Best Card Game Ever — The Black Jack

Yuet Shing ,Au

March 6, 2016

Here i am going to introduce a card game to you.Thats the BLACK JACK, we usually play it with poker, but what it there is in poker near you? So here are the game black jack with code but not a single poker cards, it can still be fun and work let check it out!

THE NEW LANGUAGE

We are using a new language now named JAVA. Java have a special working system than python, they need Classes and Methods, also they always needed to specialise the data type of and object, and the organisation of the code doesn't matter how it operates.

```
public class "the name you want for the class"
```

This code is what we always need and started with.

JAVA CODE

```
import java.util.*;
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];
```

We need to import util* when we are going to use the methods Scanner or Arrylist in the following code.The * have included that all things in java.util.

The count represents the number of cards remaining in the deck.

This is going to pick a random number for the range that you set.

This is the methods with no return type, that why we put "void" and this methods help us to shuffle the card in deck with using for loop.

```
     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];}

 public static int aces(String the_card){
  if(the_card.charAt(0)=='A'){
      return 1;}
    else{
      return 0;}
  }
 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;
}
 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;
}
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 is the method deal is used to get one card from the deck to the player.

This method is changing what if the cards in deck that started with "A" to the value 1.

Here are a overloading code that using aces, and these code can help us to check our cards in array that start with "A" and turn it to 1.

Here are a overloading code that using aces, and these code can help us to check our cards in arraylist that start with "A" and turn it to 1.

This method is going to changing the card in deck that start with "1", "J", "Q", "K", to the value 10, and if it is "A" get the value 11, and the other number just get what their own number value.

```
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;
}

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;
}
```

Here is for loop that counting the sum value of cards in your hand.

There is a Ace system, we are using while loop to work when the sum value of your hand cards is over 21 then the Ace value will be 1, but when the sum value is less than 21, the Ace value will be 11.

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

   Scanner scan = new Scanner(System.in);

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

   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";
   for(int i=0; i<4; i++){
     for(int j=0; j<13; j++){
```

Here is the main methods setting the cards and the suit, number of cards with the for loop and if else fusions.It help us setting the 52 cards with 4 suit and use for loop to set each suit name, after that setting the card number and different name.

```
      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];
    }
  }
```

```
  String say;
boolean state=true;

  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) );

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

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

Now here we are setting a new arraylist for the hand of dealer and player, because we do not know weather we need to add or not.

Here is showing out your cards value in hand and the dealer's hand and, if the values of all your card sum is greater than 21 the system will print out a " BUST " and you lose.

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

    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 !!!!");}

  }
}
```

At the end we are using if else fusion that if dealer hand values is less than 17 he must add cards, and lastly values show out and if player value is greater then win else is lose.