Object Oriented Programming

Assignment 3

A BLACKJACK GAME USING JAVA

Section C, Third Semester, Computer Science and Engineering

Team Members:

T N Anshumanth Rao, Roll number 3

Vyshwa Manas K, Roll number 9

Shubham Natraj, Roll Number 36

Chhayank Jain, Roll Number 49

Yash Raj Sarrof, Roll Number 61

Problem Statement and Game Rules

The objective is to create a Blackjack in Java based on Applets. In Blackjack, the aim is to have a hand total greater than your opponent without exceeding 21. The hand total is the sum of all the card values in a player's hand. The cards from 2-9 have the same face value, Ace has a value of 1, Jack can be counted as 1 or 11 as convenient, and the King, Queen, and Ten all have ten points.

Each players, here the user and the computer, are dealt two cards in the beginning. The user then has the option to draw another card, or "Hit", or decide to not draw any card, that is "Fold". The former is done when the hand total is low, while the latter is done when the player decides that the total is close enough to 21. The computer's turn is automated.

If the hand total exceeds 21, the player is said to have "Busted". If a player has busted, the other player wins as long as that player has not busted as well, in which case it is a draw. If neither of the players have busted, then the player with the greater hand total is the winner.

Classes and Code

In accordance with the principles of Object Oriented Programming, we have created classes to represent real-life objects: a playing card, a deck of cards, and a player's hand. Each class contains the required functions within it, allowing us to perform necessary operations and access the necessary members from the Main part of the program through the objects.

The Card class represents a single playing card, storing the value, the suit, and the name of the card. The card name is given by a string that matches the name of the image file for that card.

Next, there is a Deck class. The Deck object will be common to all the players in the class because there is only one deck of cards in play. There is an array of cards to hold the cards already drawn that, paired with a Search function, will ensure the cards are not repeated. The important function in this class is a Draw that returns a random new Card.

The Hand class represents each individual player. It holds the player's cards, total hand value, a reference to a deck of cards, and Boolean values representing if the player is still in play and if the player has busted. There is a Draw function to add a card to the hand, and accessor functions for the values that need to be used in the Main.

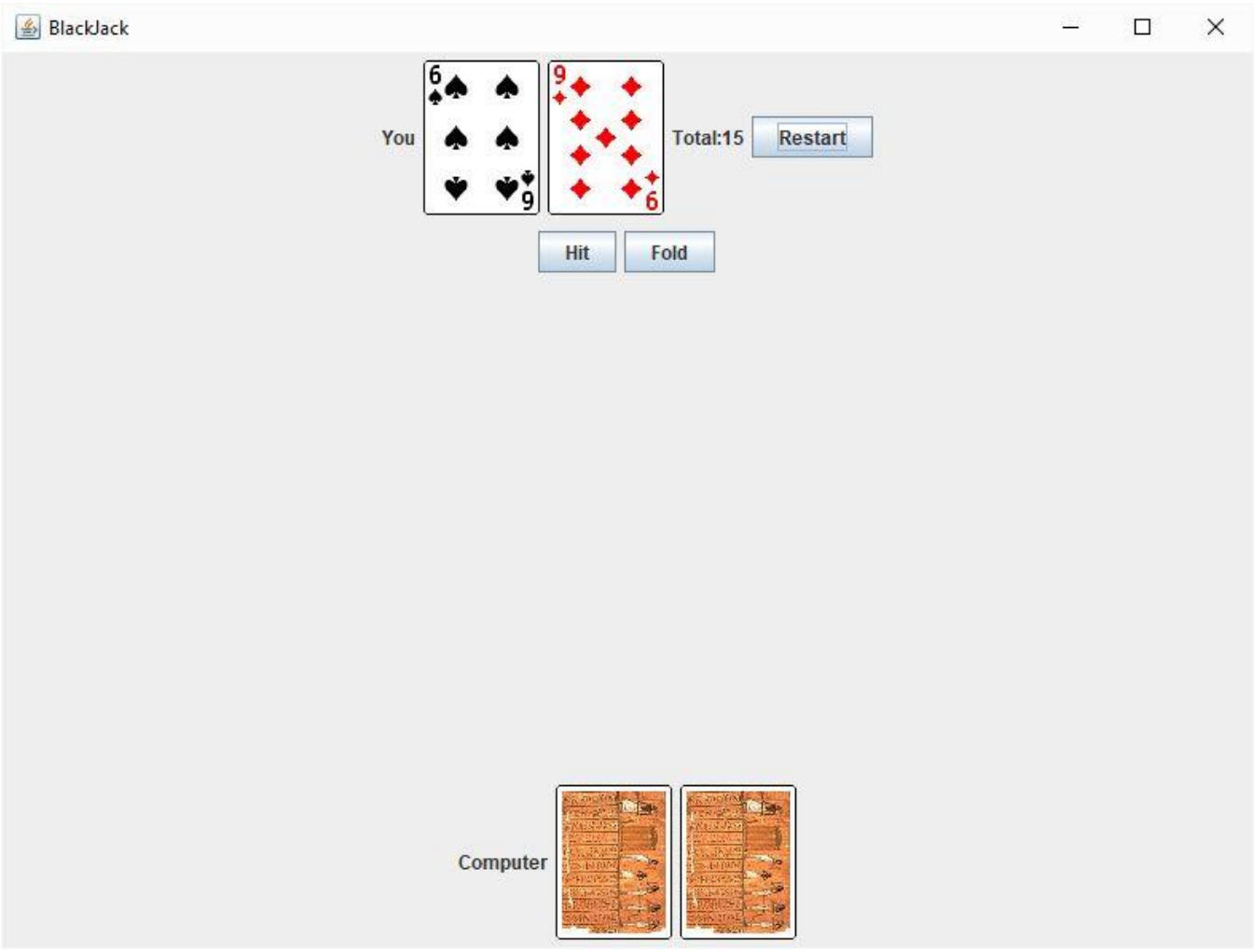
The Blackjack class contains the Main method and the implementation of the game using the above classes. It creates the user interface as well. There is a GUI function to create the JFrame and add the buttons and cards to it. The method for the player's turn is also defined in this class.

While the user can decide for himself whether he wants to hit or fold, the computer's decision has to be automated. This decision is coded to mimic an actual player's thought process to an extent. If the hand total is 14 or lower, the computer takes the decision to hit. Any value greater than 18 is quite close to the maximum of 21. Hence, the computer folds if the hand total is 19 or

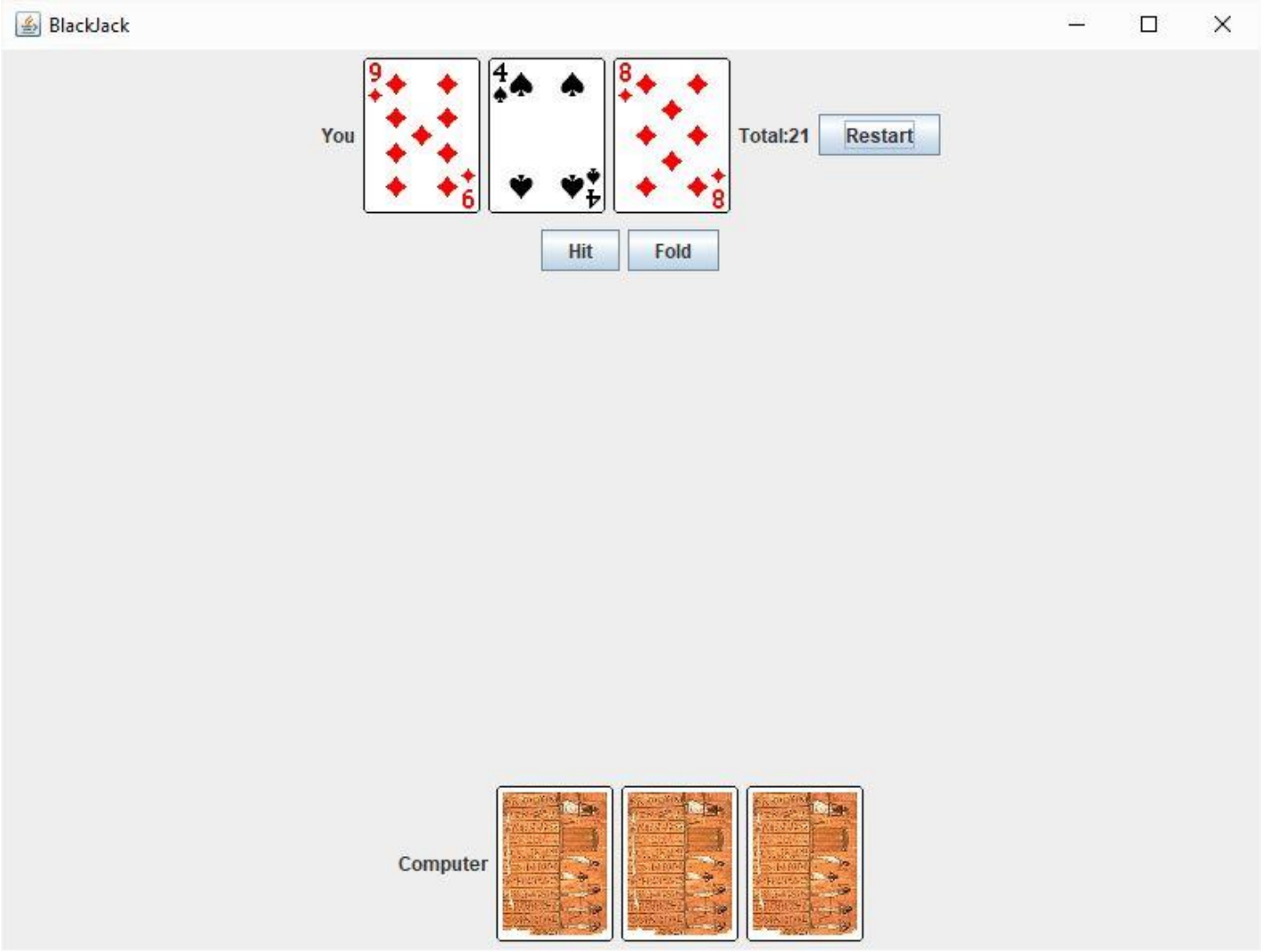
greater. With intermediate values, the computer randomly decides whether to hit or fold.

Input/Output

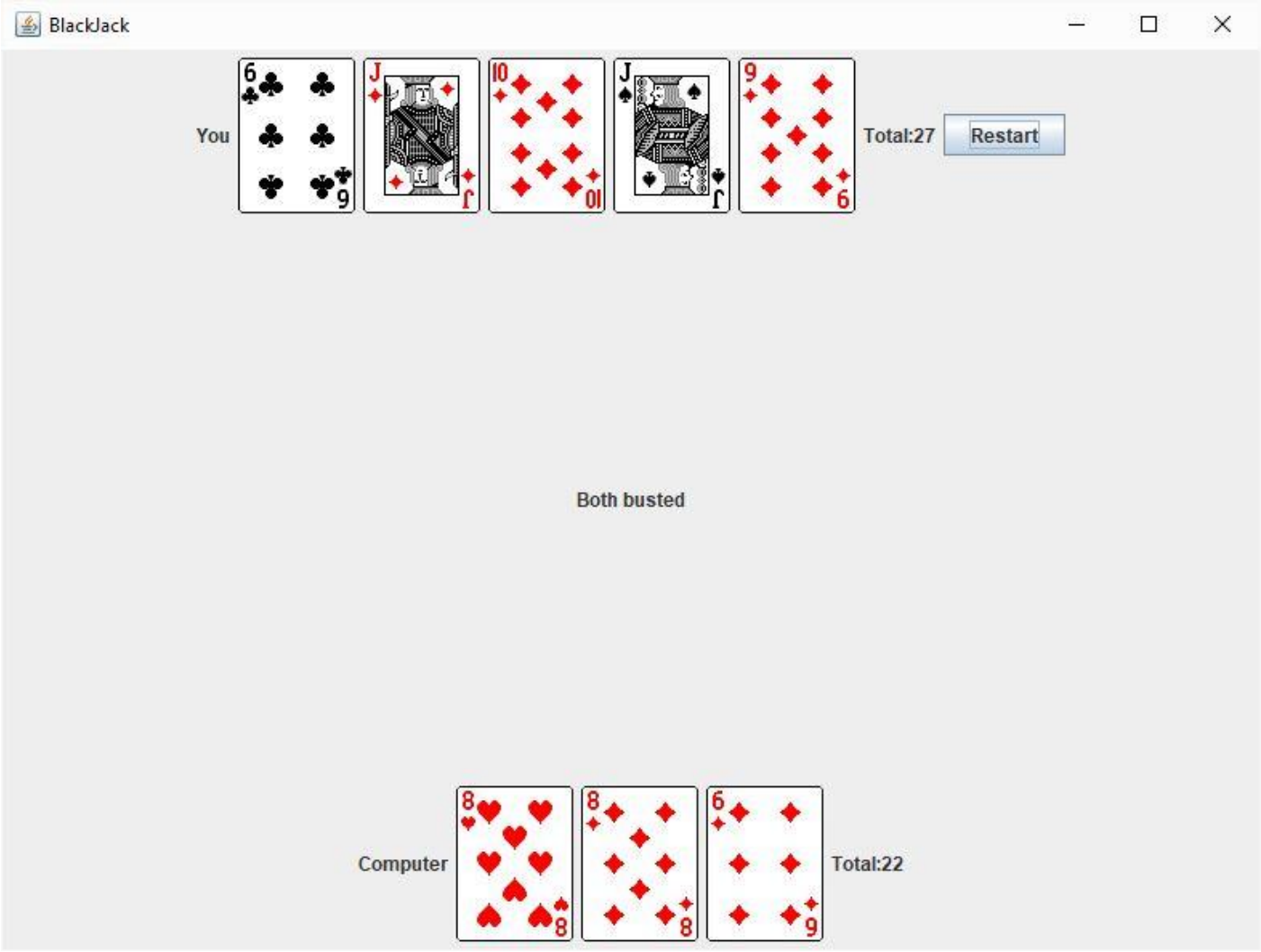
Beginning of Game

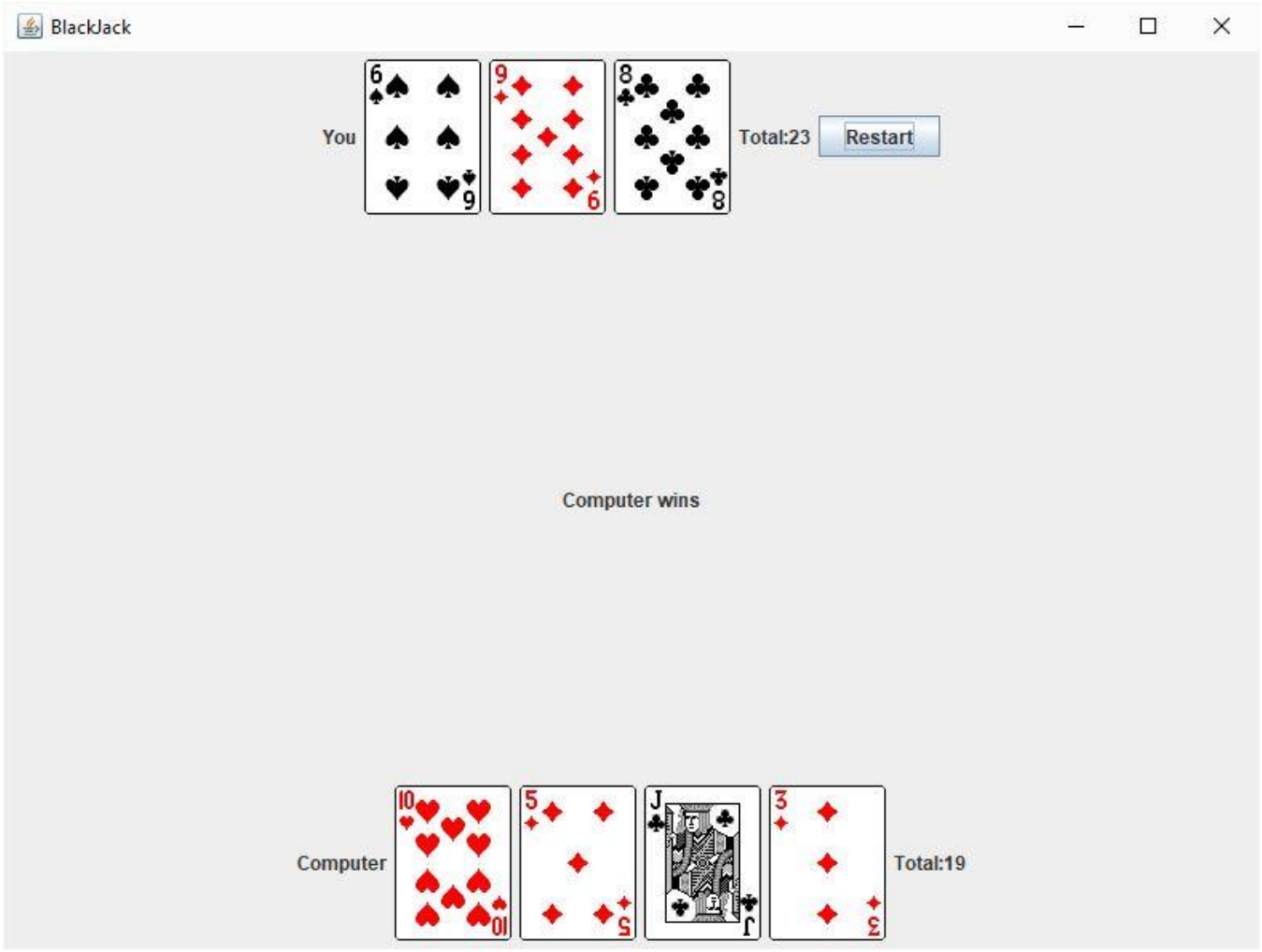


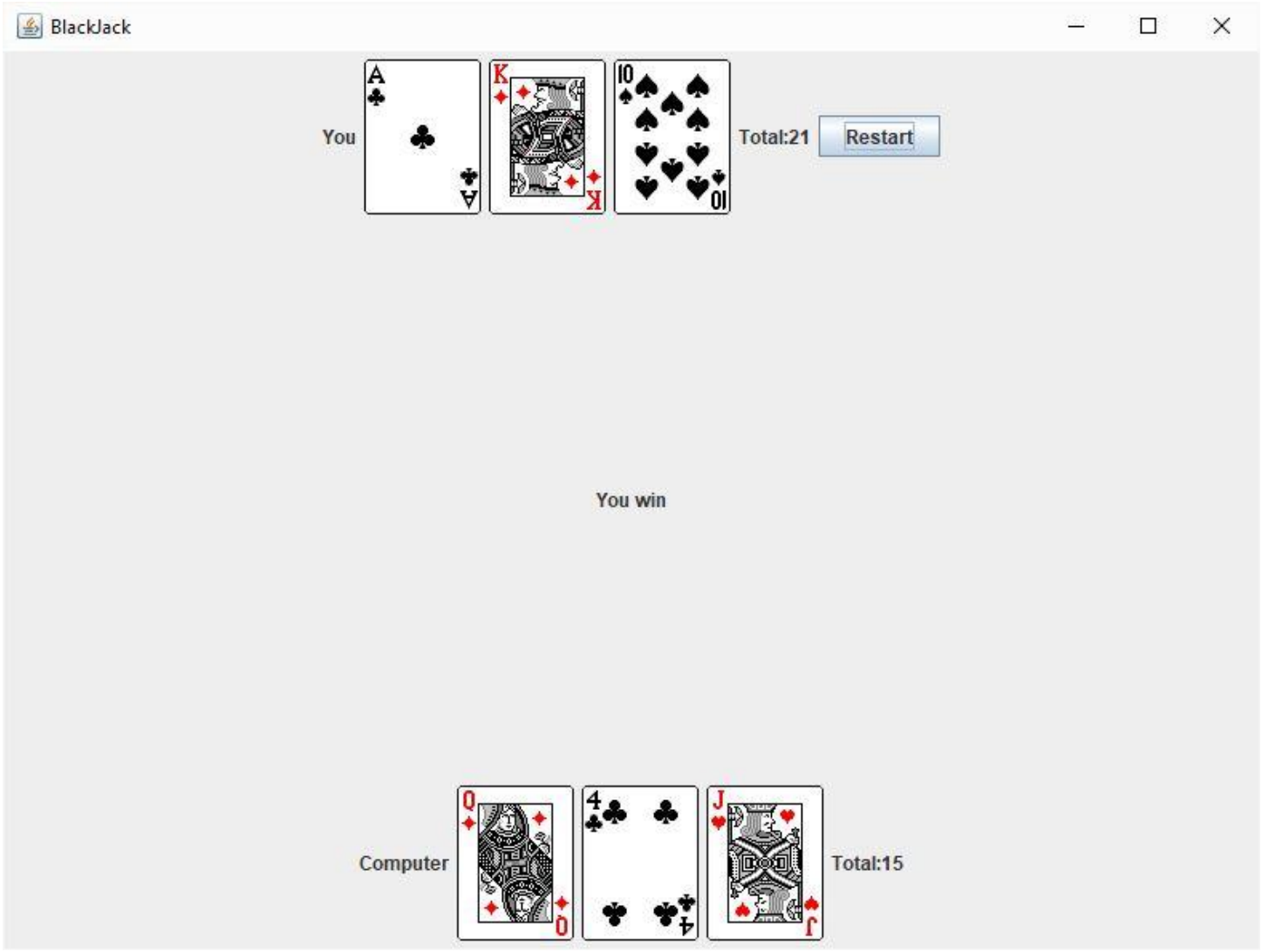
During the game



Three different outcomes at end of game




BlackJack
You
Total:23
Restart
Computer wins
Computer
Total:19


BlackJack
You
Total:21
Restart
You win
Computer
Total:15

Limitations of Design

The computer's decision to Hit or Fold, while based on its own hand total, does not mimic human behaviour completely. It's risk assessment is hard-coded, and does not take into account how many cards the other player has drawn, or what cards are already dealt.

References

- Java: The Complete Reference by Herbert Schildt
- Java: A Beginner's Guide by Herbert Schildt
- Wikipedia
- Stack Overflow