Card Games

Chiu Zi Xin

February 22, 2016

Black jack is a famous card game. Player who get the closest value to 21 wins the game, bigger than it will lose.

Figure 1: Lets see how much luck we got.

The Java Code

```
import java.util.*;
import java.util.Scanner;

public class Cards{

  static int count=52;

  public static int rand(int high){
      return (int) (high*Math.random()+1);
    }

  public static void shuffle(String[] the_deck, int
  switches){
    String temp;
    int a; int b;
    for(int i=0; i<switches; i++){
      a = rand(52);
      b = rand(52);
      temp = the_deck[a-1];
      the_deck[a-1] = the_deck[b-1];
      the_deck[b-1] = temp;
    }
  }

  public static String deal(String[] the_deck){
    count=count-1;
    return the_deck[count];}
```

We need to import these to use Arraylist and scanner method.

This method returns a random integer number from the range that user setting the highest value.

This method uses for loop with the rand method to shuffle the cards.

This method will simulate dealing a card to a player.

This method use if statement to see the input is ace.

```
 public static int aces(String the_card){
  if(the_card.charAt(0)==’A’){
      return 1;}
    else{
      return 0;}
  }
```

This method overload the "aces" method and it returns the number of aces in a array.

```
public static int aces(String[] the_hand){
  int sum=0;
  for(int i=0; i<the_hand.length;i++){
    sum = sum + aces(the_hand[i]);
  }
  return sum;
}
```

This method overload the "aces" method and it returns the number of aces in a arraylist.

```
public static int aces(ArrayList the_hand){
  int sum=0;
  for(int i=0; i<the_hand.size();i++){
    sum = sum + aces(the_hand.get(i).toString());
  }
  return sum;
}
```

This method counts the value of the card that have more than one unit.

```
public static int value(String the_card){
  char first = the_card.charAt(0);
  if (first==’1’|first==’J’|first==’Q’|first==’K’){
    return 10;
  }
    else if(first==’A’){
      return 11;}
    else{
      return Character.getNumericValue(first);
    }
  }
```

This method overload the "value" method and sum up the value of the cards in the array.

```
public static int value(String[] the_hand){
  int sum=0;
  for(int i=0; i<the_hand.length;i++){
    sum = sum + value(the_hand[i]);
  }
  return sum;
}
```

This method overload the "value" method and sum up the value of the cards in the arraylist with the ace logic.

```
public static int value(ArrayList the_hand){
  int sum=0;
  int num_aces=aces(the_hand);
  for(int i=0; i<the_hand.size();i++){
    sum = sum + value(the_hand.get(i).toString());
  }
  while(num_aces>0 && sum>21){
    sum=sum-10;
    num_aces=num_aces-1;
  }
  return sum;
}
```

We are getting in to the main method.

```
 public static void main(String[] args){

   Scanner scan = new Scanner(System.in);
```

Here are creating a deck of cards.

```
   String[] deck = new String[52];
   String[] suit = new String[4];
   int[] card = new int[13];
```

Here uses for loop to define the suit of the cards.

```
   for (int i=0; i<card.length; i++){
     card[i]=i+1;}
   String cardName;
   suit[0] = "Clubs";
   suit[1] = "Diamonds";
   suit[2] = "Hearts" ;
   suit[3] = "Spades";
```

Here uses nested for loop to give a card name to the elements in the array.

```
    for(int i=0; i<4; i++){
     for(int j=0; j<13; j++){
       if(j==0){cardName="Ace";}
       else if(j==10){cardName="Jack";}
       else if(j==11){cardName="Queen";}
       else if(j==12){cardName="King";}
       else {cardName=Integer.toString(card[j]);}
       deck[ 13*i+j ]= cardName + "_" +suit[i];
     }
   }
```

Shuffling the deck 1000 times...

```
shuffle(deck, 1000);

  String say;
boolean state=true;
```

Creating arraylist for player and dealer and dealing cards to them.

```
  ArrayList hand = new ArrayList();
  ArrayList dealer_hand = new ArrayList();
  dealer_hand.add( deal(deck) );
  dealer_hand.add( deal(deck) );
  hand.add( deal(deck) );

  while(state){

  hand.add( deal(deck) );
```

Printing the deck informations and values of player and dealer.

```
  System.out.println("Dealer showing: " +
  dealer_hand.get(1));
  System.out.println("Contents of hand: " + hand);
  System.out.println("Your score is: " + value(hand));
```

If player’s hand is already higher than 21 then busted.

```
  if(value(hand)>21){
    System.out.println("BUST!!!!");
    break;
  }
```

Or asking player if they hit or stand.

```
  System.out.println( "hit[H] or stand[S]?");
       say=scan.nextLine();
       if(say.equals("H")){state=true;}
       else{state=false;}
  }
```

A dealer logic that if their hand’s value are smaller than 17, they will hit for a card.

```
  while( value(dealer_hand)<17 ){
    dealer_hand.add( deal(deck) );
  }
```

Printing the deck informations and values of player and dealer again.

```
  System.out.println("Dealer has: " + dealer_hand);
  System.out.println("Dealer score is: " +
  value(dealer_hand));
```

```
    if( (value(hand)>value(dealer_hand) &&
    value(hand)<22) | (value(dealer_hand) > 21) ){

      System.out.println( "YOU WIN !!!!");
    }
    else{System.out.println( "YOU LOSE. BOO !!!!");}

  }
}
```

An if statement to see if you win or not.

Output sample

```
Dealer showing: 5_Diamonds
Contents of hand: [7_Spades, 4_Clubs]
Your score is: 11
hit[H] or stand[S]?
H
Dealer showing: 5_Diamonds
Contents of hand: [7_Spades, 4_Clubs, Jack_Spades]
Your score is: 21
hit[H] or stand[S]?
S
Dealer has: [2_Diamonds, 5_Diamonds, King_Hearts]
Dealer score is: 17
YOU WIN !!!!
```

Here is an example of the output of the code.