Table of Contents

Chapter 1 Introduction

Background:

This is card game played between n players where n is greater than 1 and less than five. In starting deck of 52 cards are shuffled and share of each played is dealt to them. One or two players can be human and other will be computer players, which will play their move randomly. In each round, each player plays his/her card and in the end player who has highest rank card will be declared winner of that round. That winner of round will get all cards played in that round by other players. Player will be out of the game if he/she has left no cards to play or he/she lost all cards. Winner of game is decided when only player has all 52 cards in his/her hand and all other players have no cards to play further game.

Ranking of Cards:

SPADE has higher rank than HEART while HEART has higher rank than DIAMOND and DIAMOND has higher rank than CLUB does. Therefore, SPADE is highest rank suit while CLUB is lowest rank. In case of value of suite this order is followed from increasing rank to decreasing: ACE, KING, QUEEN, JACK, 9, 8, 7, 6, 5, 4, 3, 2. In case of same value cards, ranking of suite is checked and will be decided winner accordingly.
SPADE > HEART > DIAMOND > CLUB

Number of Players:

Maximum 4 number of players will play and minimum 2. In addition, minimum 1 and maximum 2 player will be human and other will be computer.

Saving Game Result:

Result of game and player data is saved to XML and XML file is saved in the directory of project, where project is saved.

Chapter 2 Design

Ranking of cards:

As already discussed in introduction that how ranking of cards is done. Here is technical explanation of how we have done. So, there is function in CARD class named int compareTo(Card card). Here it first compares values of both card, if value like 8>7 is greater than returns 1 else -1. Also, values of JACK, QUEEN, KING and ACE are 10, 11, 12 and 13. In case of conflict, that is when values of both cards are equal then we compare value of color or suite variable. We have set value of different suites like:
SPADE = 3
HEART = 2
DIAMOND = 1
CLUB = 0

Printing of cards:

We have saved two arrays, which saves the name of suite or color and their value. Then each card has two index values, which point to those two arrays, and arrays have name of color and value.

Shuffling of deck of cards:

This operation is done in DECK class, where we first pick two random cards in deck and exchange their positions. This process of picking up random card and exchanging them is done n times where as n is given in parameter of function. More the value of n, more random deck will be.

How Cards are Dealt:

Cards are dealt by maintaining variable `startIndexToRemoveCard.` When we start dealing of cards in main function, we give card at `startIndexToRemoveCard` positon. `startIndexToRemoveCard` is first initialized to 0 and will incremented whenever any card is dealt and will go till 51 since we have 52 cards.

Chapter 3 Software Specifications

Tools needed to run game:

This game has built on Eclipse Version: 2019-03 (4.11.0), Build id: 20190314-1200.

Language:

Java

External Libraries:

```
... GUI Dependencies ...
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

... User Input Dependencies ...
import java.util.Scanner;

... Random Value Generator ...
import java.util.Random;

... System Date Generator ...
import java.util.Date;

...XML Dependencies...
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import java.io.File;
import java.text.SimpleDateFormat;
```