# BLACKJACK PROJECT

# (BLACKJACK GAME)

## PROJECT REPORT

*in partial fulfilment for the award of the degree*

*of*

**BACHELOR IN COMPUTER APPLICATIONS (B.C.A.)**

PROJECT GUIDE:

**Ms. Shreya Agrawal**

SUBMITTED BY:

**Ahmad Faraz Ansari**

**Enrollment No.: U1946007**

**Date Of Submission: 18-01-2022**

**CENTER OF COMPUTER EDUCATION**

**Institute of Professional Studies, University of Allahabad**

**2019 – 2022**

# Table of Contents

# CERTIFICATE

This is to certify that **Ahmad Faraz Ansari** of **Centre of Computer Education, Institute of Professional Studies, University of Allahabad, Prayagraj** has successfully completed his project on the topic **BlackJack Project (BlackJack Game)** under the guidance of **Ms. Shreya Agrawal** during the academic year 2019 - 22 as per guidelines given by **University of Allahabad, Prayagraj**.

| | |
|---|---|
| **Ms. Shreya Agrawal**<br>**Guide** | **Dr. Ashish Khare**<br>**Course Coordinator** |

# ACKNOWLEDGEMENT

Project is an important milestone in the completion of any Professional Course. As a student of B.C.A, I got the golden opportunity to do this work.

It gives me immense pleasure to express my feelings of deep gratitude towards the subjects without whom, it would have been very difficult to accomplish this mammoth project.

I wish to express my thank to my parents, my supervisor **Ms. Shreya Agrawal** as well as **Dr. Ashish Khare (Course Coordinator),** who provided me this golden opportunity to work on this wonderful project called "**BlackJack Project (BlackJack Game)",** which also helped me in doing lot of research, which gave me insight on so many new things are going to help me in the foreseeable future.

I would like to thank all those who have helped me in providing direction, information and advice at all stages in this Project.

I take this opportunity to thank the **University of Allahabad** for giving me chance to do this project.

# DECLARATION

I, **Ahmad Faraz Ansari**, hereby declare that the project report entitled “**BlackJack Project (BlackJack Game)**” has been submitted to **University of Allahabad** in partial fulfilment of the requirement for the award of degree of B.C.A., is a record of Bonafede Project work carried out by me under the guidance of **Ms. Shreya Agrawal**.

I further declare that this project has not been submitted and will not be submitted, either in part or full, for the award of any other degree or diploma in this institute or any other institute or university.

The work contained in the report is original and has been done by me under the general supervision of my supervisor.

I have followed the guidelines provided by the University of Allahabad in writing this report.

**Date: 18-01-2022** **AHMAD FARAZ ANSARI**

**Place: Allahabad** **B.C.A. -5th Semester**

# INTRODUCTION

BlackJack Project or popularly known as BlackJack Game, is a Multimedia Game, developed using core Java functionalities.

Blackjack is one of the world's most renowned on-line / off-line casino games. It's one of those casino games that everybody can have learned of, and is additionally called twenty-one. Before we begin taking part in blackjack on-line / off-line, we must have a tendency to advocate that we are taking it slow to be trained the rules of BlackJack.

All the foundations of BlackJack are processed. Of all on-line/off-line casino games, BlackJack is the game that gives you the simplest possibilities of feat the table as a winner. It utilizes all the best of Java concepts from creating window to making

The main objective of this project includes:

- To create a simple yet enjoyable casino – based game.
- To create a game which is less time and resource consuming.
- Contains simple rules which can be easily understood by the Player.
- No unfair means can interfere with decision - making.

# PROPOSED SYSTEM

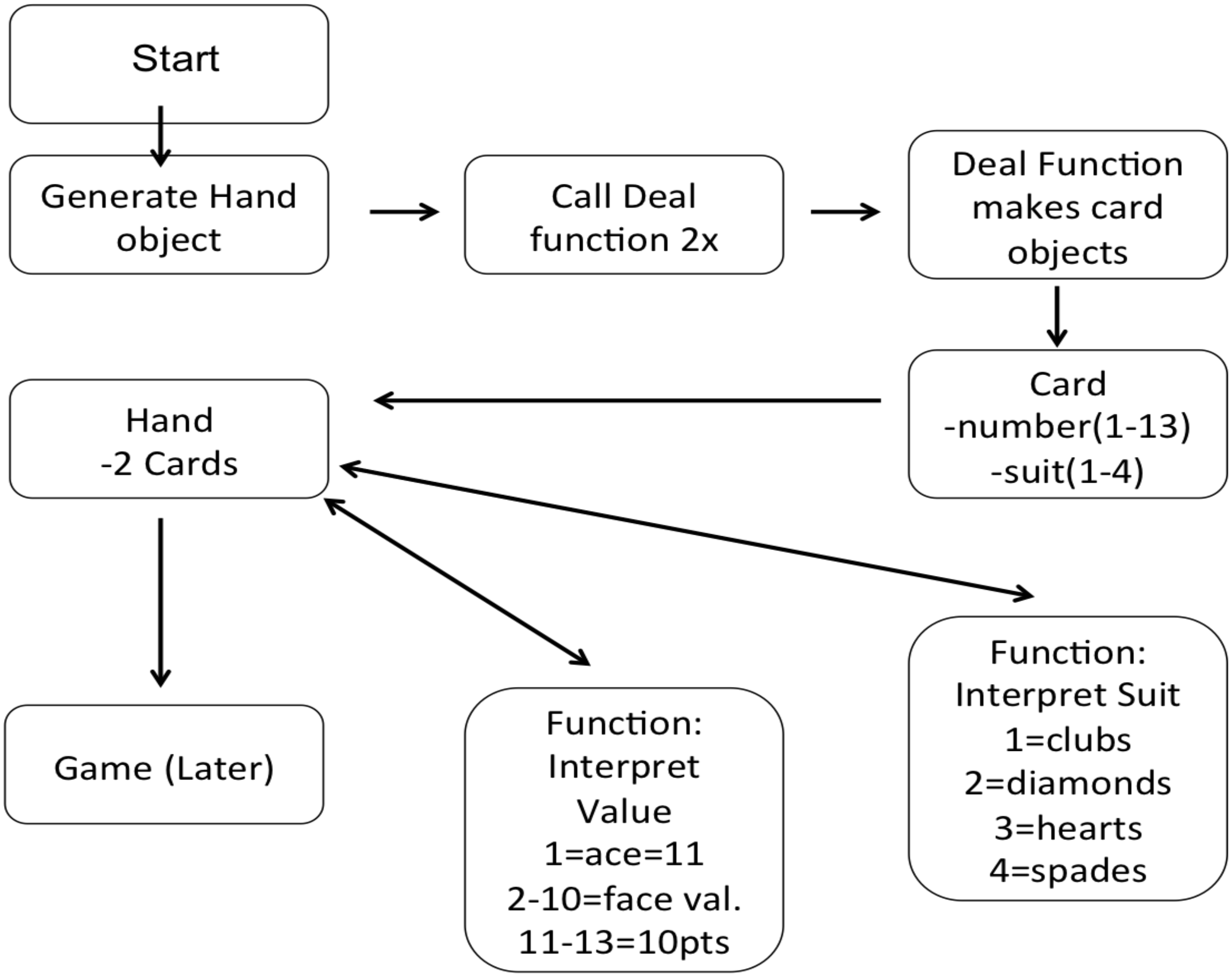
The Primary Reason for the existence of the BlackJack Project (BlackJack Game) is the problem associated with online casino games.

They are very time, resource consuming and very expensive. BlackJack Project (BlackJack Game), on the other hand, is made to deliver best user-experience possible in lesser time and resources.

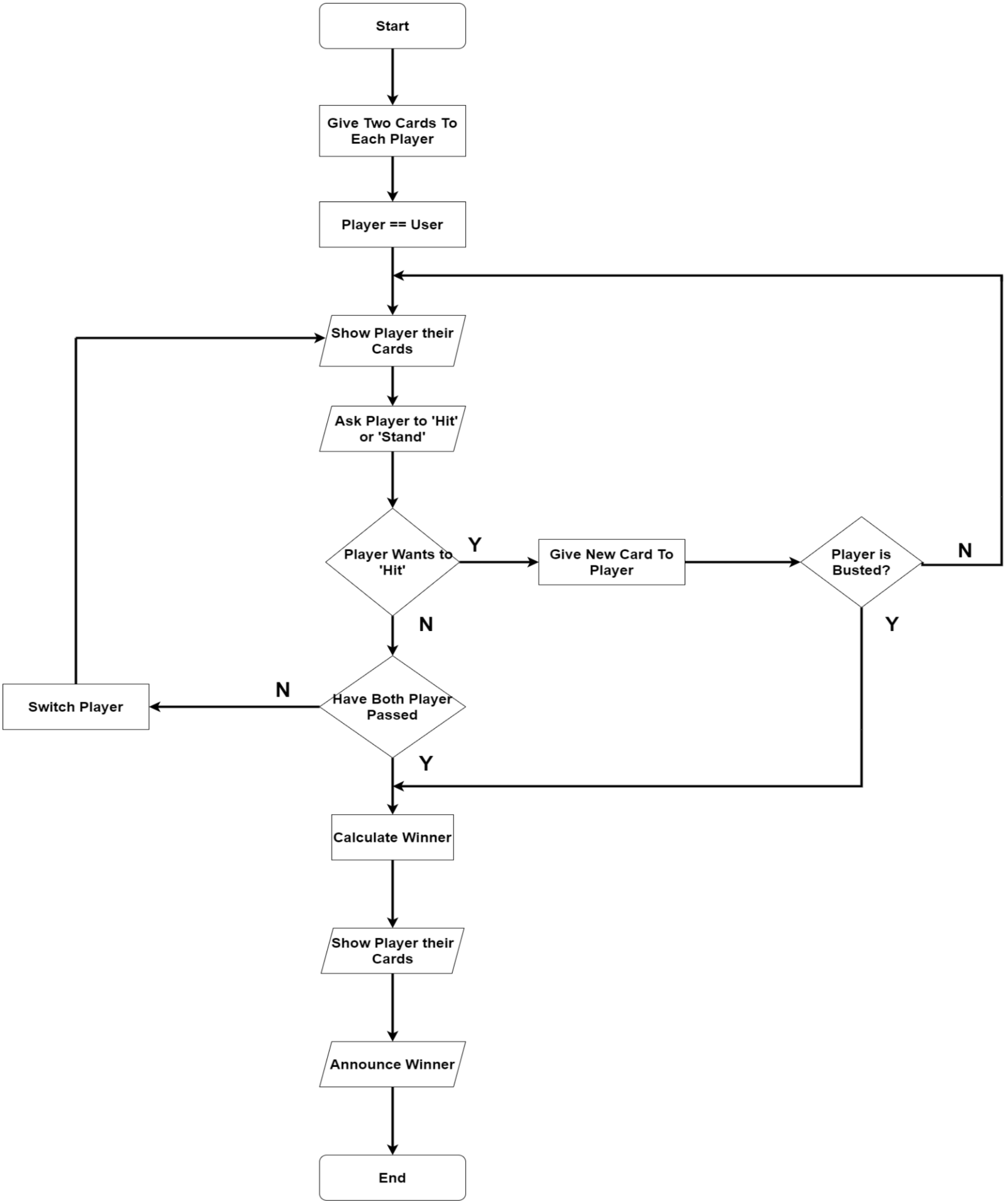
It is simpler, more enjoyable and less-resource/time consuming than other games available in the same genre. It is very quick, responsive and user-friendly.

The proposed BlackJack Project (BlackJack Game) will take on off-line casinos and other games associated with BlackJack. It will take care of all the Player resources without requiring any Player Interaction. Player is not supposed to get into the details of underlying software technicalities and its model (abstraction).

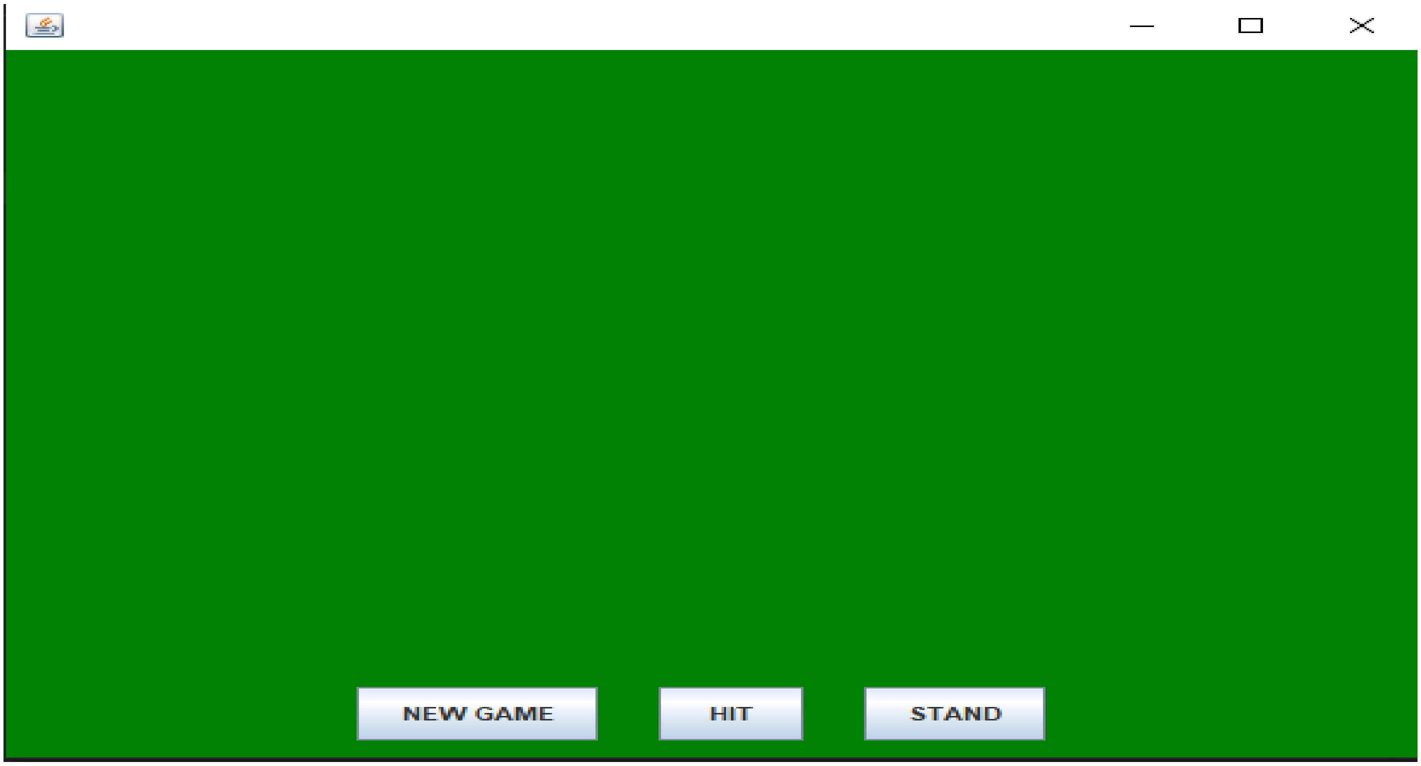
Flow of Data in BlackJack Project (BlackJack Game)


Start
Generate Hand object
Call Deal function 2x
Deal Function makes card objects
Card -number(1-13) -suit(1-4)
Hand -2 Cards
Game (Later)
Function: Interpret Value 1=ace=11 2-10=face val. 11-13=10pts
Function: Interpret Suit 1=clubs 2=diamonds 3=hearts 4=spades

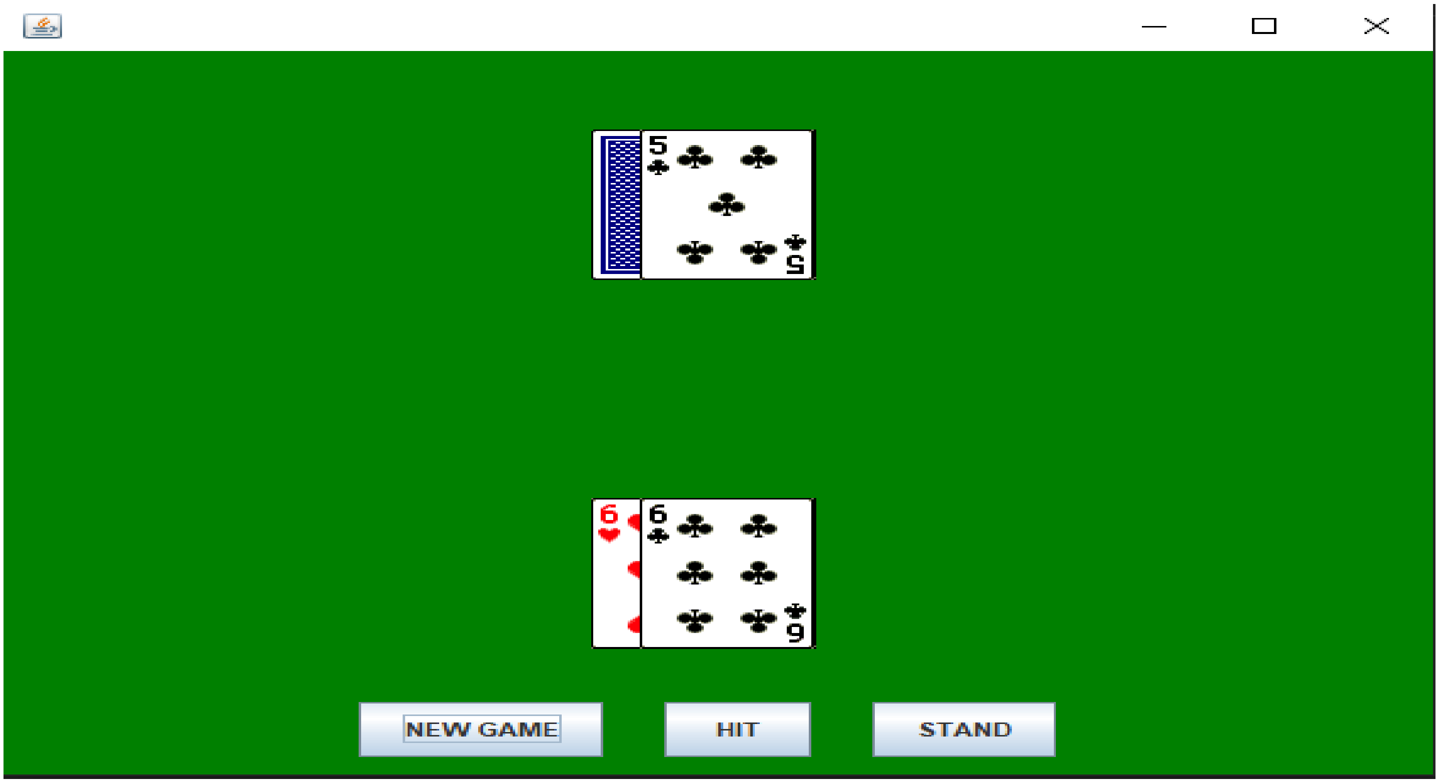
# Flow Chart of BlackJack Project (BlackJack Game)



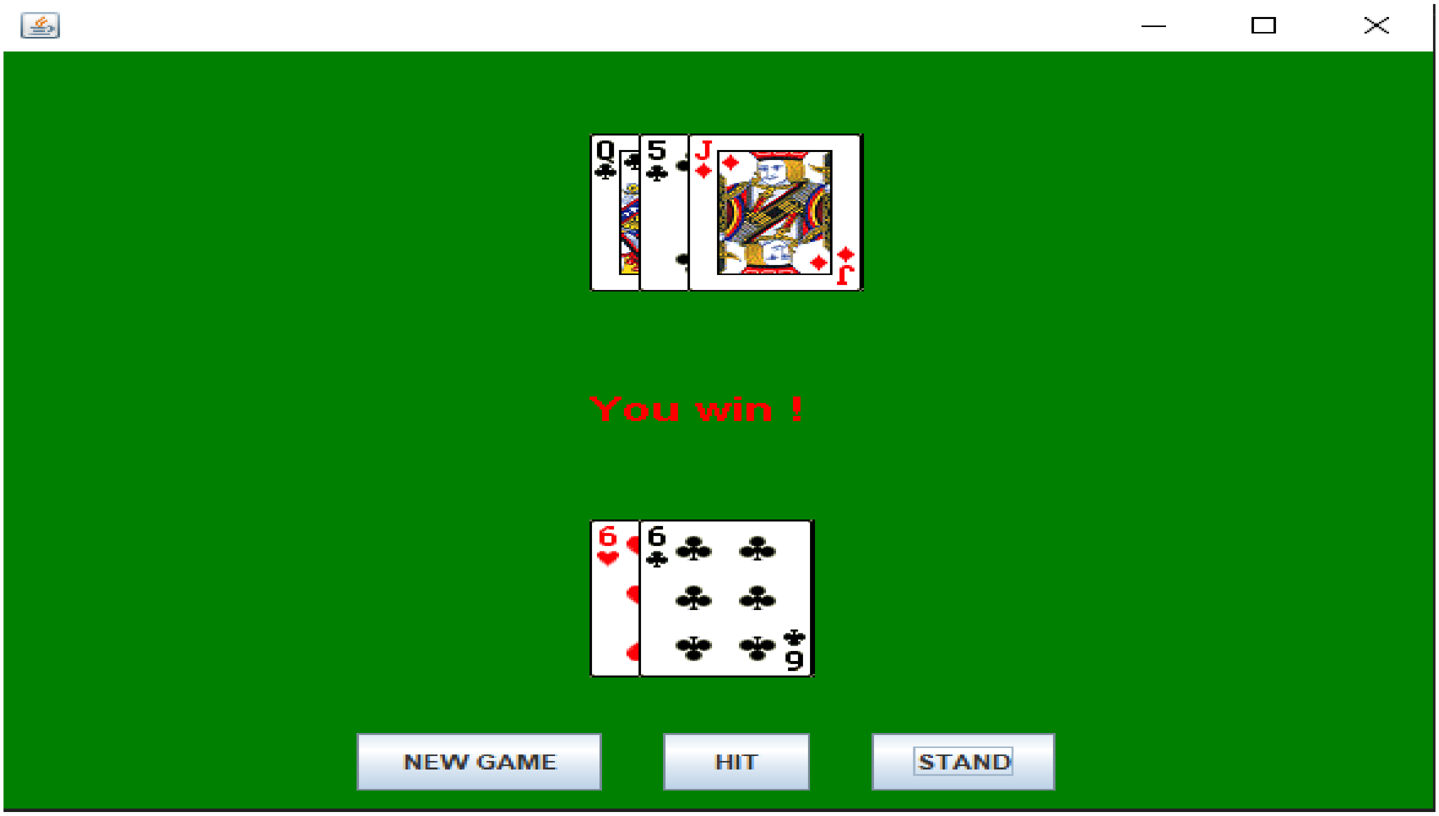
# SNAPSHOTS of PROJECT (GUI)



Home Screen

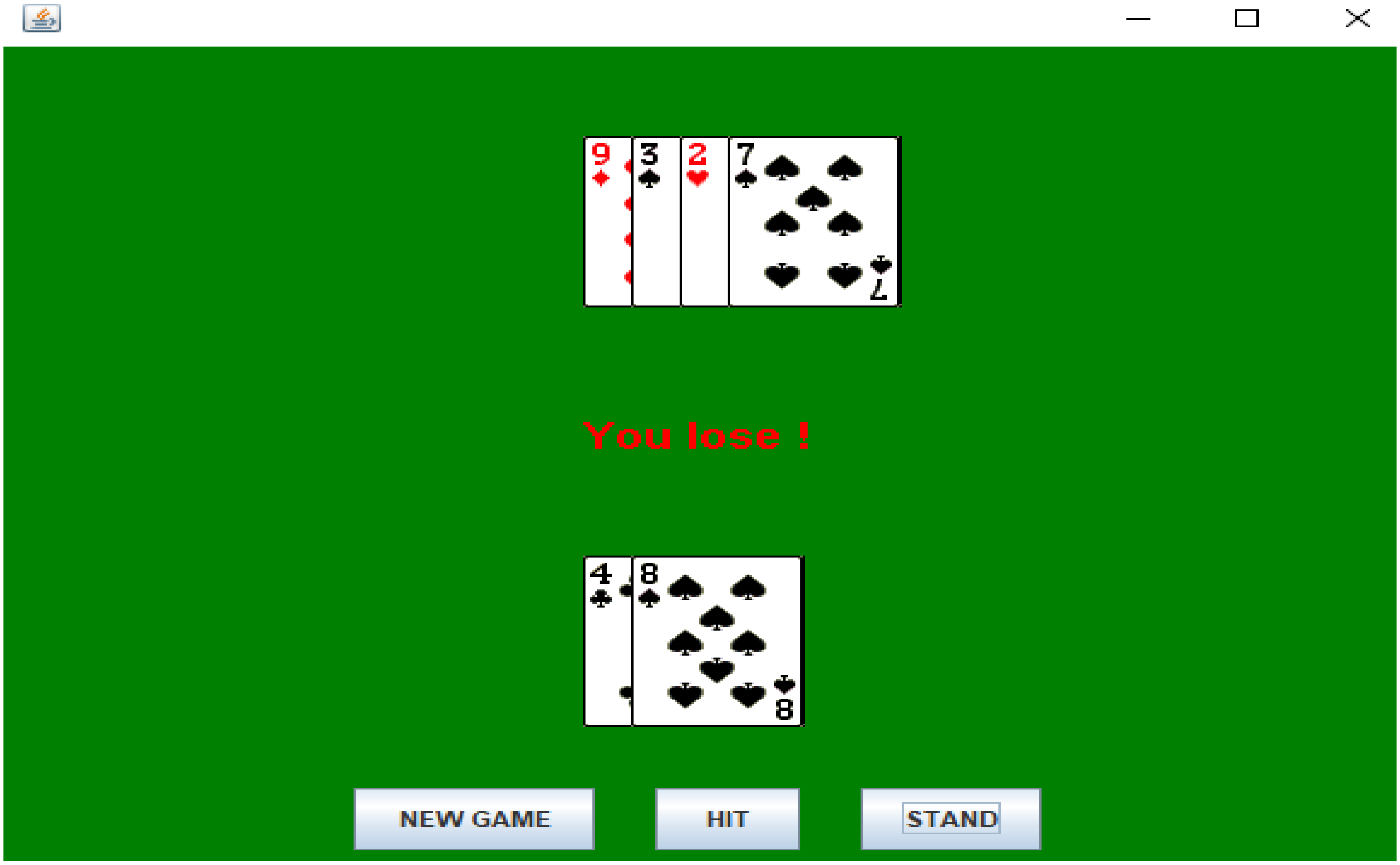


Active Game Screen



You win !

Win screen

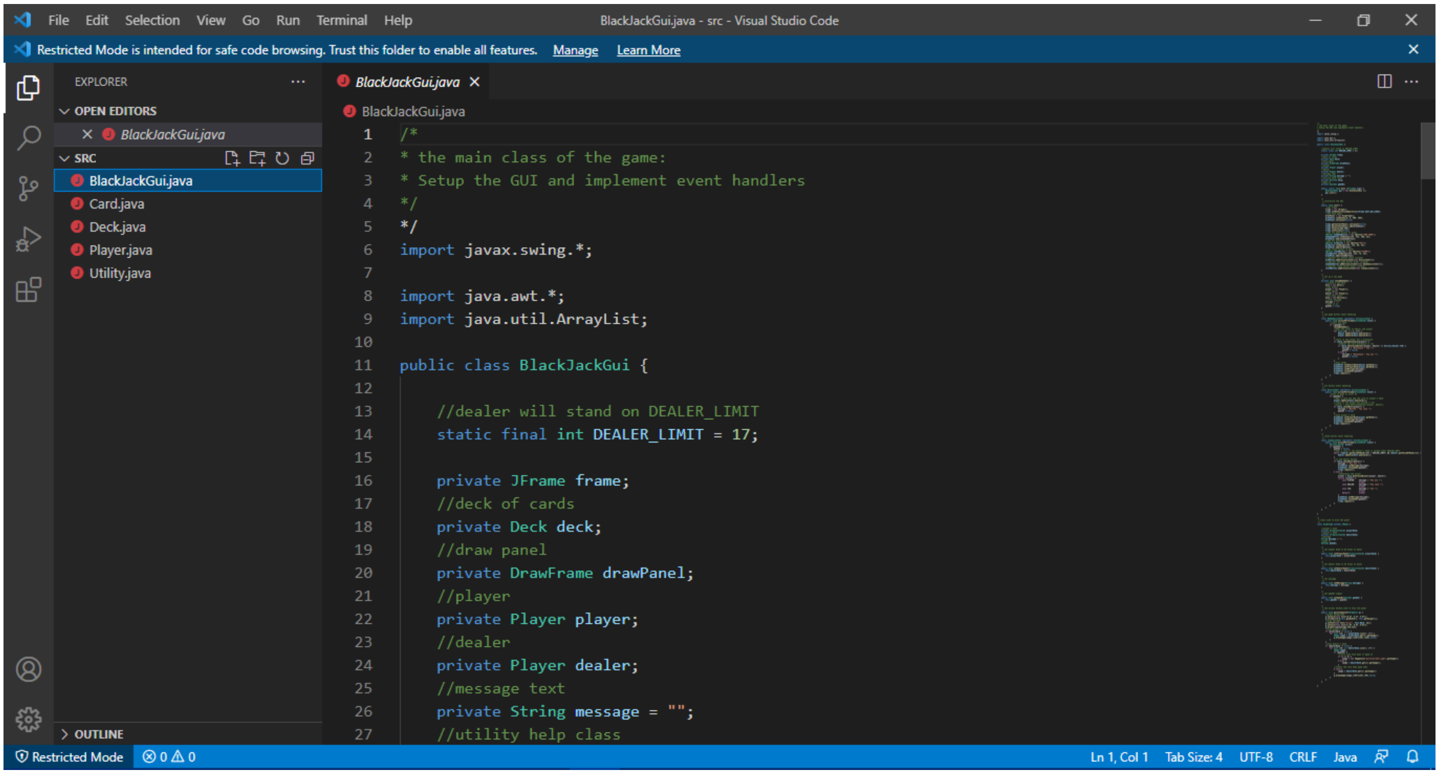


You lose !

Lose Screen

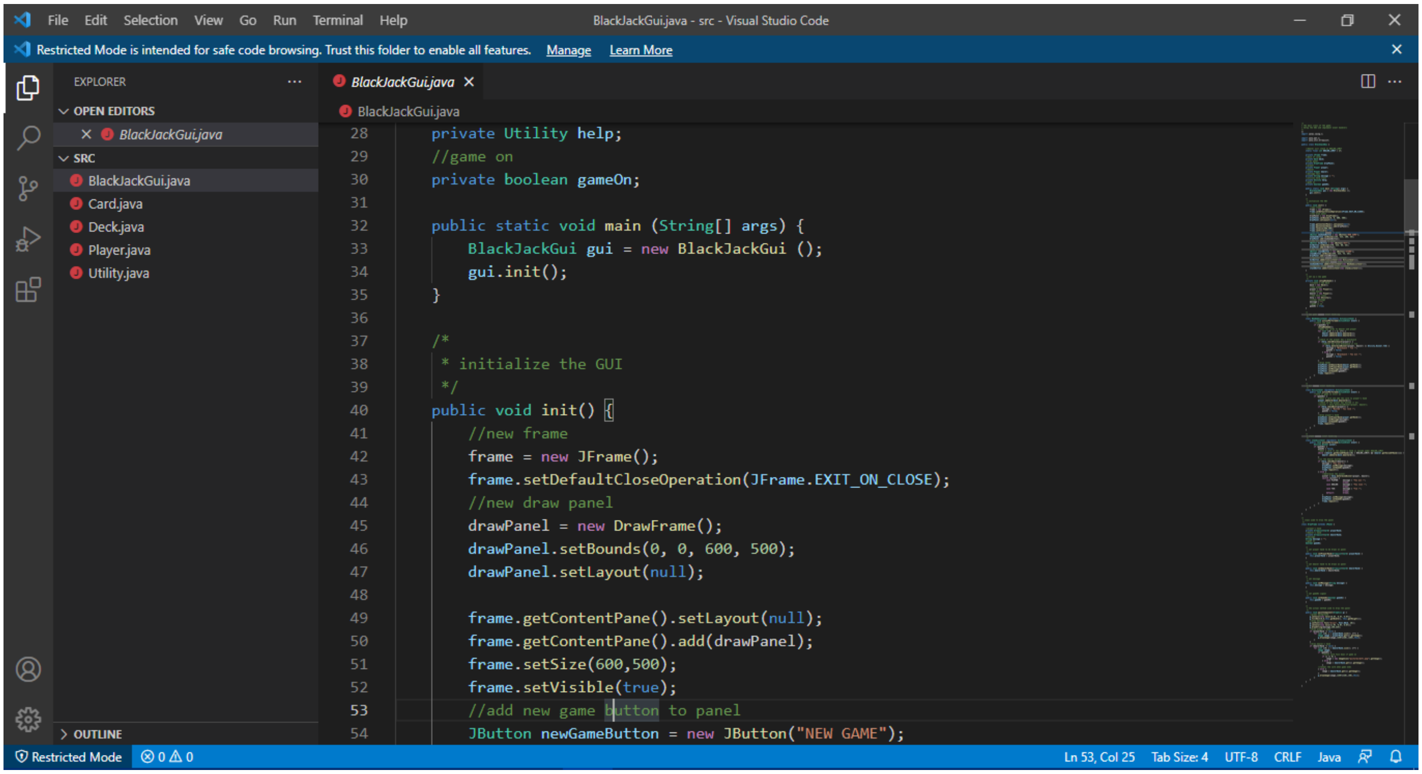
# CODING

## BlackJackGUI.java



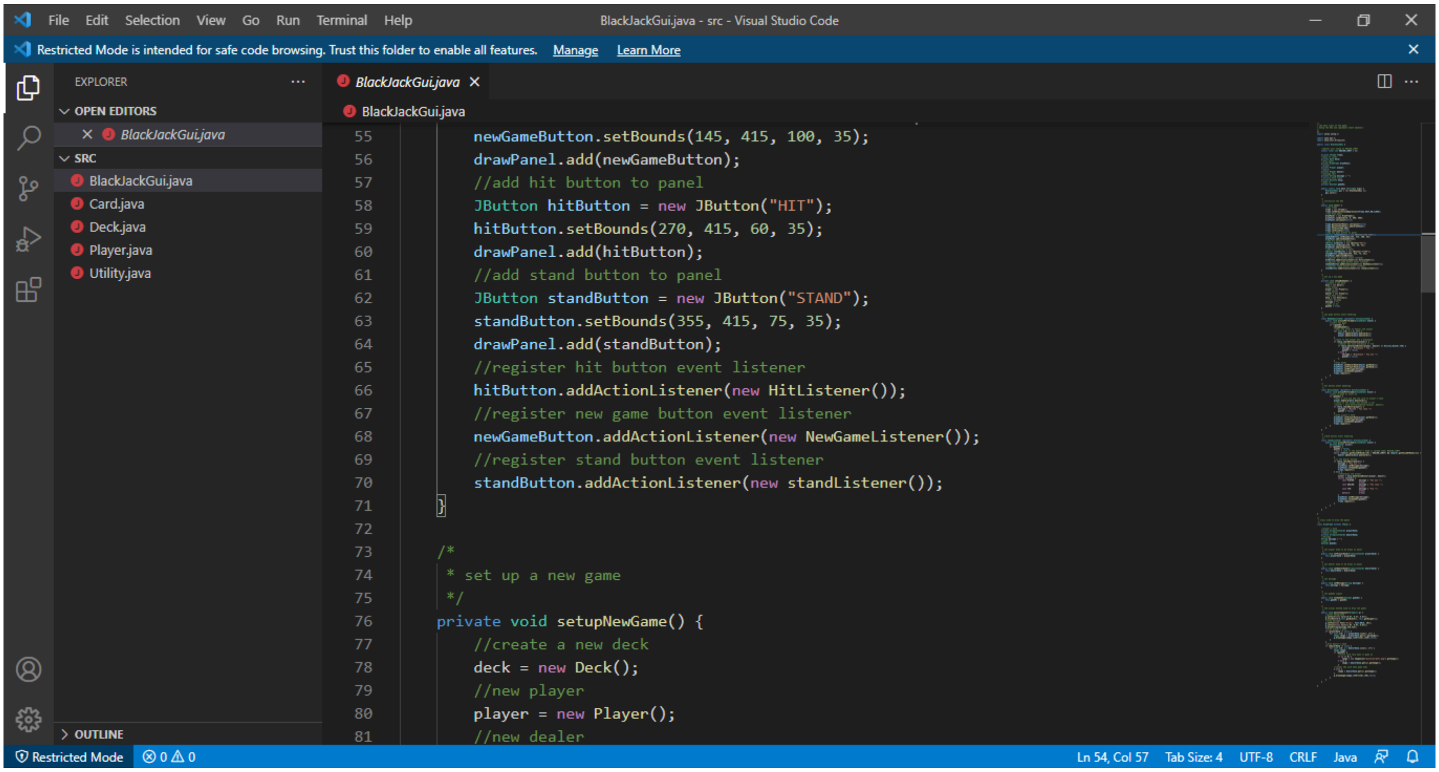
Restricted Mode is intended for safe code browsing. Trust this folder to enable all features.

```java
/*
* the main class of the game:
* Setup the GUI and implement event handlers
*/
*/
import javax.swing.*;

import java.awt.*;
import java.util.ArrayList;

public class BlackJackGui {

    //dealer will stand on DEALER_LIMIT
    static final int DEALER_LIMIT = 17;

    private JFrame frame;
    //deck of cards
    private Deck deck;
    //draw panel
    private DrawFrame drawPanel;
    //player
    private Player player;
    //dealer
    private Player dealer;
    //message text
    private String message = "";
    //utility help class
```

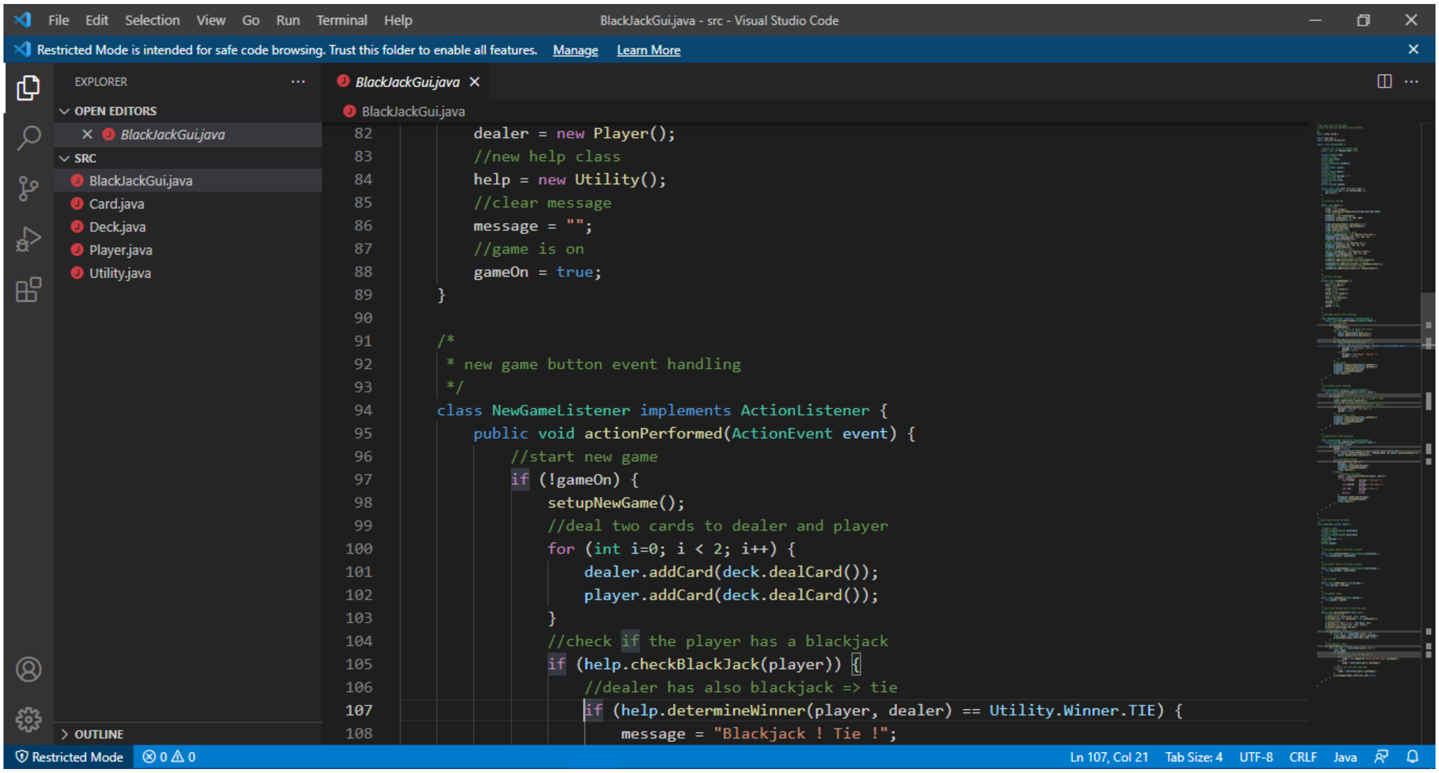


Restricted Mode is intended for safe code browsing. Trust this folder to enable all features.

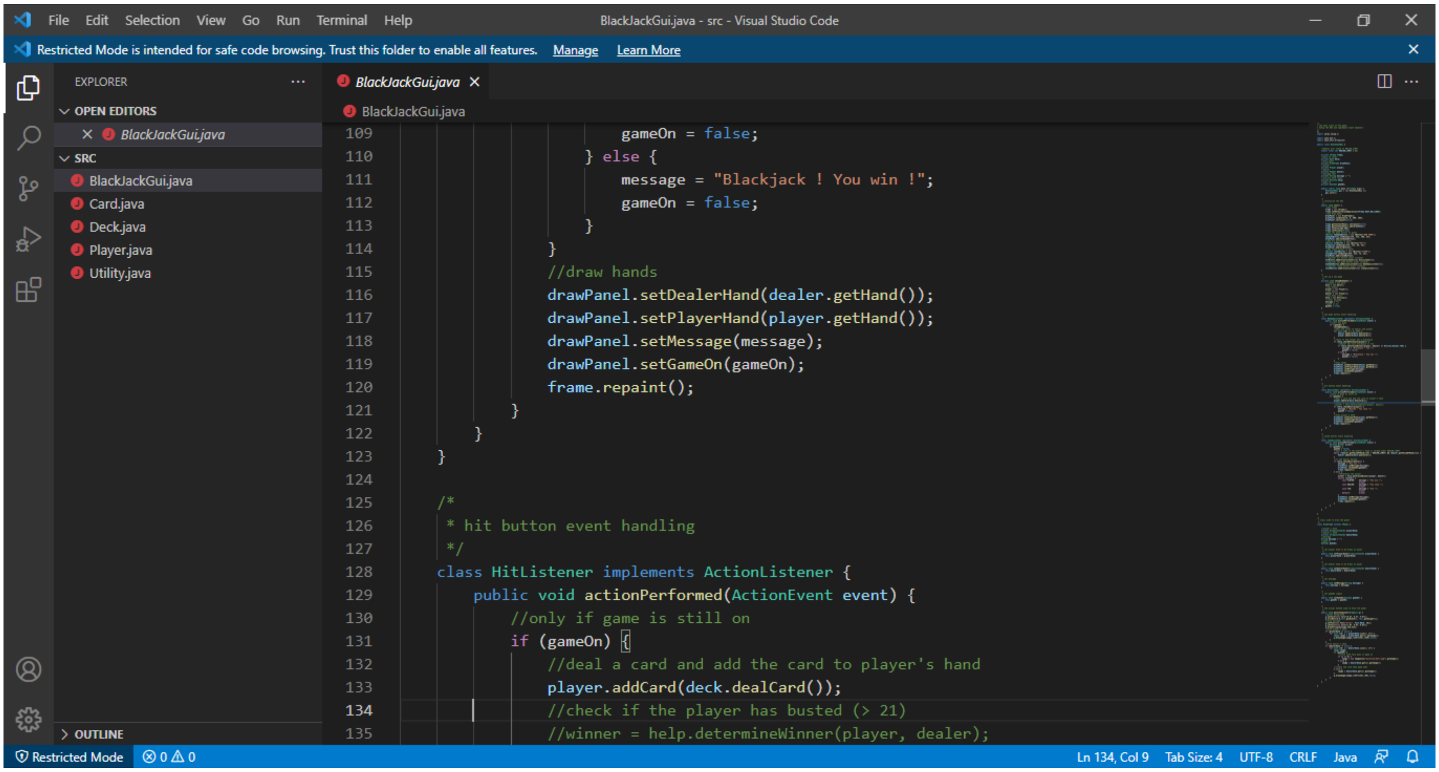
```java
    private Utility help;
    //game on
    private boolean gameOn;

    public static void main (String[] args) {
        BlackJackGui gui = new BlackJackGui ();
        gui.init();
    }

    /*
     * initialize the GUI
     */
    public void init() {
        //new frame
        frame = new JFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //new draw panel
        drawPanel = new DrawFrame();
        drawPanel.setBounds(0, 0, 600, 500);
        drawPanel.setLayout(null);

        frame.getContentPane().setLayout(null);
        frame.getContentPane().add(drawPanel);
        frame.setSize(600,500);
        frame.setVisible(true);
        //add new game button to panel
        JButton newGameButton = new JButton("NEW GAME");
```


File Edit Selection View Go Run Terminal Help
BlackJackGui.java - src - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this folder to enable all features. Manage Learn More
EXPLORER
OPEN EDITORS
BlackJackGui.java
SRC
BlackJackGui.java
Card.java
Deck.java
Player.java
Utility.java
OUTLINE
Restricted Mode ⊗ 0 ⚠ 0
Ln 54, Col 57 Tab Size: 4 UTF-8 CRLF Java

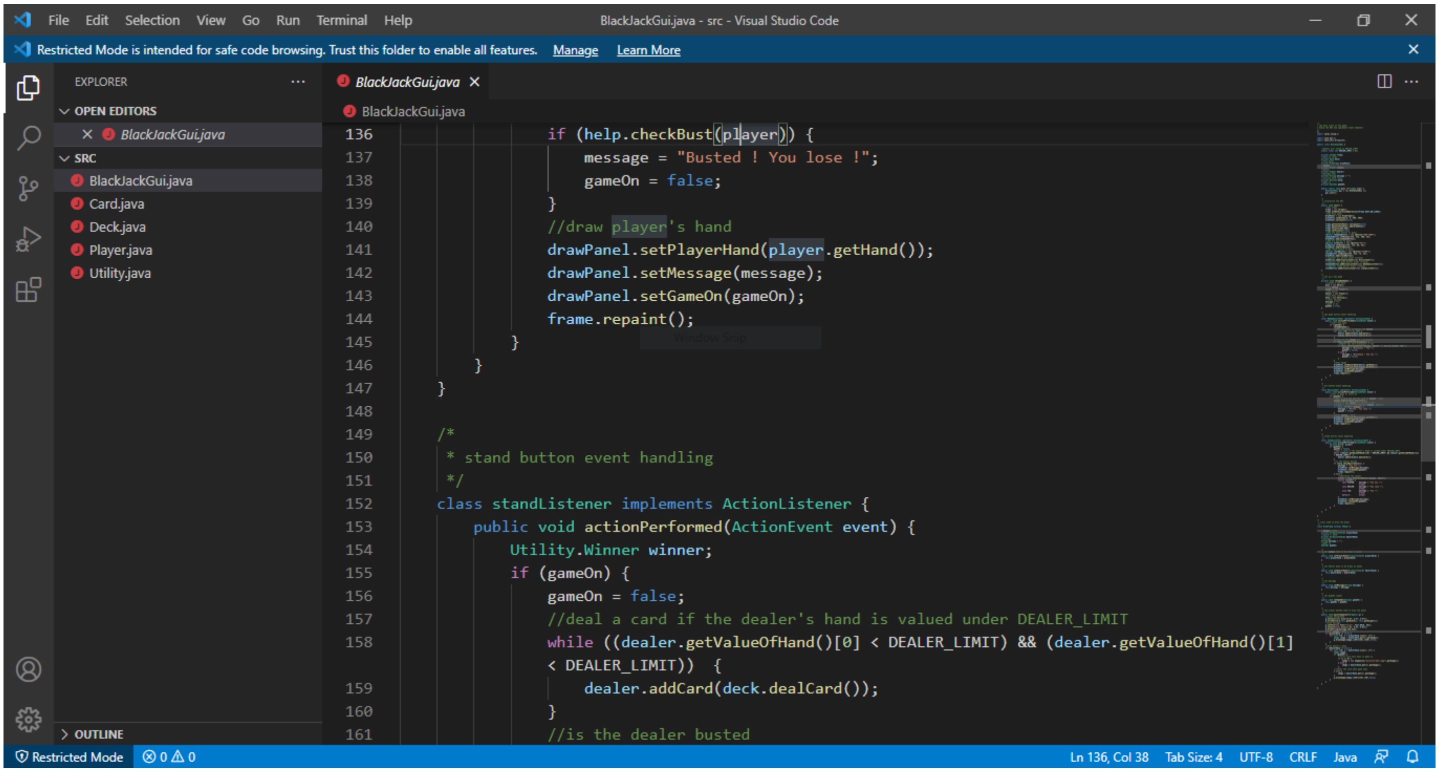
```java
        newGameButton.setBounds(145, 415, 100, 35);
        drawPanel.add(newGameButton);
        //add hit button to panel
        JButton hitButton = new JButton("HIT");
        hitButton.setBounds(270, 415, 60, 35);
        drawPanel.add(hitButton);
        //add stand button to panel
        JButton standButton = new JButton("STAND");
        standButton.setBounds(355, 415, 75, 35);
        drawPanel.add(standButton);
        //register hit button event listener
        hitButton.addActionListener(new HitListener());
        //register new game button event listener
        newGameButton.addActionListener(new NewGameListener());
        //register stand button event listener
        standButton.addActionListener(new standListener());
    }

    /*
     * set up a new game
     */
    private void setupNewGame() {
        //create a new deck
        deck = new Deck();
        //new player
        player = new Player();
        //new dealer
```


File Edit Selection View Go Run Terminal Help
BlackJackGui.java - src - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this folder to enable all features. Manage Learn More
EXPLORER
OPEN EDITORS
BlackJackGui.java
SRC
BlackJackGui.java
Card.java
Deck.java
Player.java
Utility.java
OUTLINE
Restricted Mode ⊗ 0 ⚠ 0
Ln 107, Col 21 Tab Size: 4 UTF-8 CRLF Java

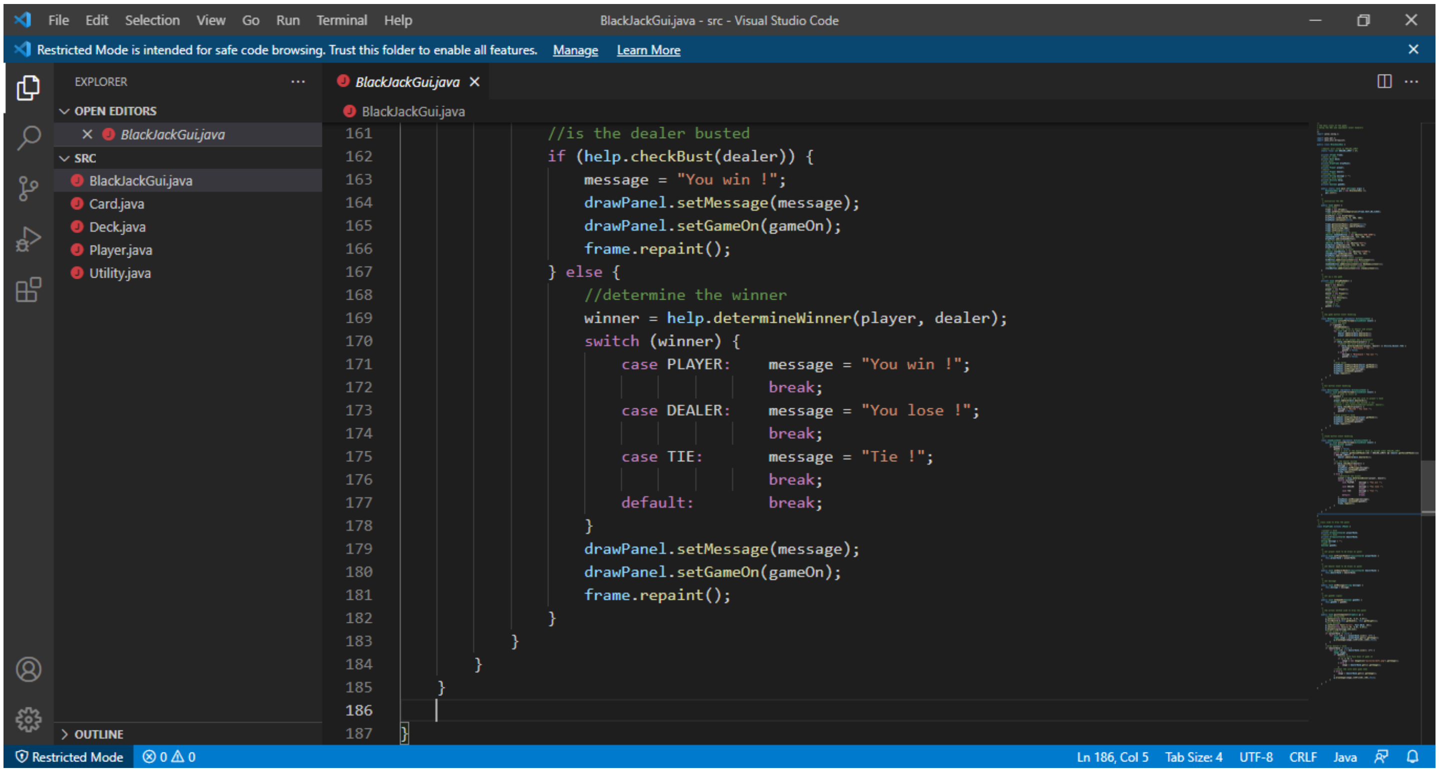
```java
        dealer = new Player();
        //new help class
        help = new Utility();
        //clear message
        message = "";
        //game is on
        gameOn = true;
    }

    /*
     * new game button event handling
     */
    class NewGameListener implements ActionListener {
        public void actionPerformed(ActionEvent event) {
            //start new game
            if (!gameOn) {
                setupNewGame();
                //deal two cards to dealer and player
                for (int i=0; i < 2; i++) {
                    dealer.addCard(deck.dealCard());
                    player.addCard(deck.dealCard());
                }
                //check if the player has a blackjack
                if (help.checkBlackJack(player)) {
                    //dealer has also blackjack => tie
                    if (help.determineWinner(player, dealer) == Utility.Winner.TIE) {
                        message = "Blackjack ! Tie !";
```



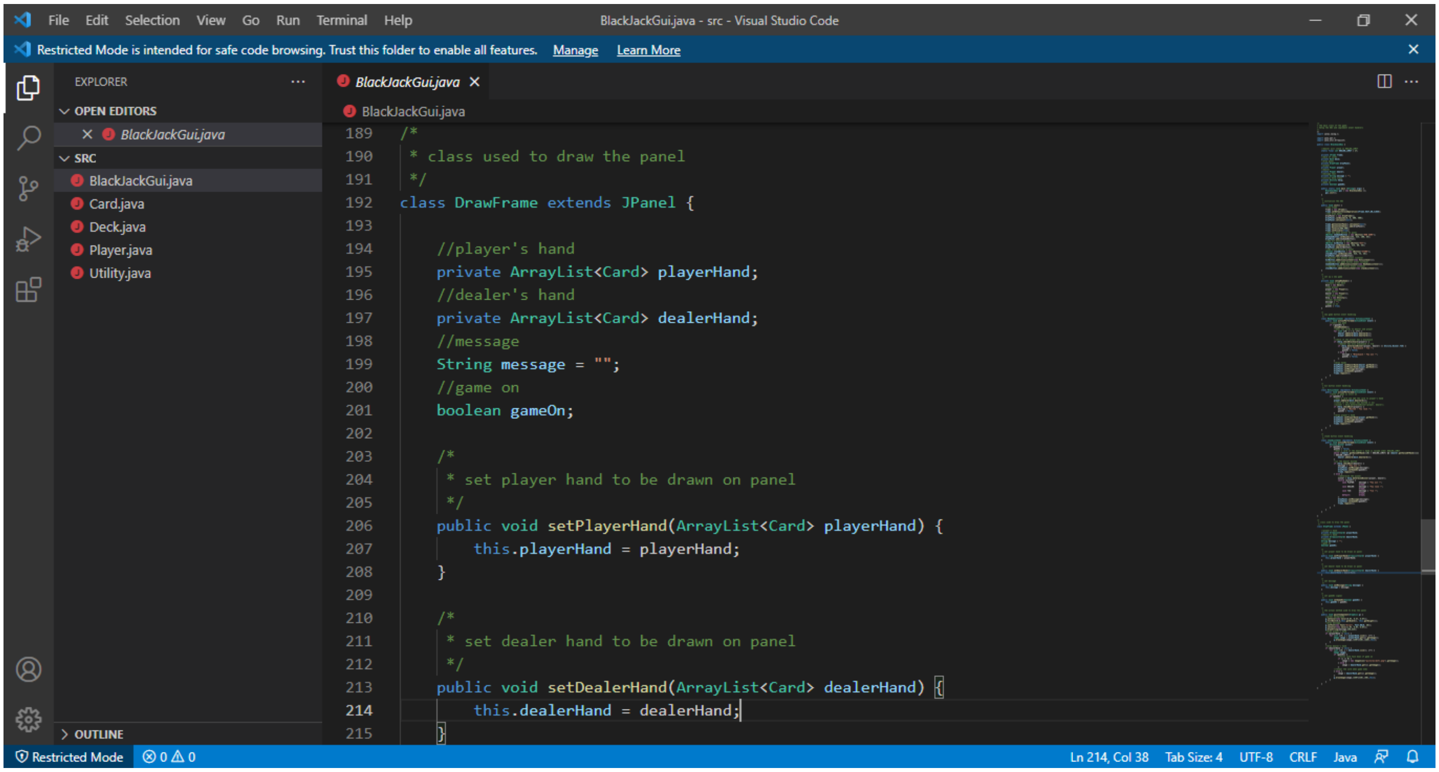
```java
                        gameOn = false;
                    } else {
                        message = "Blackjack ! You win !";
                        gameOn = false;
                    }
                }
                //draw hands
                drawPanel.setDealerHand(dealer.getHand());
                drawPanel.setPlayerHand(player.getHand());
                drawPanel.setMessage(message);
                drawPanel.setGameOn(gameOn);
                frame.repaint();
            }
        }
    }

    /*
     * hit button event handling
     */
    class HitListener implements ActionListener {
        public void actionPerformed(ActionEvent event) {
            //only if game is still on
            if (gameOn) {
                //deal a card and add the card to player's hand
                player.addCard(deck.dealCard());
                //check if the player has busted (> 21)
                //winner = help.determineWinner(player, dealer);
```



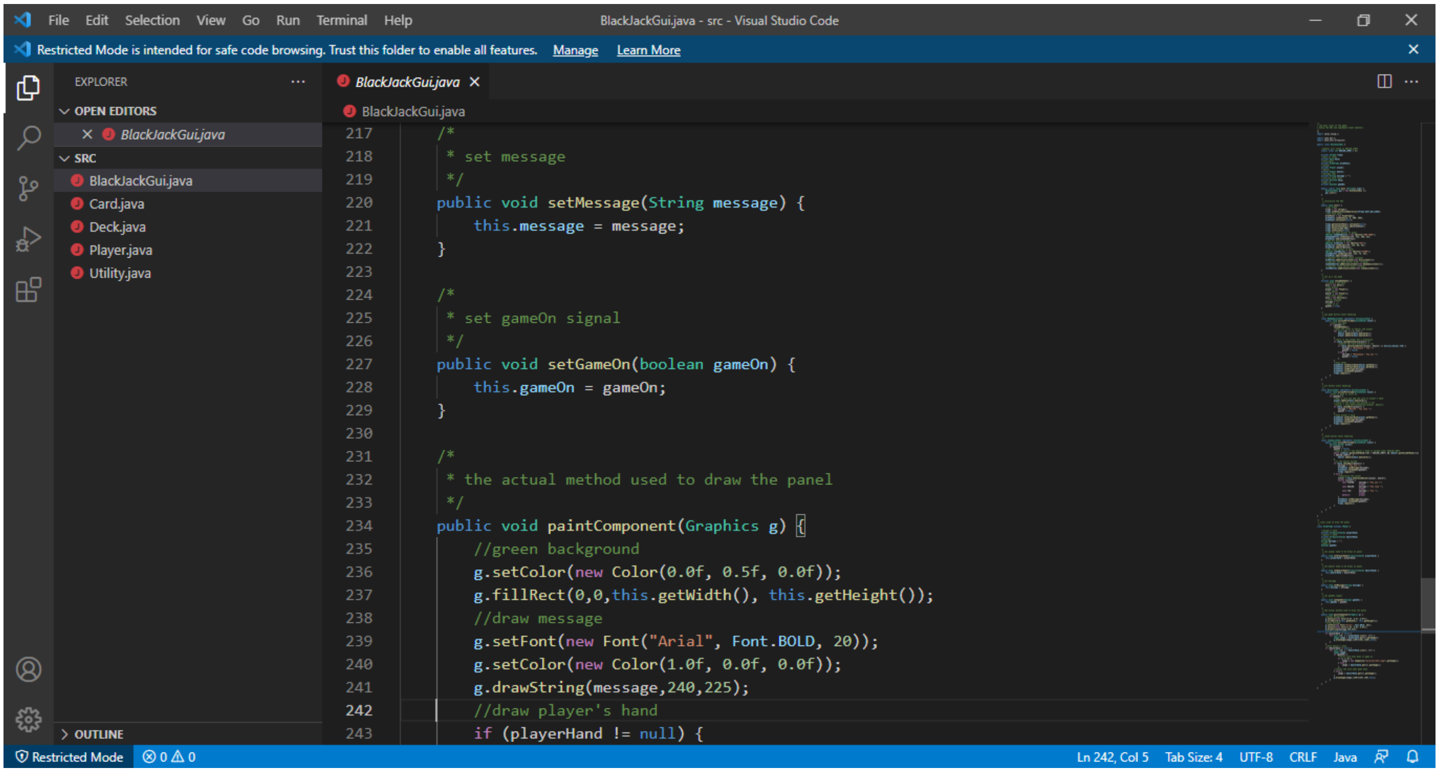
```java
                if (help.checkBust(player)) {
                    message = "Busted ! You lose !";
                    gameOn = false;
                }
                //draw player's hand
                drawPanel.setPlayerHand(player.getHand());
                drawPanel.setMessage(message);
                drawPanel.setGameOn(gameOn);
                frame.repaint();
            }
        }
    }

    /*
     * stand button event handling
     */
    class standListener implements ActionListener {
        public void actionPerformed(ActionEvent event) {
            Utility.Winner winner;
            if (gameOn) {
                gameOn = false;
                //deal a card if the dealer's hand is valued under DEALER_LIMIT
                while ((dealer.getValueOfHand()[0] < DEALER_LIMIT) && (dealer.getValueOfHand()[1] < DEALER_LIMIT))  {
                    dealer.addCard(deck.dealCard());
                }
                //is the dealer busted
```


File Edit Selection View Go Run Terminal Help
BlackJackGui.java - src - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this folder to enable all features. Manage Learn More
EXPLORER
OPEN EDITORS
BlackJackGui.java
SRC
BlackJackGui.java
Card.java
Deck.java
Player.java
Utility.java
OUTLINE
Restricted Mode ⊗ 0 ⚠ 0
Ln 186, Col 5 Tab Size: 4 UTF-8 CRLF Java

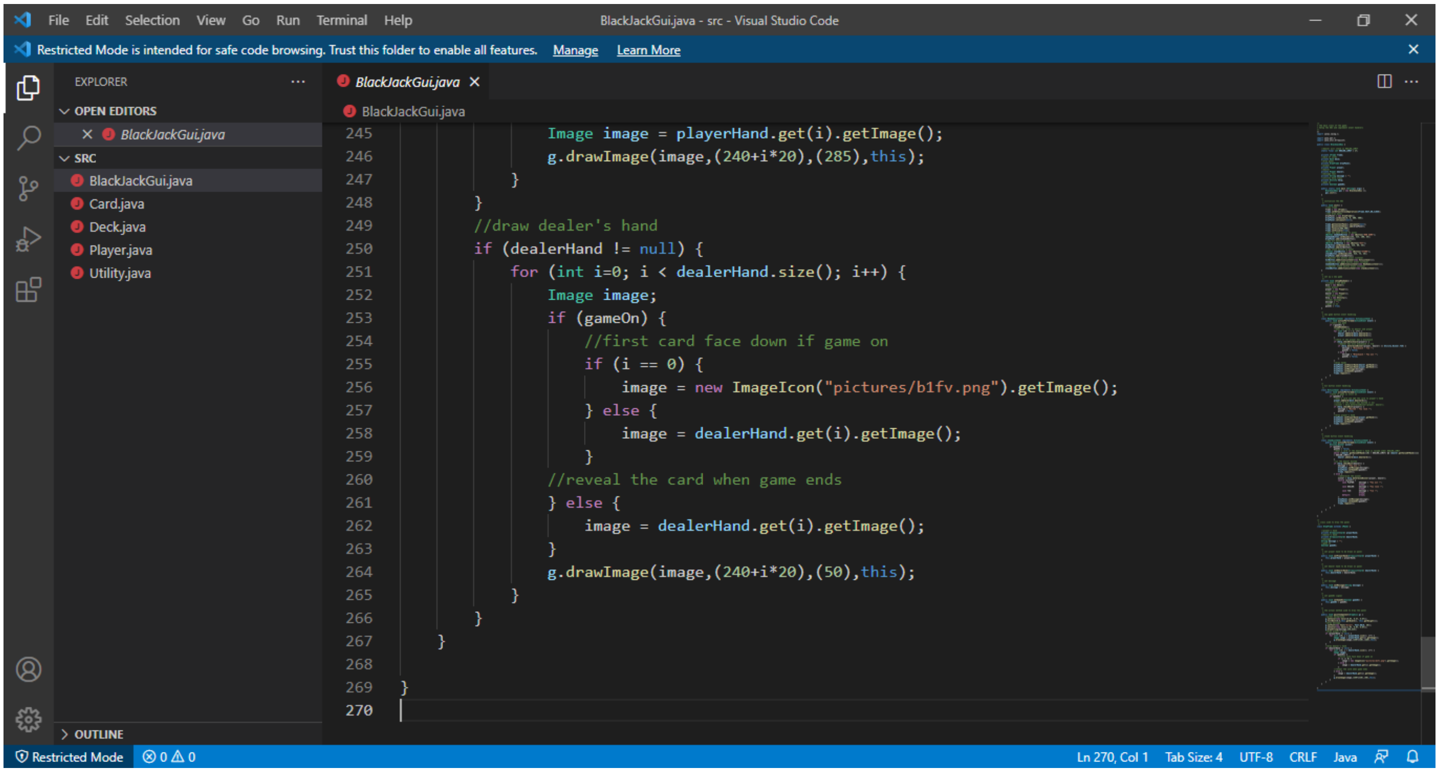
```java
                //is the dealer busted
                if (help.checkBust(dealer)) {
                    message = "You win !";
                    drawPanel.setMessage(message);
                    drawPanel.setGameOn(gameOn);
                    frame.repaint();
                } else {
                    //determine the winner
                    winner = help.determineWinner(player, dealer);
                    switch (winner) {
                        case PLAYER:    message = "You win !";
                                        break;
                        case DEALER:    message = "You lose !";
                                        break;
                        case TIE:       message = "Tie !";
                                        break;
                        default:        break;
                    }
                    drawPanel.setMessage(message);
                    drawPanel.setGameOn(gameOn);
                    frame.repaint();
                }
            }
        }
    }
    
}
```


File Edit Selection View Go Run Terminal Help
BlackJackGui.java - src - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this folder to enable all features. Manage Learn More
EXPLORER
OPEN EDITORS
BlackJackGui.java
SRC
BlackJackGui.java
Card.java
Deck.java
Player.java
Utility.java
OUTLINE
Restricted Mode ⊗ 0 ⚠ 0
Ln 214, Col 38 Tab Size: 4 UTF-8 CRLF Java

```java
/*
 * class used to draw the panel
 */
class DrawFrame extends JPanel {

    //player's hand
    private ArrayList<Card> playerHand;
    //dealer's hand
    private ArrayList<Card> dealerHand;
    //message
    String message = "";
    //game on
    boolean gameOn;

    /*
     * set player hand to be drawn on panel
     */
    public void setPlayerHand(ArrayList<Card> playerHand) {
        this.playerHand = playerHand;
    }

    /*
     * set dealer hand to be drawn on panel
     */
    public void setDealerHand(ArrayList<Card> dealerHand) {
        this.dealerHand = dealerHand;
    }
```

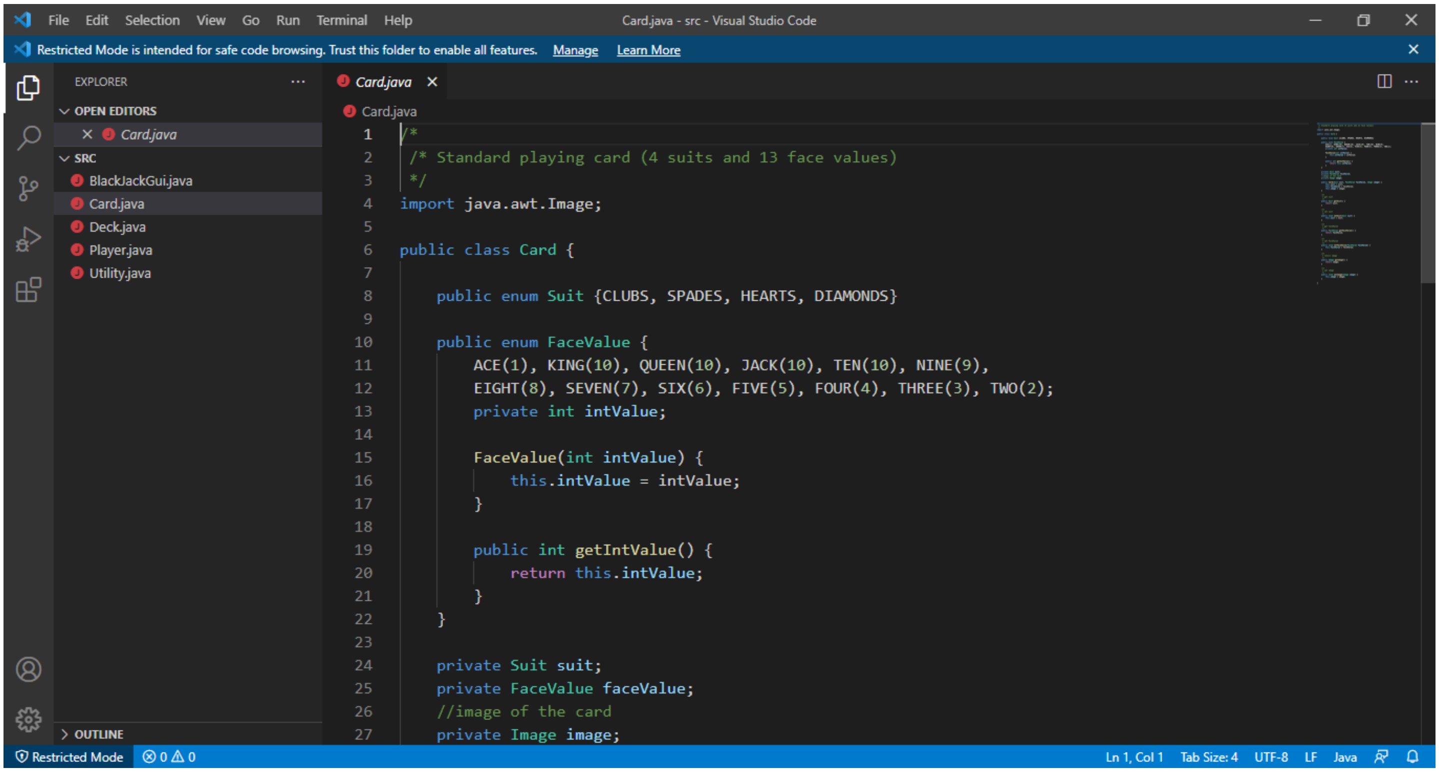

File Edit Selection View Go Run Terminal Help
BlackJackGui.java - src - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this folder to enable all features. Manage Learn More
EXPLORER
OPEN EDITORS
BlackJackGui.java
SRC
BlackJackGui.java
Card.java
Deck.java
Player.java
Utility.java
OUTLINE
Restricted Mode
Ln 242, Col 5 Tab Size: 4 UTF-8 CRLF Java

```java
    /*
     * set message
     */
    public void setMessage(String message) {
        this.message = message;
    }

    /*
     * set gameOn signal
     */
    public void setGameOn(boolean gameOn) {
        this.gameOn = gameOn;
    }

    /*
     * the actual method used to draw the panel
     */
    public void paintComponent(Graphics g) {
        //green background
        g.setColor(new Color(0.0f, 0.5f, 0.0f));
        g.fillRect(0,0,this.getWidth(), this.getHeight());
        //draw message
        g.setFont(new Font("Arial", Font.BOLD, 20));
        g.setColor(new Color(1.0f, 0.0f, 0.0f));
        g.drawString(message,240,225);
        //draw player's hand
        if (playerHand != null) {
```

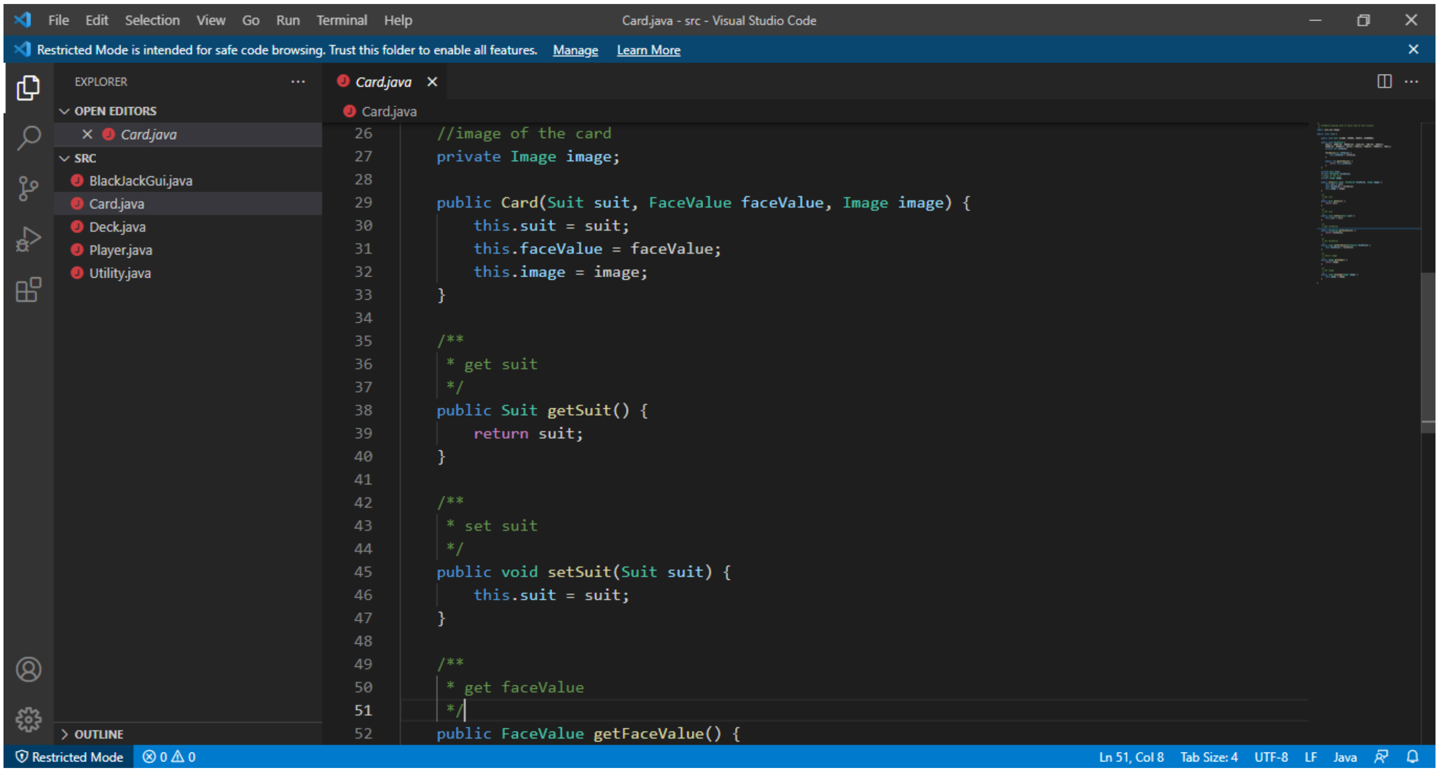

File Edit Selection View Go Run Terminal Help
BlackJackGui.java - src - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this folder to enable all features. Manage Learn More
EXPLORER
OPEN EDITORS
BlackJackGui.java
SRC
BlackJackGui.java
Card.java
Deck.java
Player.java
Utility.java
OUTLINE
Restricted Mode
Ln 270, Col 1 Tab Size: 4 UTF-8 CRLF Java

```java
                Image image = playerHand.get(i).getImage();
                g.drawImage(image,(240+i*20),(285),this);
            }
        }
        //draw dealer's hand
        if (dealerHand != null) {
            for (int i=0; i < dealerHand.size(); i++) {
                Image image;
                if (gameOn) {
                    //first card face down if game on
                    if (i == 0) {
                        image = new ImageIcon("pictures/b1fv.png").getImage();
                    } else {
                        image = dealerHand.get(i).getImage();
                    }
                //reveal the card when game ends
                } else {
                    image = dealerHand.get(i).getImage();
                }
                g.drawImage(image,(240+i*20),(50),this);
            }
        }
    }
}
```

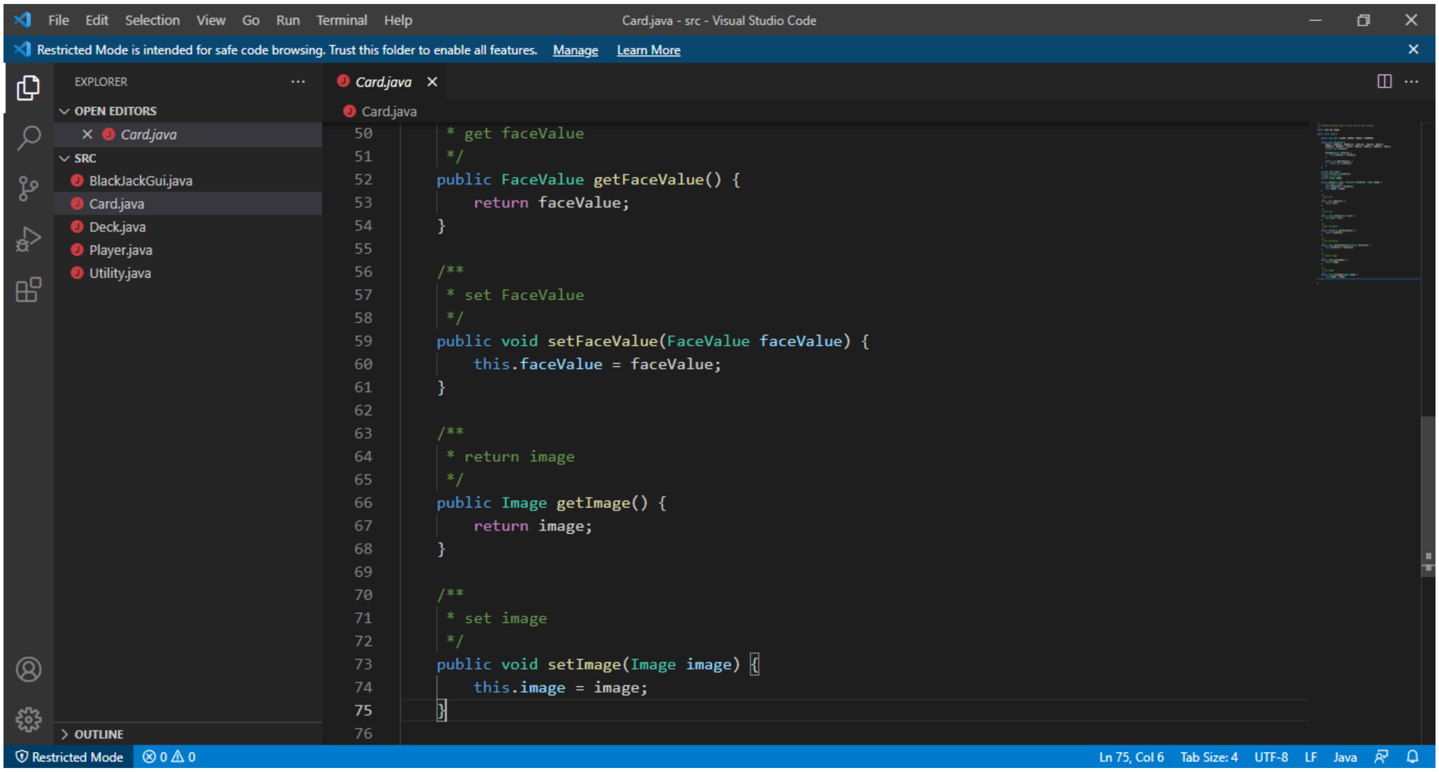
# Card.java


File Edit Selection View Go Run Terminal Help
Card.java - src - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this folder to enable all features. Manage Learn More
EXPLORER
OPEN EDITORS
Card.java
SRC
BlackJackGui.java
Card.java
Deck.java
Player.java
Utility.java
OUTLINE
Restricted Mode
⊗ 0 ⚠ 0
Ln 1, Col 1 Tab Size: 4 UTF-8 LF Java

```java
/*
 /* Standard playing card (4 suits and 13 face values)
 */
import java.awt.Image;

public class Card {

    public enum Suit {CLUBS, SPADES, HEARTS, DIAMONDS}

    public enum FaceValue {
        ACE(1), KING(10), QUEEN(10), JACK(10), TEN(10), NINE(9),
        EIGHT(8), SEVEN(7), SIX(6), FIVE(5), FOUR(4), THREE(3), TWO(2);
        private int intValue;

        FaceValue(int intValue) {
            this.intValue = intValue;
        }

        public int getIntValue() {
            return this.intValue;
        }
    }

    private Suit suit;
    private FaceValue faceValue;
    //image of the card
    private Image image;
```

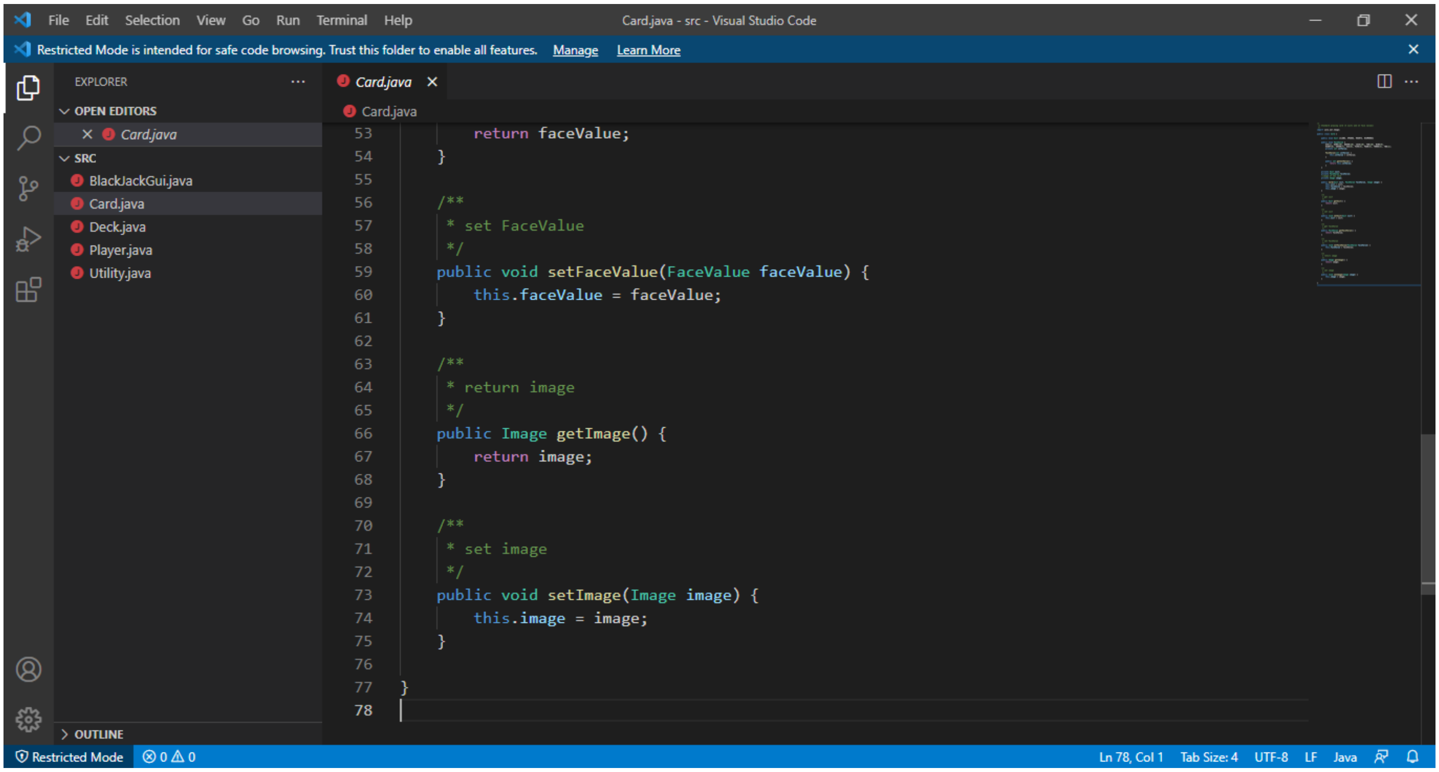

File Edit Selection View Go Run Terminal Help
Card.java - src - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this folder to enable all features. Manage Learn More
EXPLORER
OPEN EDITORS
Card.java
SRC
BlackJackGui.java
Card.java
Deck.java
Player.java
Utility.java
OUTLINE
Restricted Mode
⊗ 0 ⚠ 0
Ln 51, Col 8 Tab Size: 4 UTF-8 LF Java

```java
    //image of the card
    private Image image;

    public Card(Suit suit, FaceValue faceValue, Image image) {
        this.suit = suit;
        this.faceValue = faceValue;
        this.image = image;
    }

    /**
     * get suit
     */
    public Suit getSuit() {
        return suit;
    }

    /**
     * set suit
     */
    public void setSuit(Suit suit) {
        this.suit = suit;
    }

    /**
     * get faceValue
     */
    public FaceValue getFaceValue() {
```



Restricted Mode is intended for safe code browsing. Trust this folder to enable all features.

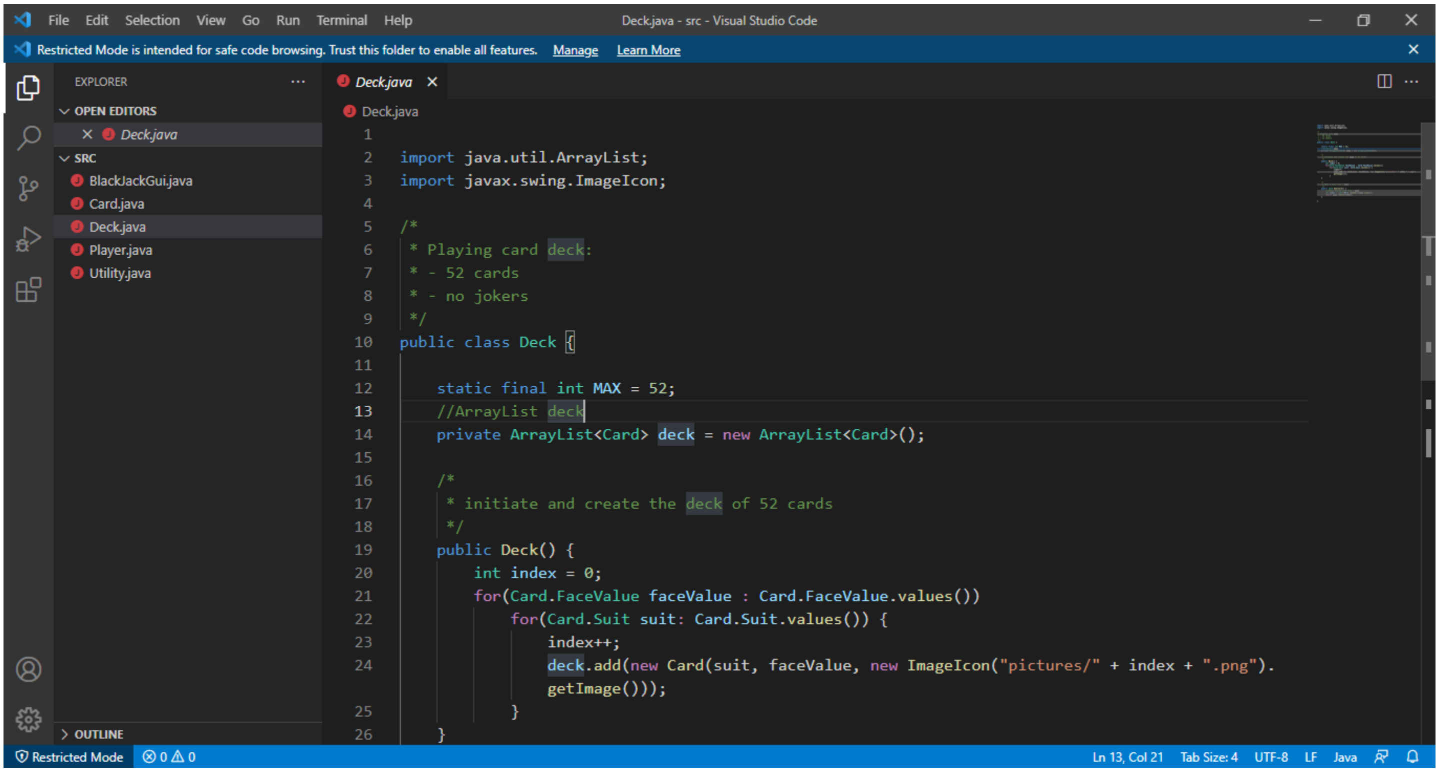
```java
     * get faceValue
     */
    public FaceValue getFaceValue() {
        return faceValue;
    }

    /**
     * set FaceValue
     */
    public void setFaceValue(FaceValue faceValue) {
        this.faceValue = faceValue;
    }

    /**
     * return image
     */
    public Image getImage() {
        return image;
    }

    /**
     * set image
     */
    public void setImage(Image image) {
        this.image = image;
    }

```



Restricted Mode is intended for safe code browsing. Trust this folder to enable all features.

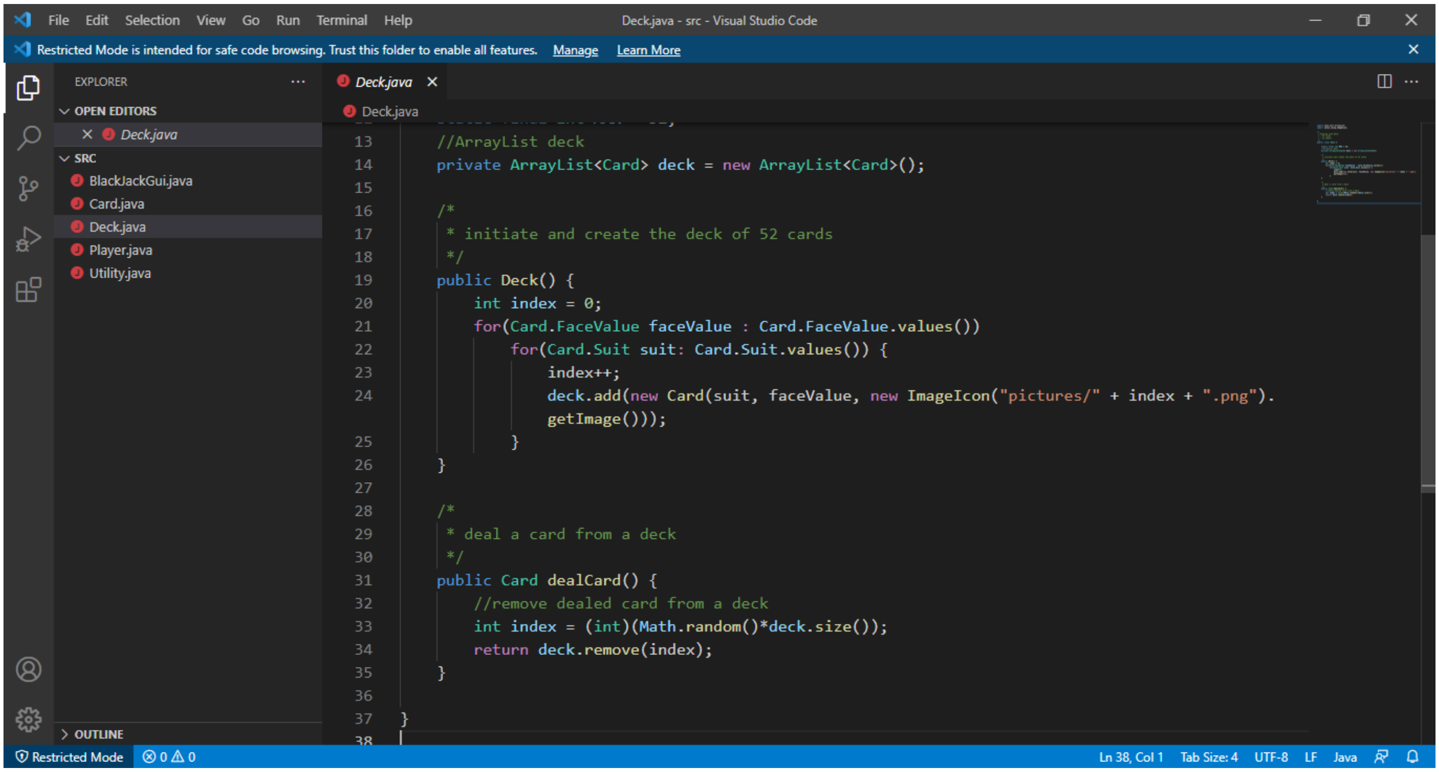
```java
        return faceValue;
    }

    /**
     * set FaceValue
     */
    public void setFaceValue(FaceValue faceValue) {
        this.faceValue = faceValue;
    }

    /**
     * return image
     */
    public Image getImage() {
        return image;
    }

    /**
     * set image
     */
    public void setImage(Image image) {
        this.image = image;
    }

}

```

# Deck.java



Restricted Mode is intended for safe code browsing. Trust this folder to enable all features. Manage Learn More

```java

import java.util.ArrayList;
import javax.swing.ImageIcon;

/*
 * Playing card deck:
 * - 52 cards
 * - no jokers
 */
public class Deck {

    static final int MAX = 52;
    //ArrayList deck
    private ArrayList<Card> deck = new ArrayList<Card>();

    /*
     * initiate and create the deck of 52 cards
     */
    public Deck() {
        int index = 0;
        for(Card.FaceValue faceValue : Card.FaceValue.values())
            for(Card.Suit suit: Card.Suit.values()) {
                index++;
                deck.add(new Card(suit, faceValue, new ImageIcon("pictures/" + index + ".png").getImage()));
            }
    }
```

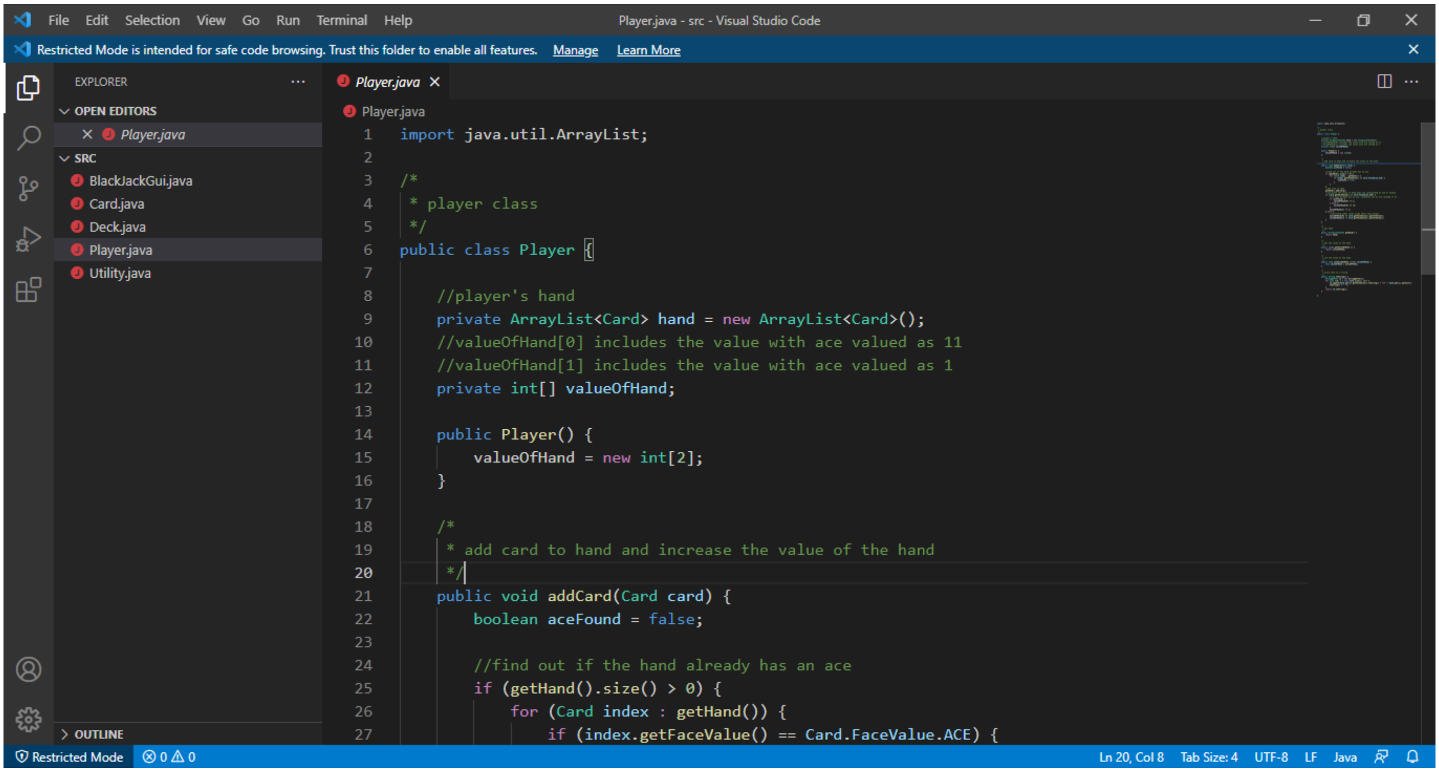


Restricted Mode is intended for safe code browsing. Trust this folder to enable all features. Manage Learn More

```java
    //ArrayList deck
    private ArrayList<Card> deck = new ArrayList<Card>();

    /*
     * initiate and create the deck of 52 cards
     */
    public Deck() {
        int index = 0;
        for(Card.FaceValue faceValue : Card.FaceValue.values())
            for(Card.Suit suit: Card.Suit.values()) {
                index++;
                deck.add(new Card(suit, faceValue, new ImageIcon("pictures/" + index + ".png").getImage()));
            }
    }
```

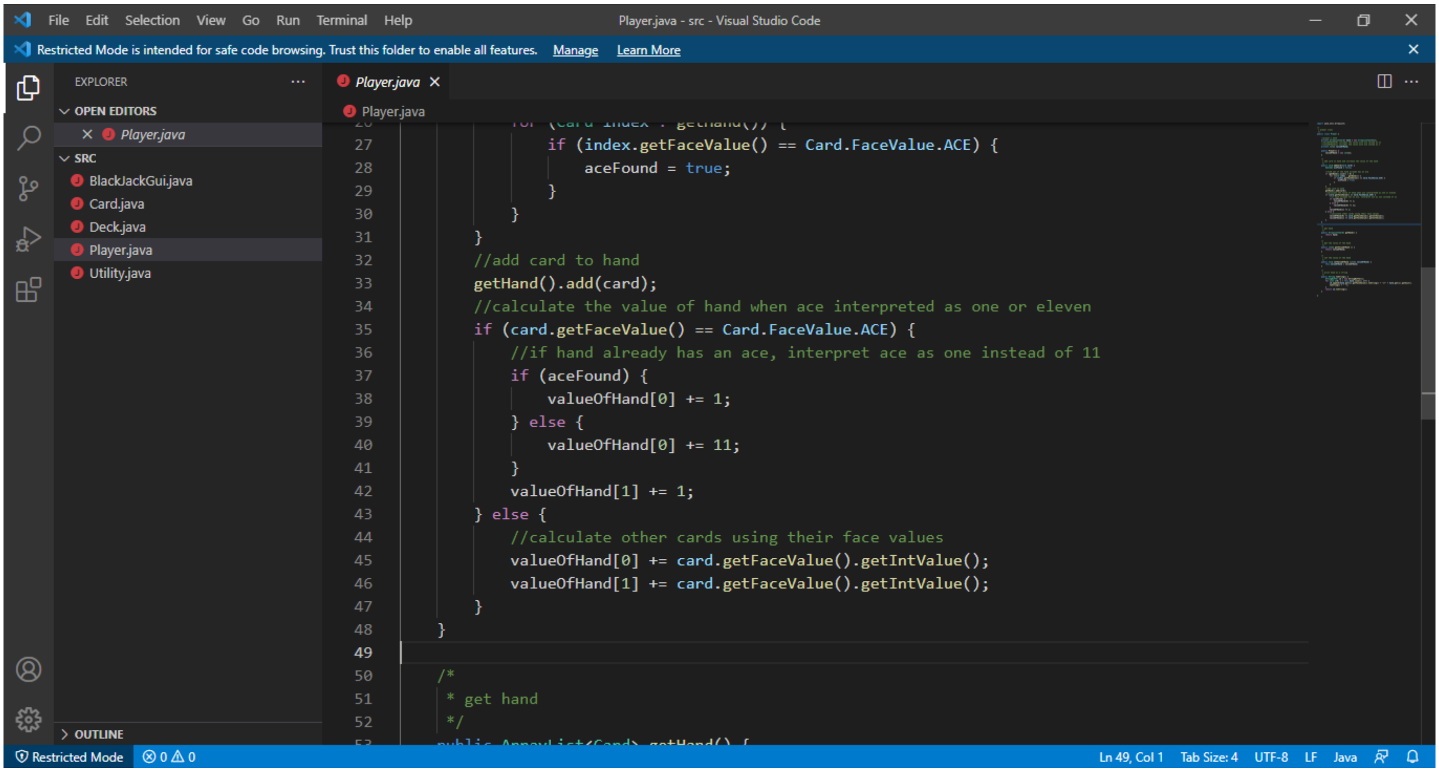
```java

    /*
     * deal a card from a deck
     */
    public Card dealCard() {
        //remove dealed card from a deck
        int index = (int)(Math.random()*deck.size());
        return deck.remove(index);
    }

}

```

# Player.java



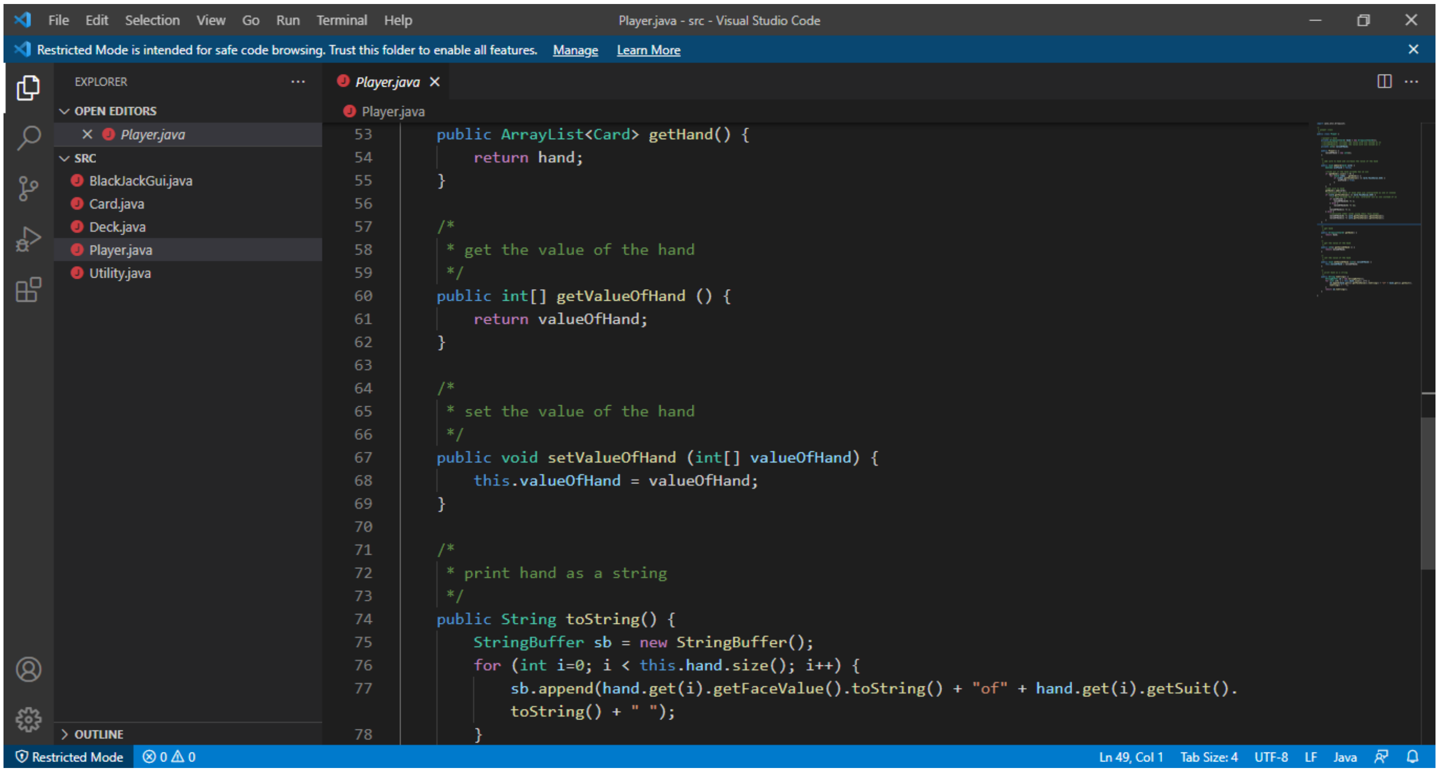
Restricted Mode is intended for safe code browsing. Trust this folder to enable all features.

```java
import java.util.ArrayList;

/*
 * player class
 */
public class Player {

    //player's hand
    private ArrayList<Card> hand = new ArrayList<Card>();
    //valueOfHand[0] includes the value with ace valued as 11
    //valueOfHand[1] includes the value with ace valued as 1
    private int[] valueOfHand;

    public Player() {
        valueOfHand = new int[2];
    }

    /*
     * add card to hand and increase the value of the hand
     */
    public void addCard(Card card) {
        boolean aceFound = false;

        //find out if the hand already has an ace
        if (getHand().size() > 0) {
            for (Card index : getHand()) {
                if (index.getFaceValue() == Card.FaceValue.ACE) {
```

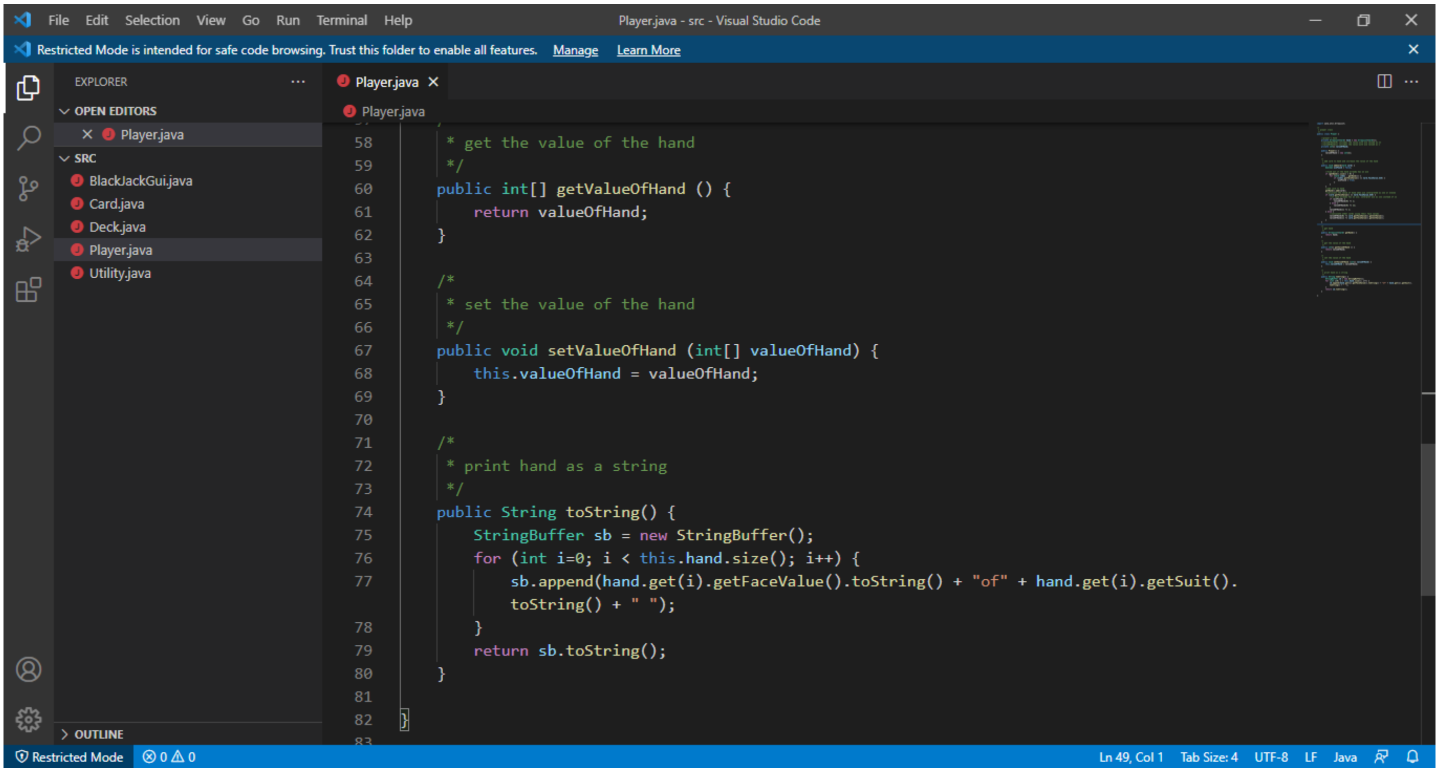


Restricted Mode is intended for safe code browsing. Trust this folder to enable all features.

```java
            for (Card index : getHand()) {
                if (index.getFaceValue() == Card.FaceValue.ACE) {
                    aceFound = true;
                }
            }
        }
        //add card to hand
        getHand().add(card);
        //calculate the value of hand when ace interpreted as one or eleven
        if (card.getFaceValue() == Card.FaceValue.ACE) {
            //if hand already has an ace, interpret ace as one instead of 11
            if (aceFound) {
                valueOfHand[0] += 1;
            } else {
                valueOfHand[0] += 11;
            }
            valueOfHand[1] += 1;
        } else {
            //calculate other cards using their face values
            valueOfHand[0] += card.getFaceValue().getIntValue();
            valueOfHand[1] += card.getFaceValue().getIntValue();
        }
    }

    /*
     * get hand
     */
    public ArrayList<Card> getHand() {
```

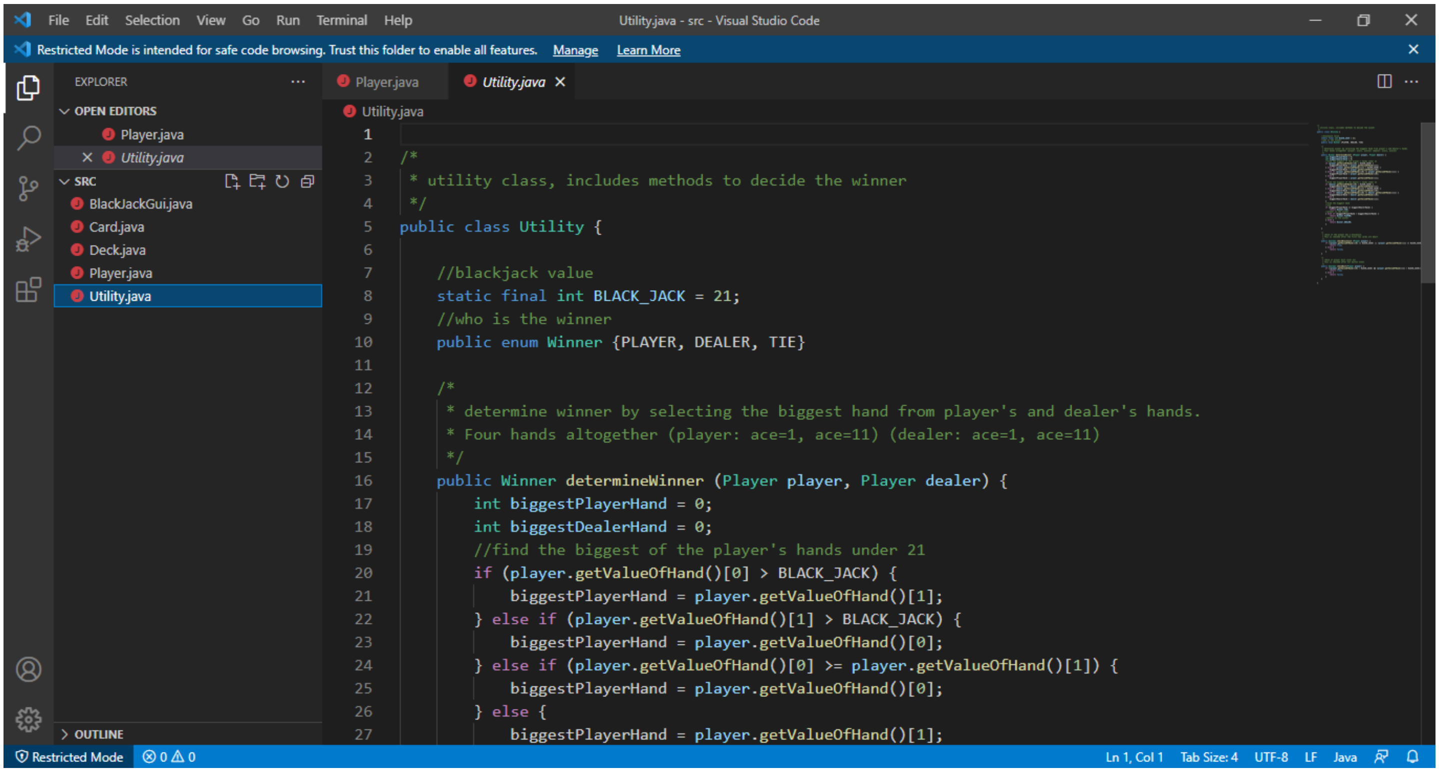


```java
    public ArrayList<Card> getHand() {
        return hand;
    }

    /*
     * get the value of the hand
     */
    public int[] getValueOfHand () {
        return valueOfHand;
    }

    /*
     * set the value of the hand
     */
    public void setValueOfHand (int[] valueOfHand) {
        this.valueOfHand = valueOfHand;
    }

    /*
     * print hand as a string
     */
    public String toString() {
        StringBuffer sb = new StringBuffer();
        for (int i=0; i < this.hand.size(); i++) {
            sb.append(hand.get(i).getFaceValue().toString() + "of" + hand.get(i).getSuit().
            toString() + " ");
        }
```

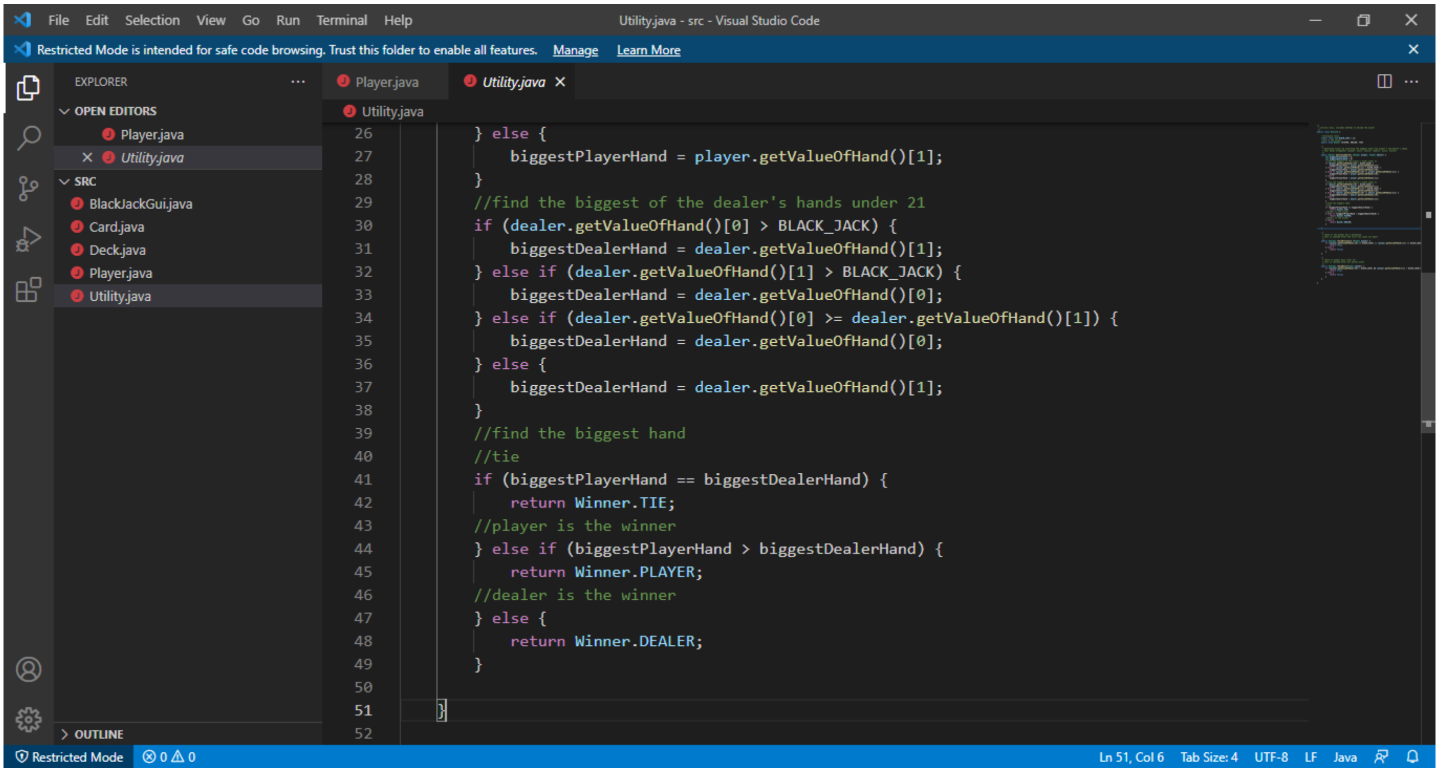


```java
     * get the value of the hand
     */
    public int[] getValueOfHand () {
        return valueOfHand;
    }

    /*
     * set the value of the hand
     */
    public void setValueOfHand (int[] valueOfHand) {
        this.valueOfHand = valueOfHand;
    }

    /*
     * print hand as a string
     */
    public String toString() {
        StringBuffer sb = new StringBuffer();
        for (int i=0; i < this.hand.size(); i++) {
            sb.append(hand.get(i).getFaceValue().toString() + "of" + hand.get(i).getSuit().
            toString() + " ");
        }
        return sb.toString();
    }

}
```

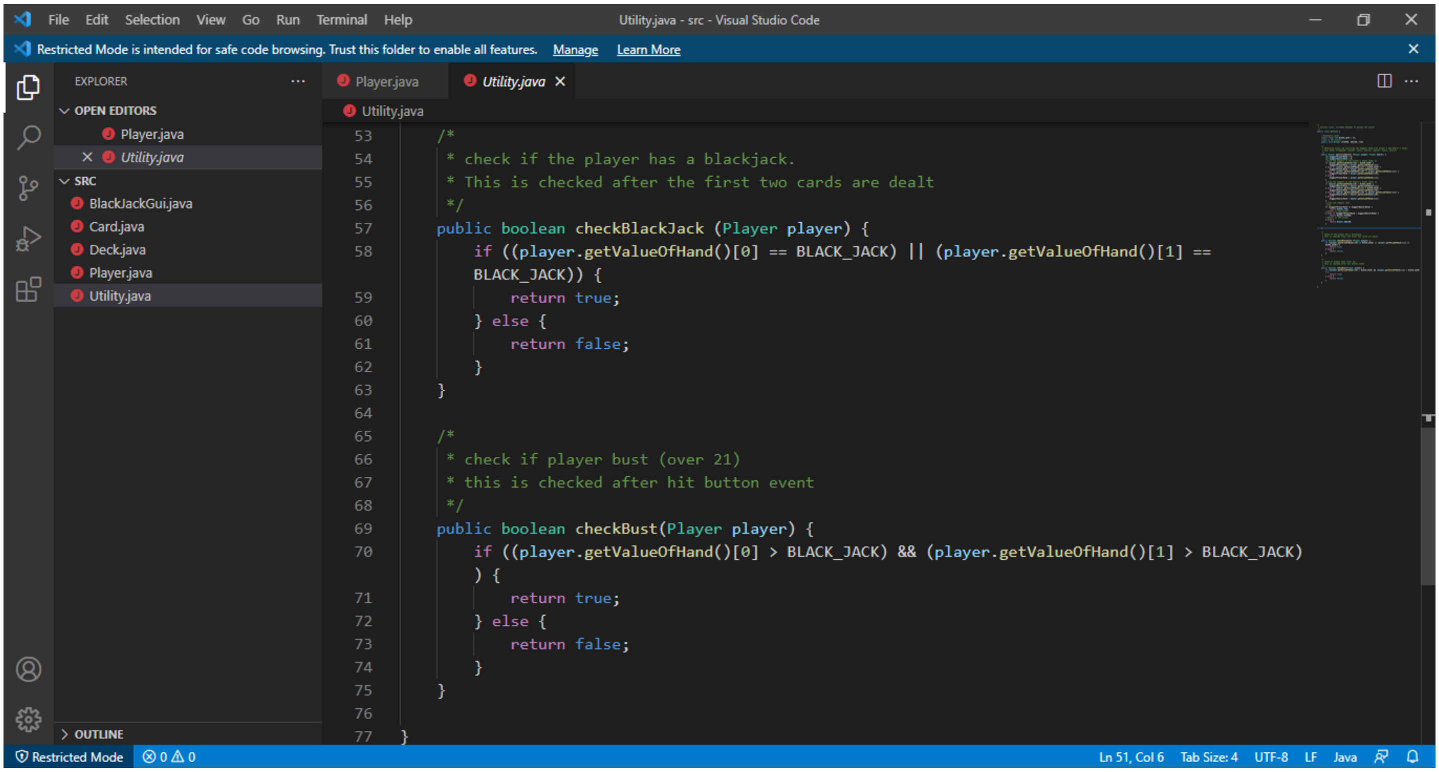
# Utility.java


File Edit Selection View Go Run Terminal Help
Utility.java - src - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this folder to enable all features. Manage Learn More
EXPLORER
OPEN EDITORS
Player.java
Utility.java
SRC
BlackJackGui.java
Card.java
Deck.java
Player.java
Utility.java
OUTLINE
Player.java
Utility.java
Restricted Mode
⊗ 0 ⚠ 0
Ln 1, Col 1 Tab Size: 4 UTF-8 LF Java

```java

/*
 * utility class, includes methods to decide the winner
 */
public class Utility {

    //blackjack value
    static final int BLACK_JACK = 21;
    //who is the winner
    public enum Winner {PLAYER, DEALER, TIE}

    /*
     * determine winner by selecting the biggest hand from player's and dealer's hands.
     * Four hands altogether (player: ace=1, ace=11) (dealer: ace=1, ace=11)
     */
    public Winner determineWinner (Player player, Player dealer) {
        int biggestPlayerHand = 0;
        int biggestDealerHand = 0;
        //find the biggest of the player's hands under 21
        if (player.getValueOfHand()[0] > BLACK_JACK) {
            biggestPlayerHand = player.getValueOfHand()[1];
        } else if (player.getValueOfHand()[1] > BLACK_JACK) {
            biggestPlayerHand = player.getValueOfHand()[0];
        } else if (player.getValueOfHand()[0] >= player.getValueOfHand()[1]) {
            biggestPlayerHand = player.getValueOfHand()[0];
        } else {
            biggestPlayerHand = player.getValueOfHand()[1];
```


File Edit Selection View Go Run Terminal Help
Utility.java - src - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this folder to enable all features. Manage Learn More
EXPLORER
OPEN EDITORS
Player.java
Utility.java
SRC
BlackJackGui.java
Card.java
Deck.java
Player.java
Utility.java
OUTLINE
Player.java
Utility.java
Restricted Mode
⊗ 0 ⚠ 0
Ln 51, Col 6 Tab Size: 4 UTF-8 LF Java

```java
        } else {
            biggestPlayerHand = player.getValueOfHand()[1];
        }
        //find the biggest of the dealer's hands under 21
        if (dealer.getValueOfHand()[0] > BLACK_JACK) {
            biggestDealerHand = dealer.getValueOfHand()[1];
        } else if (dealer.getValueOfHand()[1] > BLACK_JACK) {
            biggestDealerHand = dealer.getValueOfHand()[0];
        } else if (dealer.getValueOfHand()[0] >= dealer.getValueOfHand()[1]) {
            biggestDealerHand = dealer.getValueOfHand()[0];
        } else {
            biggestDealerHand = dealer.getValueOfHand()[1];
        }
        //find the biggest hand
        //tie
        if (biggestPlayerHand == biggestDealerHand) {
            return Winner.TIE;
        //player is the winner
        } else if (biggestPlayerHand > biggestDealerHand) {
            return Winner.PLAYER;
        //dealer is the winner
        } else {
            return Winner.DEALER;
        }

    }

```



Restricted Mode is intended for safe code browsing. Trust this folder to enable all features.

```java
    /*
     * check if the player has a blackjack.
     * This is checked after the first two cards are dealt
     */
    public boolean checkBlackJack (Player player) {
        if ((player.getValueOfHand()[0] == BLACK_JACK) || (player.getValueOfHand()[1] == BLACK_JACK)) {
            return true;
        } else {
            return false;
        }
    }

    /*
     * check if player bust (over 21)
     * this is checked after hit button event
     */
    public boolean checkBust(Player player) {
        if ((player.getValueOfHand()[0] > BLACK_JACK) && (player.getValueOfHand()[1] > BLACK_JACK)) {
            return true;
        } else {
            return false;
        }
    }
}
```

# TESTING AND DEPLOYEMENT

Software Testing is a Process of executing a program with the intent of finding errors during the run-time of program. It a feasible task to try and find the errors (whose presence is assumed) in a program, as it is a destructive process.

I have tried to understand the proposed system by detailed study of the various operations that will be performed by a system.

System analysis is the process of studying an existing system to determine how it works and how it meets user needs. System analysis lays the groundwork for improvements to the system. The analysis involves an investigation, which is turn usually involves establishing a relationship with the client (Player), for whom the analysis is done, and with the user of the system. This analysis phase is more of a thinking process. In this phase, I have improved logical aspects of the system.

To develop the system, one must deal with errors, bugs, defects etc. in more seamless way than ever, in order to preserve the integrity of Project and also to maintain the flow of maintenance.

I did thorough examination of the system processes, gathering Operational data, understanding the information flow, finding out weaknesses and evolving solutions for overcoming the weaknesses of the system so as to achieve the goals.

During the analysis phase, I dealt with:

- Data Gathering
- Data Analysis

Gathering the data for the completion of the Project was hard and also expensive, given the complexity of the Project. Once the gathering was done, Analysis phase was started, leading to thorough examination of the Project to make less prone to bugs, errors, defects etc.

# CHALLENGES AND FUTURE SCOPES

*"There is always room for improvements"*

There are lot of things that can be added to the Project in future to make it more dynamic with respect to time.

Following are the abilities that can be added to the Project to make more modern and fun and also visually – appealing.

- Making the game executable (.exe) rather than java archive file (.jar) to reduce the necessity of JDK pre – installed in Player system.
- Making the GUI modern by utilizing the concepts of UI/UX (Colour Theory, Choosing right font style).
- Making game A.I. more competitive.
- Adding the ability to play sound with each user – interaction.
- Making animations smoother.

The challenge here will be adding the features in the Project without making the Project complex which can result in poor maintainability.

Challenges can be overcome by refactoring the Project from time to time to increase Code Maintainability.

# CONCLUSION

The main objective of the project was to develop an offline casino-based game which utilizes lesser resources but does not compromise with user-experience (UX).

I had taken a wide range of literature review in order to achieve all the tasks, where I came to know about some of the products that are existing in the market. I made detailed research in that path to cover the loop holes that existing systems are facing and to eradicate them in this Project. In the process of research, I came to know about the latest technologies and different algorithms, some of which I used in this Project.

# BIBLIOGRAPHY

- https://app.diagrams.net/
- https://www.ukessays.com/essays/computer-science/blackjack-playergame-development.php
- https://www.123helpme.com/essay/Blackjack-Essay-483801
- https://www.wikipedia.com/Blackjack
- https://www.stackoverflow.com/