeground(Color.black);
            }

        }

        @Override
        public void mouseClicked(MouseEvent e) {
```

```
    }

    @Override
    public void mousePressed(MouseEvent e) {

    }

    @Override
    public void mouseReleased(MouseEvent e) {

    }
}
```

About_Game.java

```
package sahu.chopstick_game;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.Toolkit;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

import org.apache.commons.mail.EmailException;

import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JTextPane;
import javax.swing.UIManager;
import javax.swing.border.MatteBorder;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.border.EtchedBorder;

@SuppressWarnings({ "serial", "unused" })
public class About_Game extends JFrame implements ActionListener {
    private JButton back;
```

```
    public About_Game() {
        getContentPane().setForeground(Color.ORANGE);
        getContentPane().setBackground(Color.BLACK);
        setResizable(false);
        setForeground(Color.BLACK);

    setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().g
etResource("logo_F.png")));
        setBackground(Color.BLACK);
        setFont(new Font("Dialog", Font.BOLD, 20));
        setTitle("About Game");
        setBounds(150, 20, 1250, 785);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().setLayout(null);

        back = new JButton("back");
        back.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseEntered(MouseEvent e) {
                back.setBackground(Color.black);
                back.setForeground(Color.yellow);
            }
            @Override
            public void mouseExited(MouseEvent e) {

    back.setBackground(UIManager.getColor("Button.light"));
                back.setForeground(Color.black);
            }
        });
        back.setBorder(new MatteBorder(3, 3, 3, 3, (Color)
Color.BLACK));
        back.setFocusable(false);
        back.setForeground(Color.black);
        back.setBackground(UIManager.getColor("Button.light"));
        back.addActionListener(this);
        back.setFont(new Font("Tahoma", Font.BOLD, 30));
        back.setBounds(10, 692, 138, 52);
        getContentPane().add(back);

        JTextPane txtpnChopstickshandGame = new JTextPane();
        txtpnChopstickshandGame.setSelectionColor(new Color(0, 0,
0));
        txtpnChopstickshandGame.setBorder(new
EtchedBorder(EtchedBorder.RAISED, Color.BLACK, Color.YELLOW));
```

```
        txtpnChopstickshandGame.setFont(new Font("Tahoma",
Font.BOLD, 30));
        txtpnChopstickshandGame.setText("\t\t\t\t\t Chopsticks
(hand game)\r\n\tChopsticks is a hand game for two or more players,
in which players extend a number of fingers from each hand and
transfer those scores by taking turns to tap one hand against
another.\r\n \r\n\tChopsticks is an example of a \"combinatorial
game\", and is solved in the sense that with perfect play, an
optimal strategy from any point is known. Chopsticks is a game of
strategy as well as basic math.\r\n\r\n\tIt has roots in Japan and
can also called Finger Chess, Swords, Split, Magic Fingers, Chinese
Fingers, Cherries, Sticks, and Twiddly Dinks. Though there are many
variations of rules and different names, the overall theory and
spirit of the game remains the same.\r\n\r\n\t-> About This
Game(Version :- Plutonium-0.3SNL)\r\nDeveloped By :- Sahil N
Lalani\t\tStudio :- Void_Matrix[:]\r\nDeveloped in :-
JAVA\t\t\t\tPlatform :- PC, Laptop\r\ngenre :- Combinatorial
Game\t\tReleased On :- July 11, 2022");
        txtpnChopstickshandGame.setEditable(false);
        txtpnChopstickshandGame.setForeground(Color.YELLOW);
        txtpnChopstickshandGame.setBackground(Color.BLACK);
        txtpnChopstickshandGame.setBounds(10, 10, 1216, 678);
        getContentPane().add(txtpnChopstickshandGame);

        this.setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        this.dispose();
        try {
            new ChopStick().main_frm.setVisible(true);
        } catch (Exception e1) {
            e1.printStackTrace();
        }

    }
}
```

Mails.java

```
package sahu.chopstick_game;
```

```
import java.awt.Color;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.util.Properties;

import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;
import javax.swing.JFrame;

import org.apache.commons.mail.DefaultAuthenticator;
import org.apache.commons.mail.Email;
import org.apache.commons.mail.EmailException;
import org.apache.commons.mail.SimpleEmail;

@SuppressWarnings({ "deprecation", "unused" })
public class Mails extends SimpleEmail{
     public static Email sendemail;

     public static void sahil() throws EmailException{
          sendemail = new SimpleEmail();
        sendemail.setSmtpPort(587);
        sendemail.setAuthenticator(new
DefaultAuthenticator("voidmatrix00@gmail.com",

     "ozdpqmwvaptdifez"));//password and email
        sendemail.setDebug(false);
        sendemail.setHostName("smtp.gmail.com");
        sendemail.setFrom("voidmatrix00@gmail.com");
     }

     public static void Activity(String x) throws EmailException{
          sahil();
          sendemail.setSubject("activity");
        sendemail.setMsg("game is being played by "+x);
        sendemail.addTo("sahillalani1511@gmail.com");
        sendemail.setTLS(true);
```

```
        sendemail.send(); //sending email
    }

    public static void getFeedBack(String msg,String name) throws
EmailException {
         sahil();
        sendemail.setSubject("Feedback");
        sendemail.setMsg(name+" : "+msg);
        sendemail.addTo("sahillalani1511@gmail.com");
        sendemail.setTLS(true);
        sendemail.send(); //sending email
    }

    public static void sendgreet(String g,String unm,String pwd)
throws EmailException {
         sahil();
        sendemail.setSubject("VoidMatrix Account Sign Up");
        sendemail.setMsg("Thank You For Signing Up With
VoidMatrix\nUsername: "+unm+"\nPassword: "+pwd); //Your Email
Address
        sendemail.addTo(g); // Receiver Email Address
        sendemail.setTLS(true);
        sendemail.send(); //sending email
    }

}
```

Section 7: Testing of "ChopStick" game

This chapter includes some test cases for the game to check if the game works property in various situations. We are giving four test examples for four different situations here.

Database verification test case:

Sr.no	Test Case Id	Test Case Name	Test Case Description	Step	Executed Result	Actual Result	Test Case Status
1	Log in	Validate Login	To verify that login id on registration on Database	Enter login id or password and click on login button	Login Successfully or an error message	Successfully	Pass
2	Sign up	Information of the user	Successfully	Pass			
3	Play as guest	Player name stored locally	successful	pass			

Test case for game: [each test led to an improvement]

1. Animation of hands works perfectly (Tested 23-24 times).
2. Studio label animation works well (tested 5 times).
3. Launch label's progress bar animation works correctly (tested 10 times)

Section 8: User manual of “ChopStick” game with Snapshots

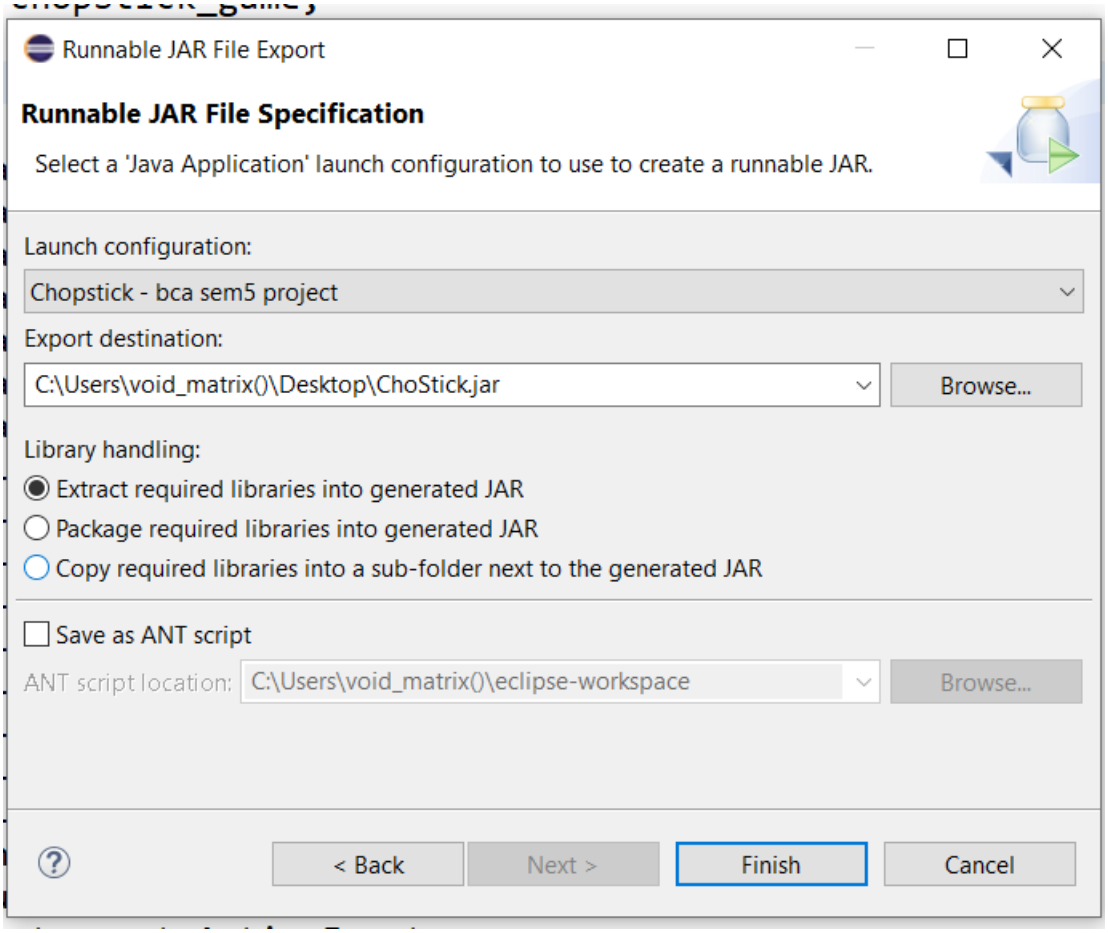
This chapter provides a user instruction for the players. It includes the procedure of playing and also contains some snapshots to give some ideas of the game to the player before starting playing it.

Playing Procedure

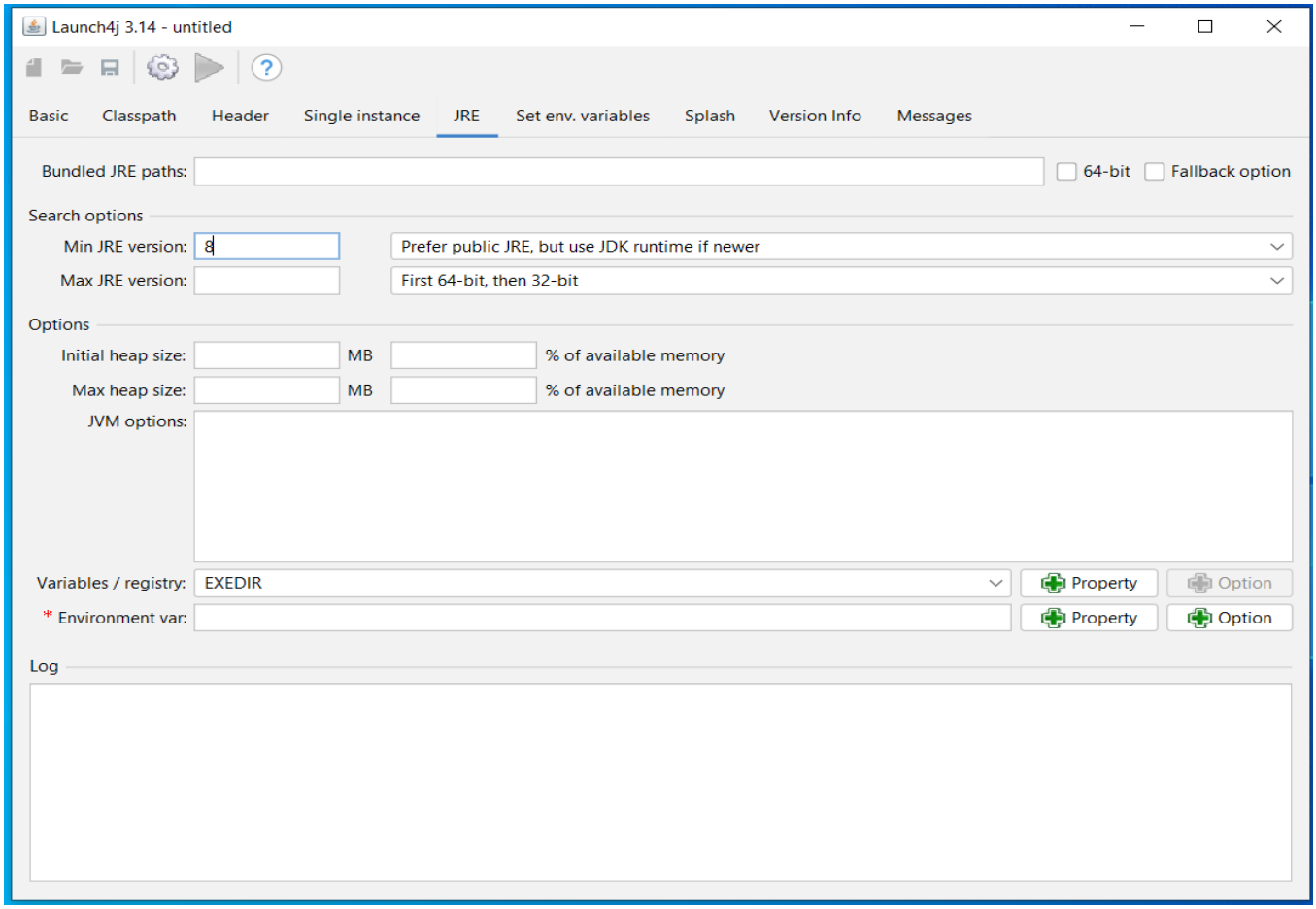
Gamer first interacts with system Ul to start playing. I provide playing tips to all users so that they can easily understand about the playing procedure.

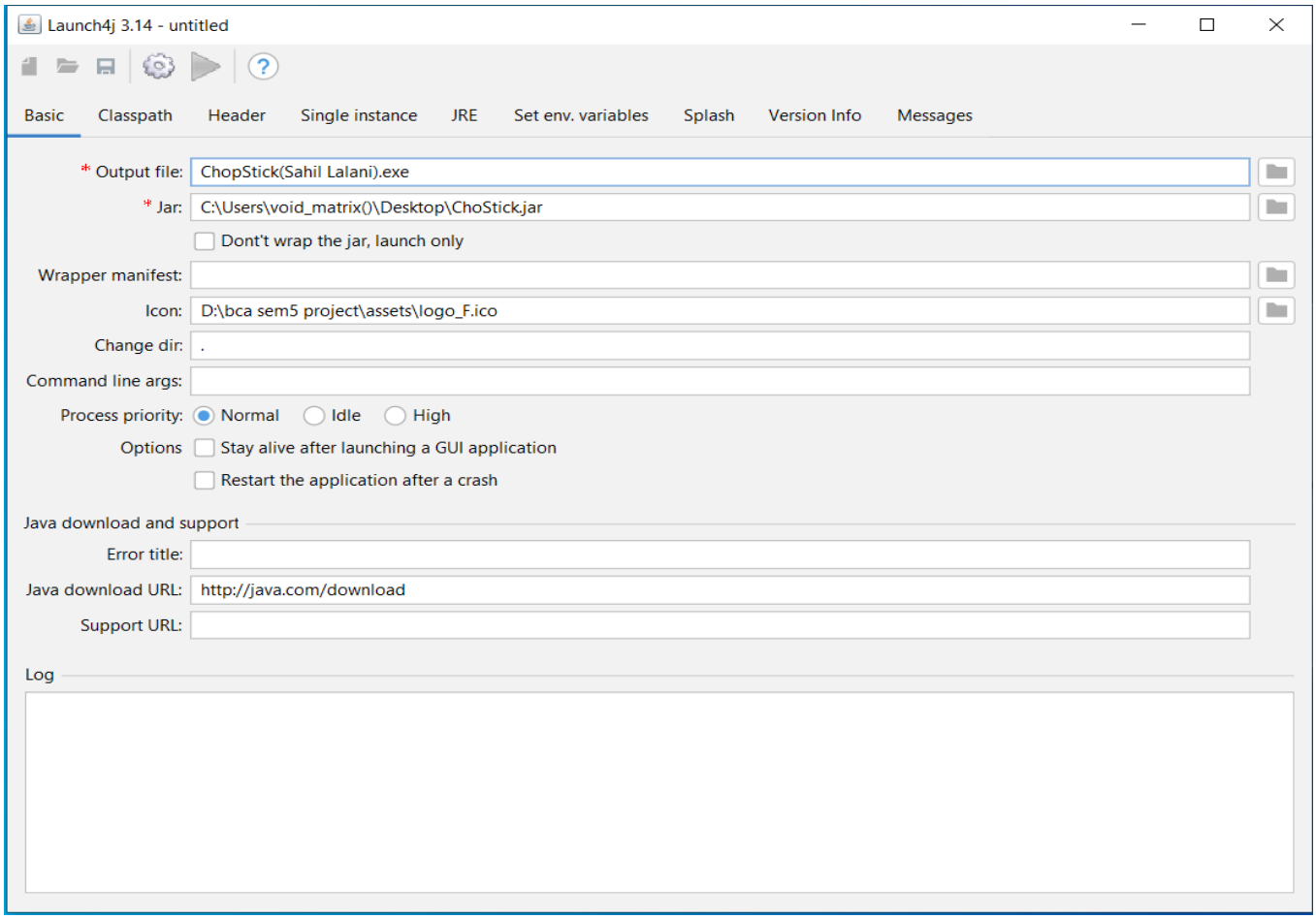
First, I am creating executable jar file using eclipse

Go to the project -> click on it -> click on export and select executable jar file.

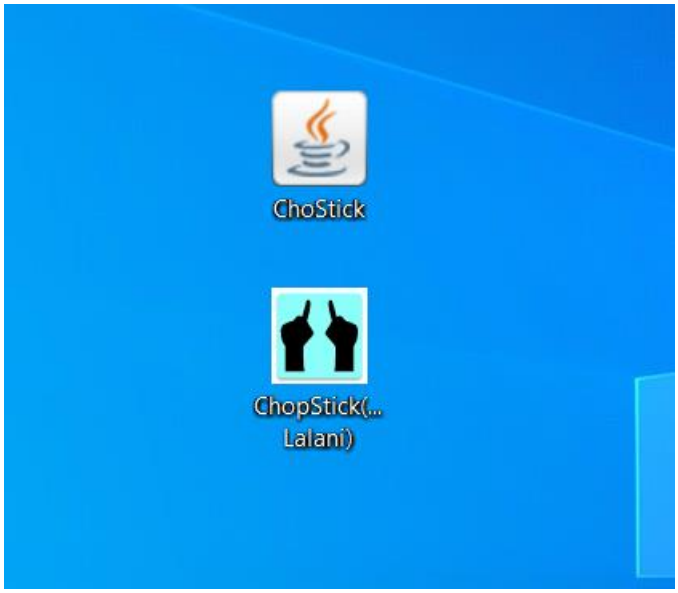


Now I am creating exe file using Launch4j

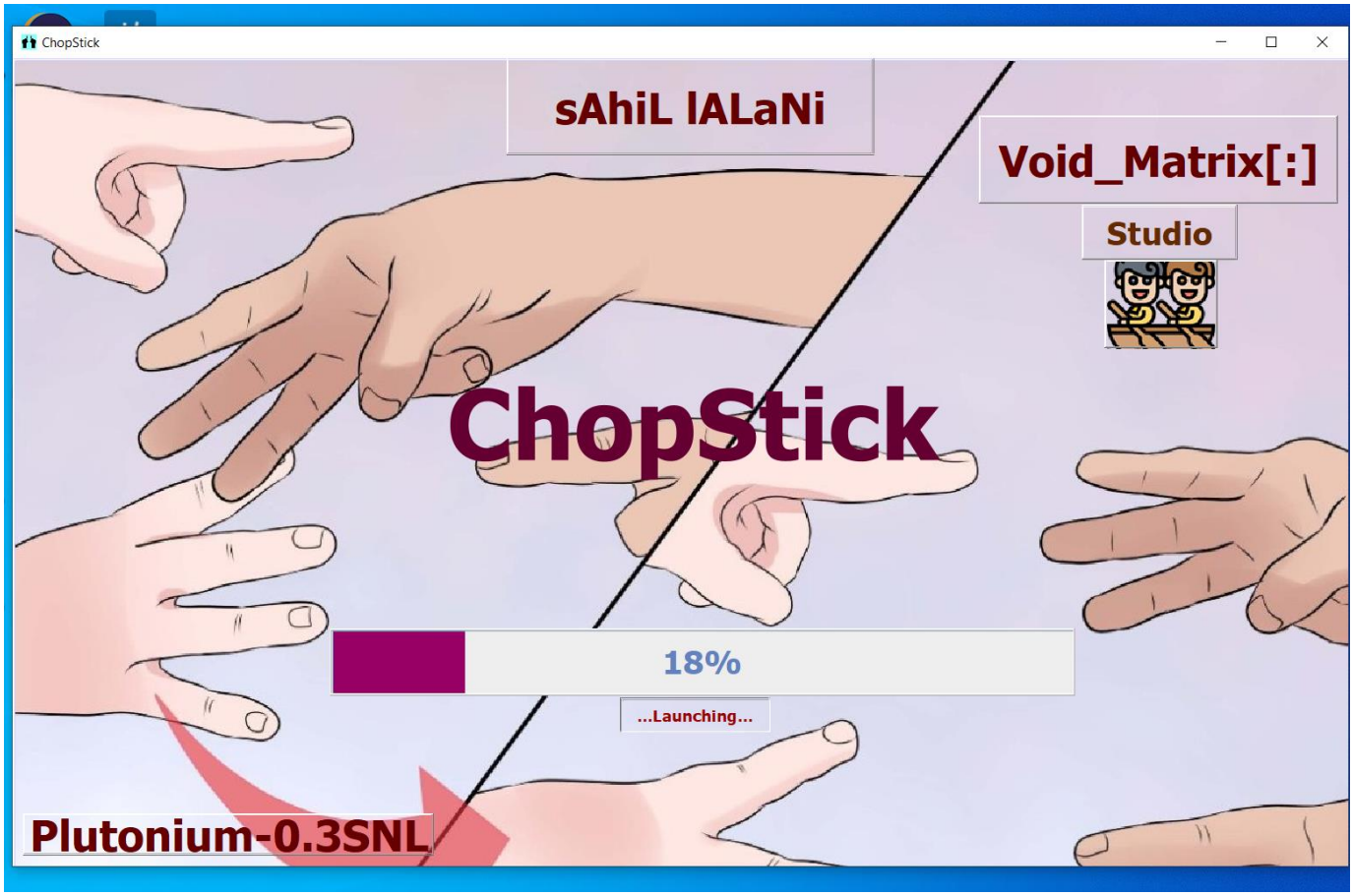




ChopStick Game Play with Snapshots

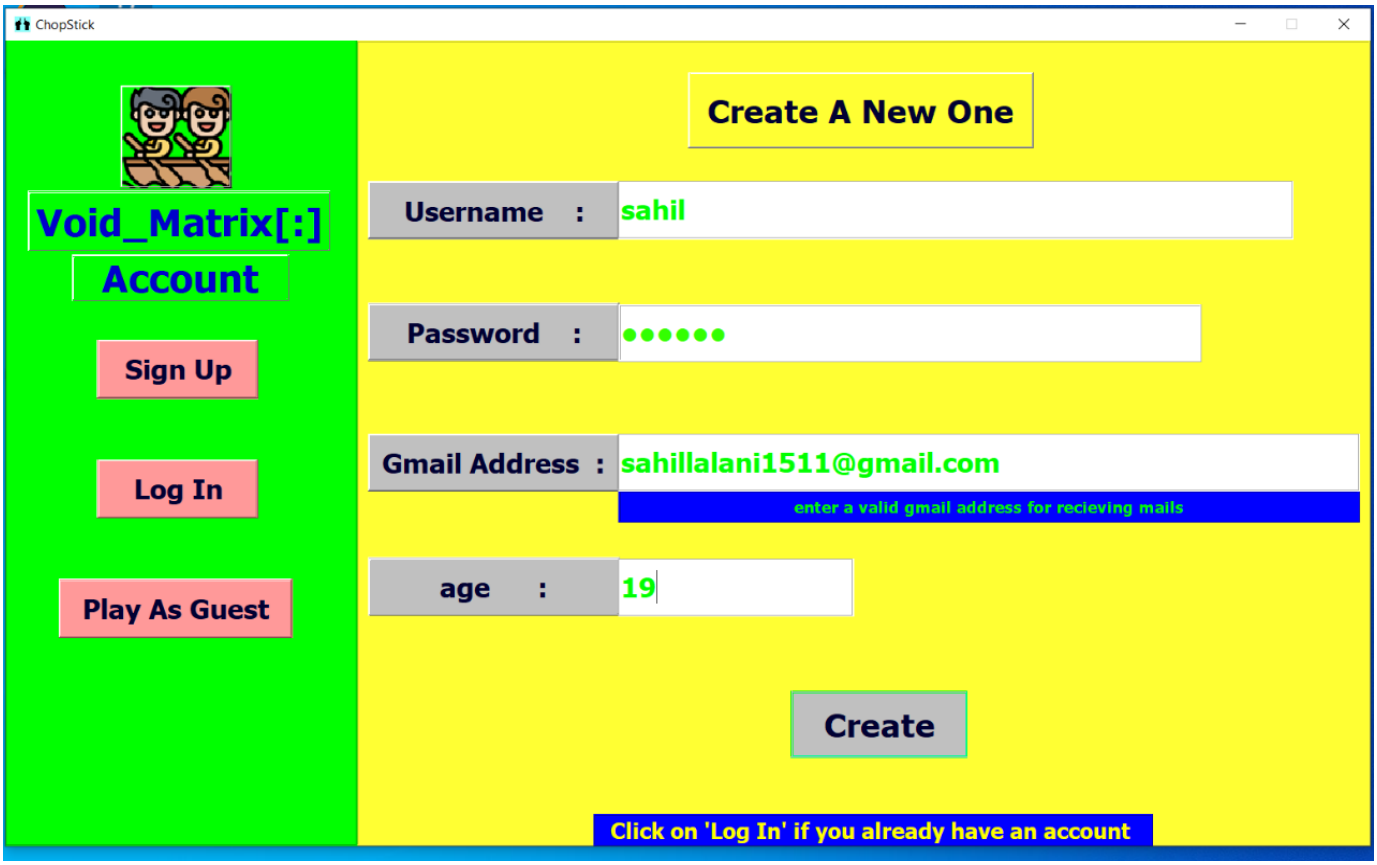


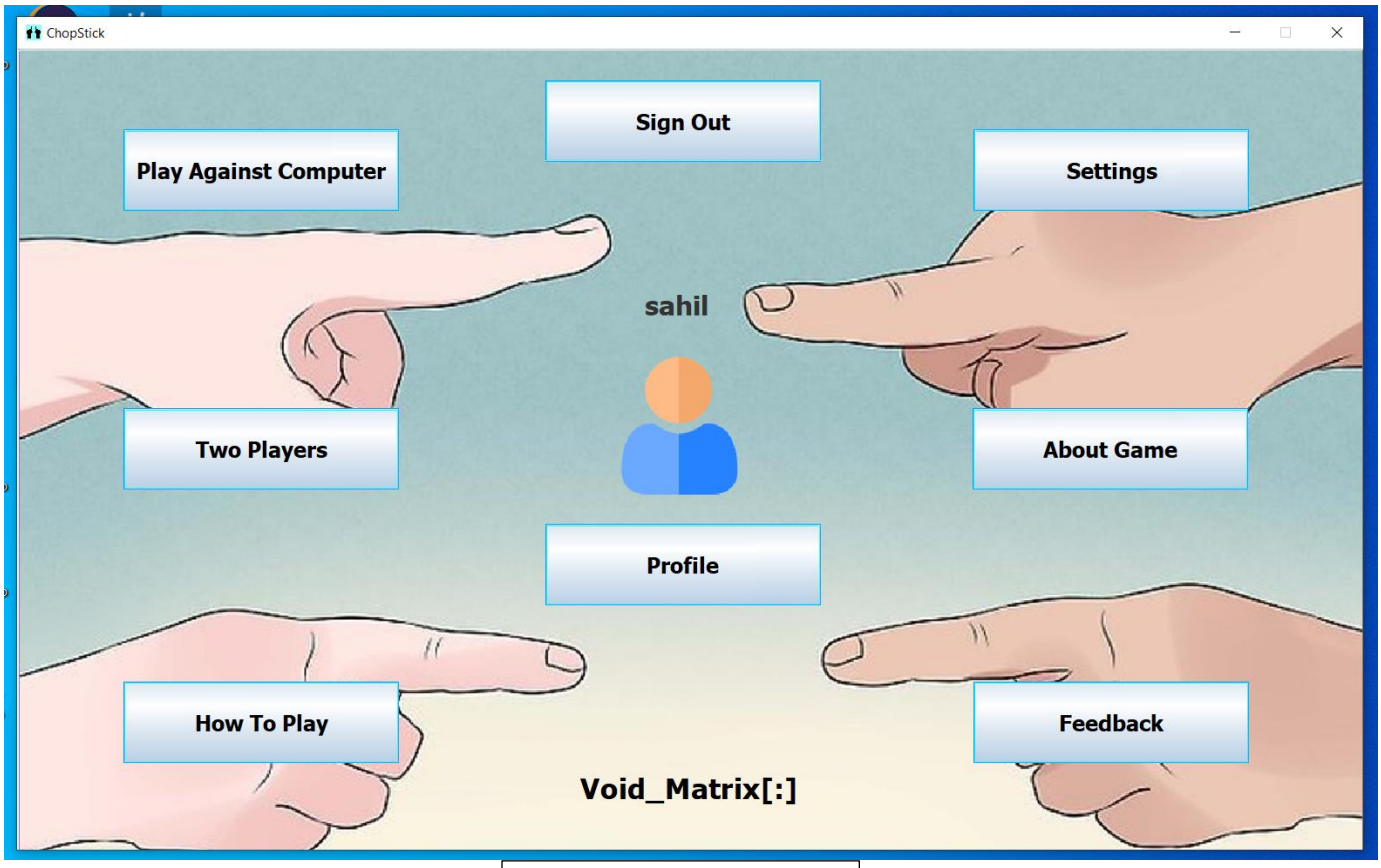
Jar & Executable file



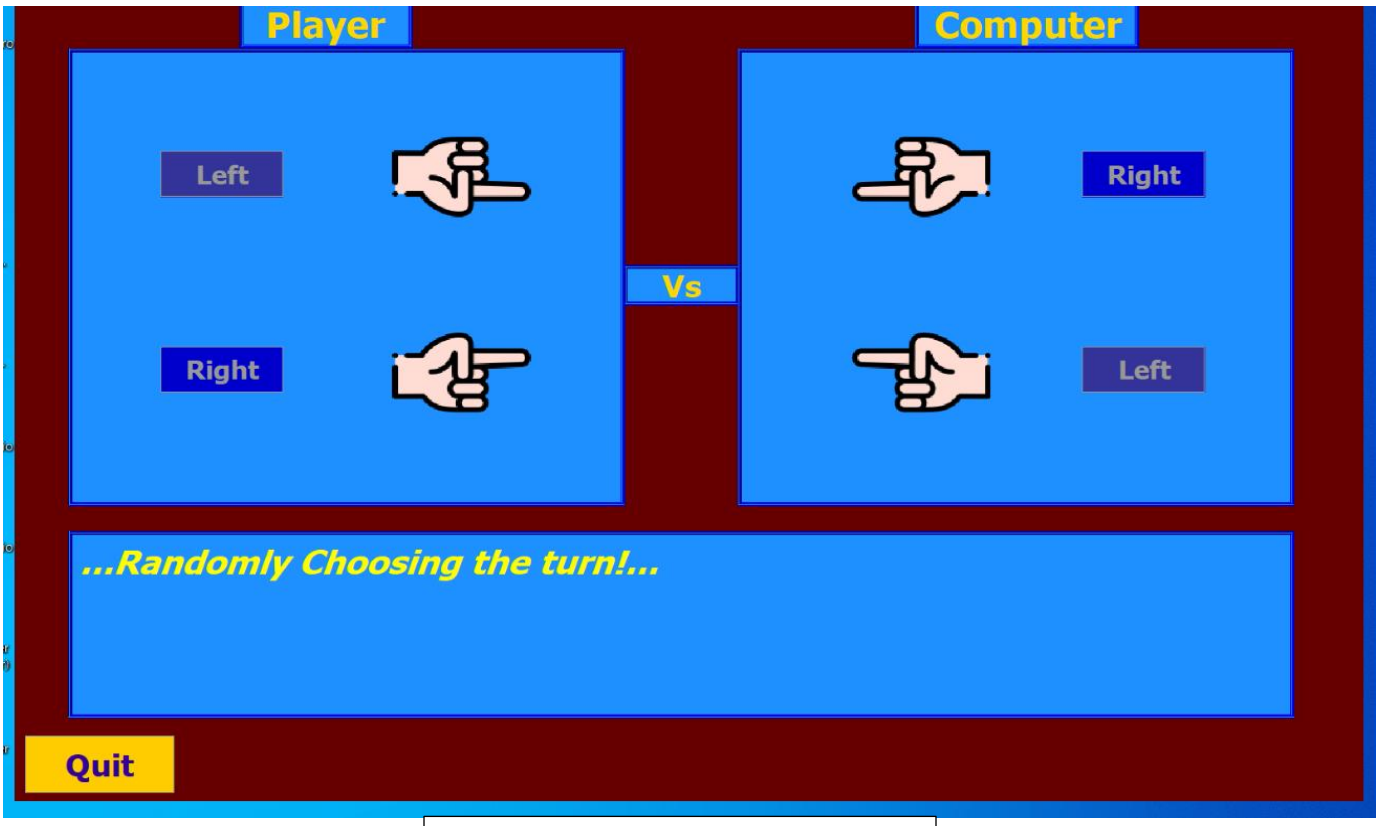
Launch frame

Void Matrix account Frame

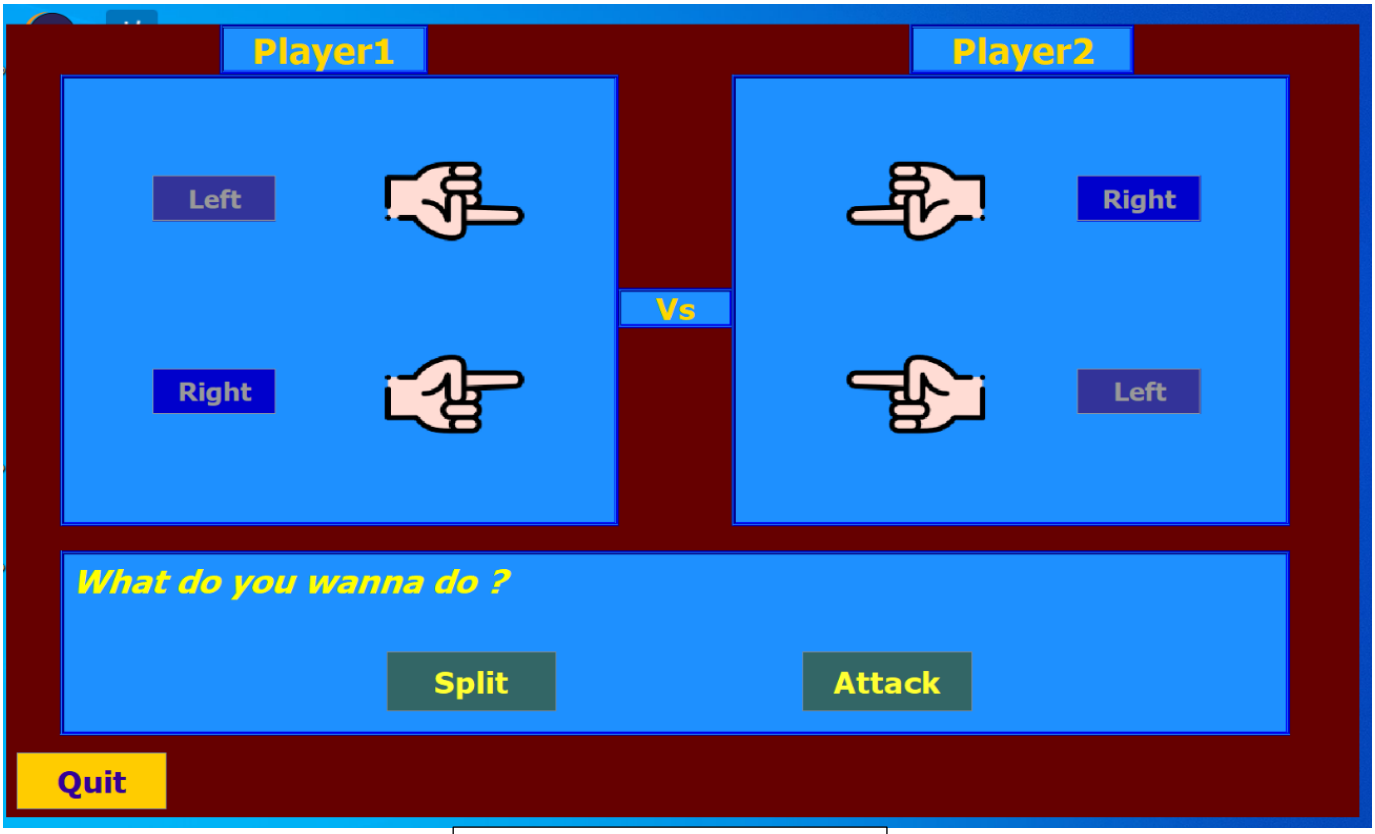




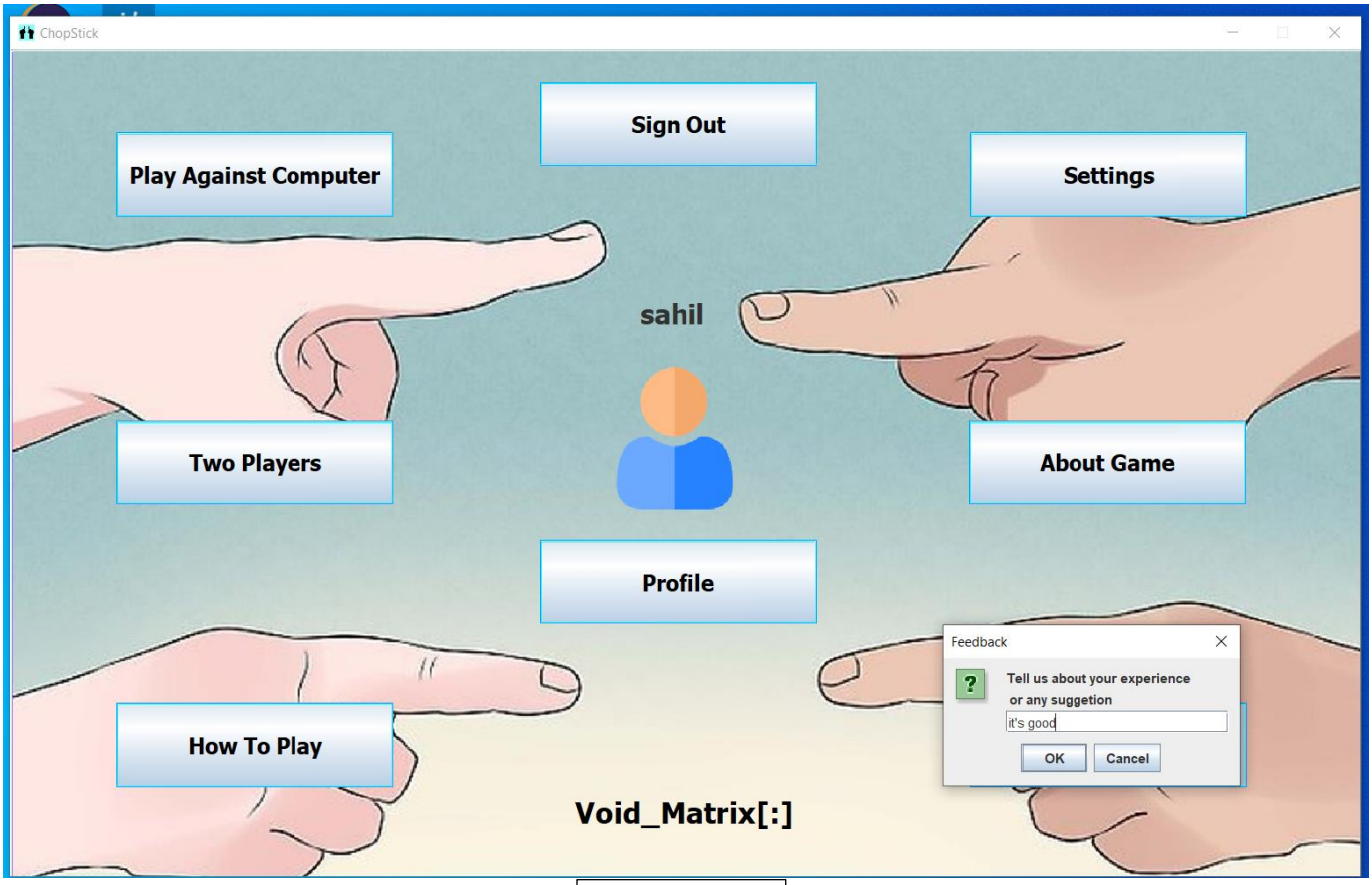
Main Menu/Frame



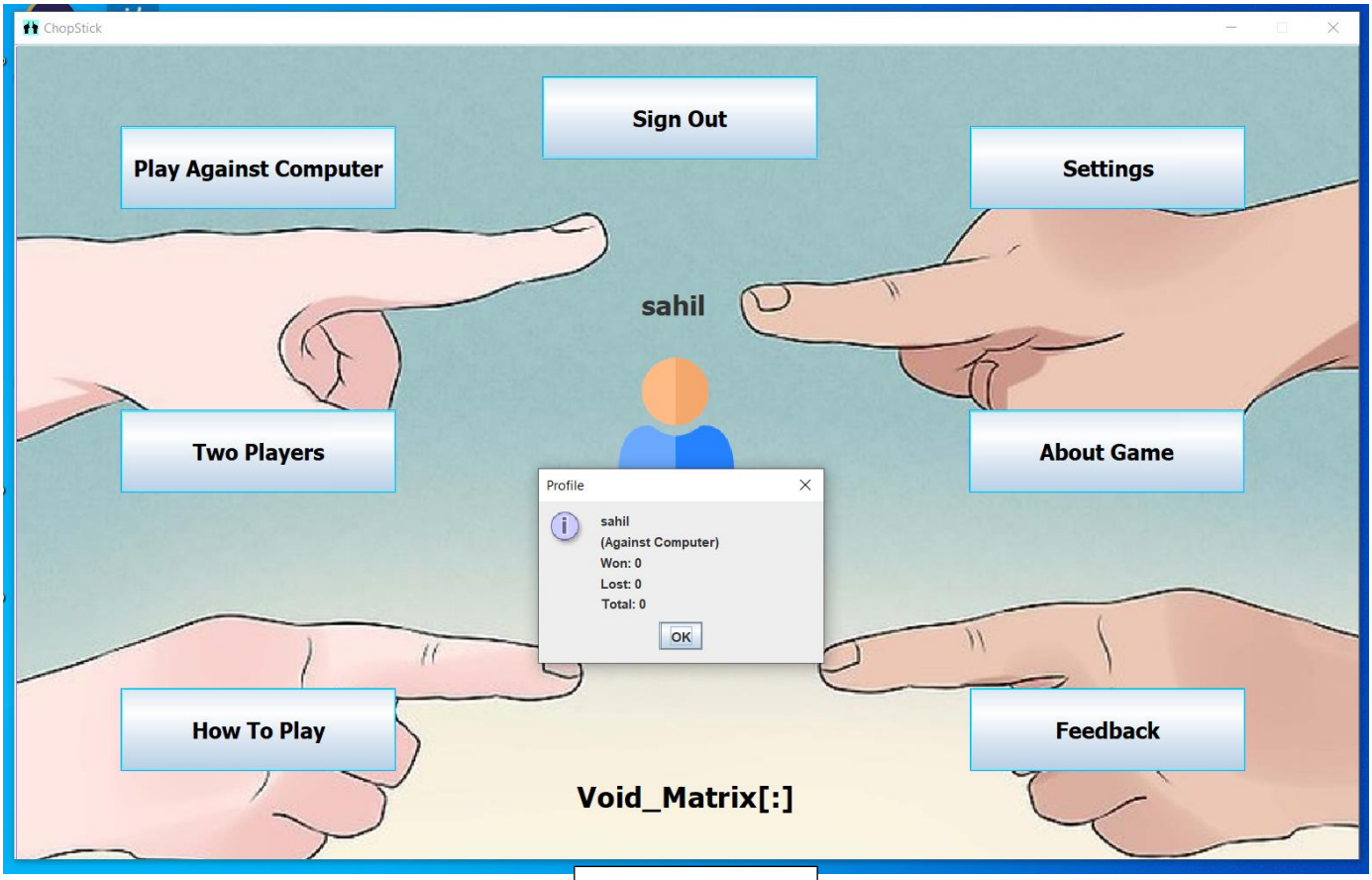
Play with computer Frame



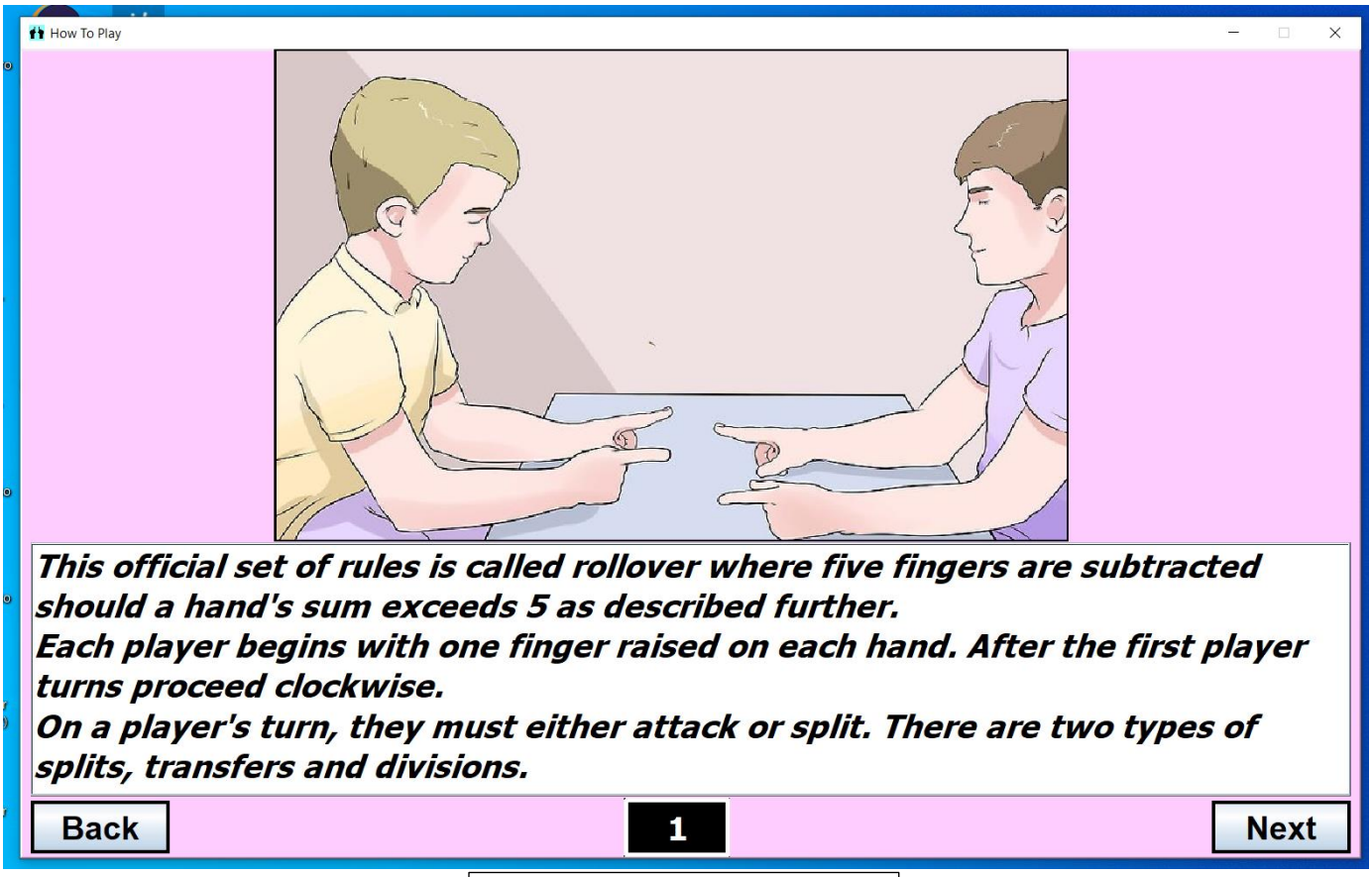
Player vs player Frame



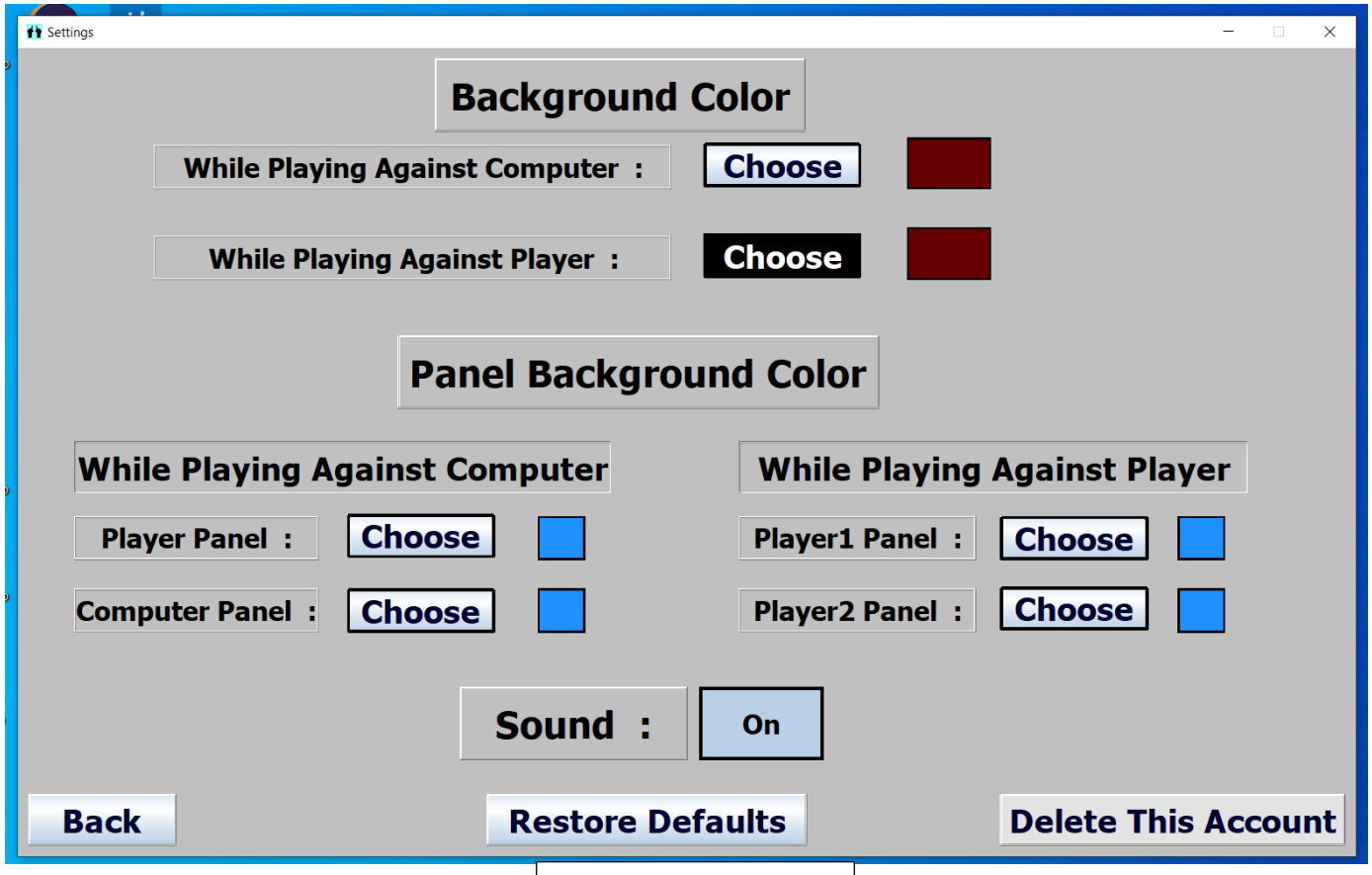
Feedback



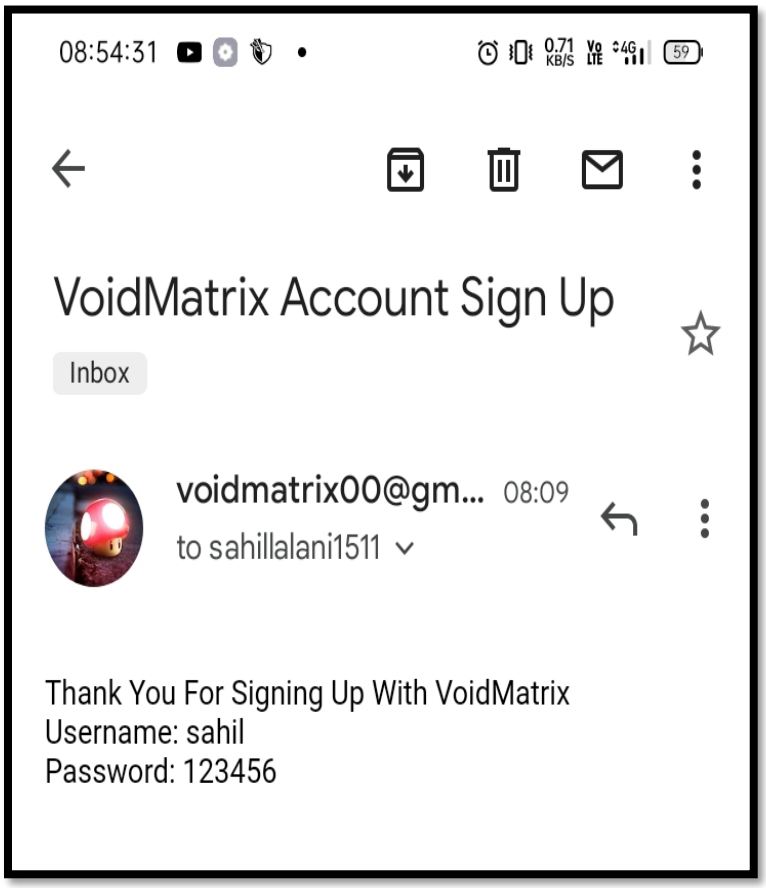
profile

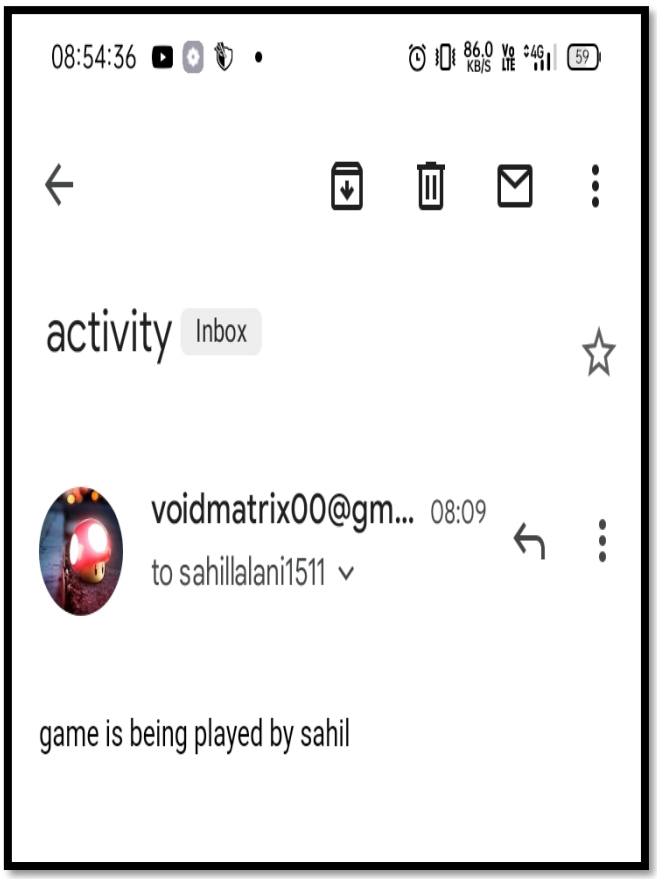


How to play Frame

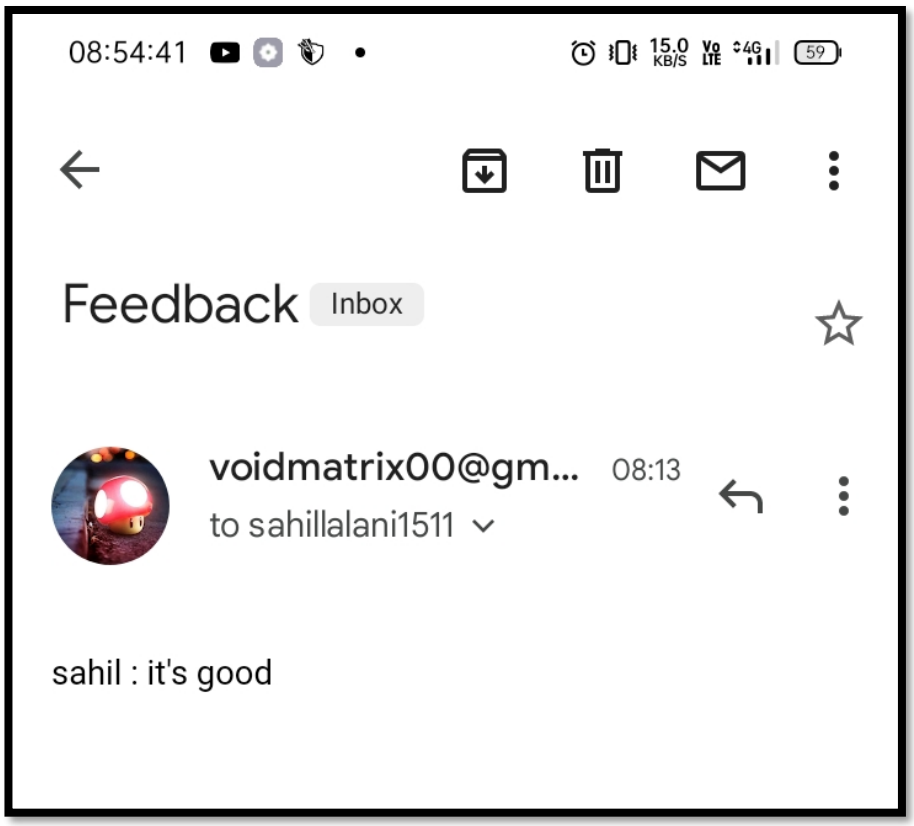


Settings Frame





Player receiving email



Me receiving feedback

Section 9: Conclusion

The main objective of this project was to design and develop a 2D Combinatorial game without having used any game engine and learn the maximum amount of things possible while doing it.

For this purpose, the different graphic elements of the game were designed, allowing me to learn how to efficiently use the core java concept, Window-Builder, swing, AWT, Flaticon.com, JAVA Mail API, remote SQL, JDBC, Launch4j in order to achieve the desired results. In this manner, each design has been done faster and better than the previous one. Similarly, I have learned how to design simple albeit necessary sound effects for the game by combining audio.

Regarding the game itself, all of the game logic and the objects were designed and subsequently implemented using Eclipse IDE. As a result, the video-game is composed of one single perfectly playable sections which includes many different objects and obstacles to overcome. This is the part that I have most enjoyed, because I really like programming and designing game logic. Additionally, coming up with finger movements' logic provided me a fun challenge, and gave me the opportunity to learn many different issues about game design and development.

Limitations

- Settings are not stored permanently
- No functionality for playing remotely with others
- No dynamic resolutions' adaption according to system
- Players can not set profile picture
- No keyboard shortcuts for buttons/processes
- The frame can not be resized as preferences while playing the game

Future enhancements

Definitely the ones that are described above in limitations and furthermore trying to make a web application for this game.

Section Appendix

Appendix A: Abbreviation & Acronyms

Term	Definition
Animation	the rapid display of a sequence of images to create an illusion of movement.
Scripting	A scripting language or script language is a programming language that supports the writing of scripts, programs written for a special runtime environment that can interpret and automate the execution of tasks which could alternatively be executed one-by-one by a human operator.
Graphics	visual presentations on some surface, such as a wall, canvas, screen, paper, or stone to brand, inform, illustrate, or entertain.
Sprites	In a computer game a sprite is the name usually given to a two-dimensional bitmap computer graphic, designed to be integrated into a level, intro or exit screen. They can be static or animated.
SRS	Software Requirements Specification
GDLC	Game Development Life Cycle
UI	User Interface
Gamer	A person who plays a game or games, typically a participant in a computer or role-playing game.
System	A system is a set of interacting or interdependent components forming an integrated whole or a set of elements (often called 'components) and relationships

	which are different from relationships of the set orits elements to other elements or sets.
JAVA Mail API	For sending emails.
JDBC	JAVA Data Base Connectivity
AWT	Abstract Window Toolkit
Swing	JAVA utility to create lightweight GUIs
Event	Changing the state of an object is known as an event. For example, click on button, dragging mouse etc.
Class	A class is a blueprint for the object.
Object	An instance of class that is real world and runtime entity that has state and behavior.
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
JRE	JAVA Runtime Environment
JVM	JAVA Virtual Machine
JDK	JAVA Development Kit
JAR	Java Archive File
JAVA File IO	For working with files and its input/output
DBMS	Data Base Management System
SQL	Structured Query Language
MySQL	MySQL is a RDBMS (Relational DataBase Management System).
phpMyAdmin	PhpMyAdmin is a web application that lets you manage (with a visual interface) MySQL Databases.

API	Application Programming Interface
RMI	Remote Method Invocation
IDE	Integrated Development Environment
GUI	Graphical User Interface
JNDI	JAVA Naming and Directory Interface

Appendix B: References

Image Icons & Symbols (flaticon.com)

https://www.wikihow.com

https://en.wikipedia.org/

The Community for Open Innovation and Collaboration | The Eclipse Foundation

https://www.db4free.net/

l0.drawio - diagrams.net

Tutorials List - Javatpoint

Java Documentation - Get Started (oracle.com)

(15872) Bro Code - YouTube

Share and Discover Knowledge on SlideShare

Launch4j - Cross-platform Java executable wrapper (sourceforge.net)

Oxford Learner's Dictionaries | Find definitions, translations, and grammar explanations at Oxford Learner's Dictionaries (oxfordlearnersdictionaries.com)

Cambridge Dictionary | English Dictionary, Translations & Thesaurus

Dictionary by Merriam-Webster: America's most-trusted online dictionary