Arizza Santos
Dr. Raunak
CS482 Software Engineering
ASN1: Design Description

Model-View-Controller Architecture:

Model : SimpleBlackjack.java, Card.java, Deck.java, Player.java, Hand.java, Rank.java, Suit.java
View: activity_main.xml, activity_main_landscape.xml
Controller: MainActivity.java

Android Emmulator Used: Nexus 4

XML Files Used & Descriptions : These are found in *res*.

AndroidManifest.xml : This specifies the resources that the app uses, such as activities. Found in *manifests*.

activity_main.xml : The GUI for the portrait orientation is defined in this. The RelativeLayout is the main view which holds the other View elements, such as the TextViews and ImageViews. Found in *res* > *layout*.

activity_main_landscape.xml : The GUI for the landscape orientation is defined in this. This is similar to activity_main.xml but the View elements are formatted for the landscape orientation. Found in *res* > *layout*.

colors.xml : This defines the colors used in the app. Found in *res* > *values*.

strings.xml : This defines the strings used in the app. Found in *res* > *values*.

styles.xml : This defines the various styles used in the app. These are used to eliminate some repetitiveness from within the attributes of the View elements. Found in *res* > *values*.

res > *drawable* contains the images used in the app

Java Files Used & Descriptions : These are found in *java* > *com.example.arizz.simpleblackjack*.

MainActivity.java : This represents the Controller of the app, which acts as the "middle-man" between the View and Model.

SimpleBlackjack.java : This represents a simple Blackjack game, which acts the Model for the app. This model utilizes other java classes.

Suit.java : This is an enumeration with the four suits a card can have.

Rank.java : This is an enumeration with the thirteen ranks a card can have.

Card.java : This represents a Card. A Card has a rank, a suit, a value based on its rank, and an image id used to access the specific card image in *drawable.*

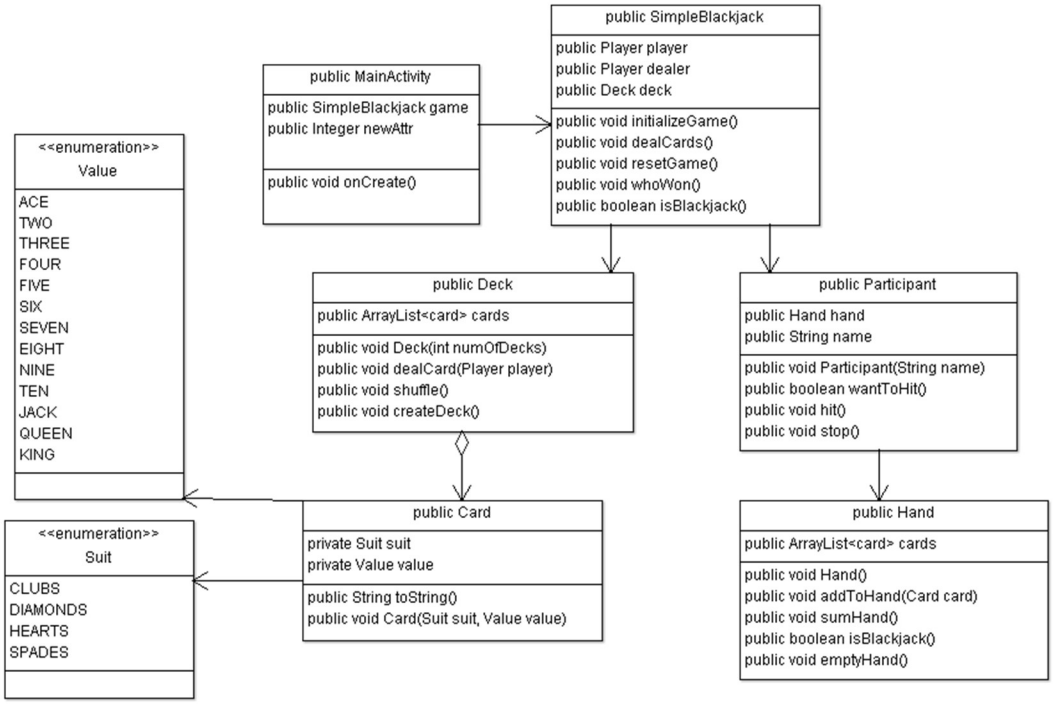
Deck.java : This represents a Deck of cards. It uses an ArrayList to store Card objects.

Player.java : This represents a Player in the game. The Player has a Hand.

Hand.java : This represents a Player's Hand. It uses an ArrayList to store current Cards in the hand.

My *StateImage* folder contains screenshots of the app at various states in the game. I used it to test that the requirements are working for the different results.

YouTube Link to Demo: https://youtu.be/shvzgC7dkgk or https://www.youtube.com/watch?v=shvzgC7dkgk


public SimpleBlackjack
public Player player
public Player dealer
public Deck deck
public void initializeGame()
public void dealCards()
public void resetGame()
public void whoWon()
public boolean isBlackjack()
public MainActivity
public SimpleBlackjack game
public Integer newAttr
public void onCreate()
<<enumeration>>
Value
ACE
TWO
THREE
FOUR
FIVE
SIX
SEVEN
EIGHT
NINE
TEN
JACK
QUEEN
KING
public Deck
public ArrayList<card> cards
public void Deck(int numOfDecks)
public void dealCard(Player player)
public void shuffle()
public void createDeck()
public Participant
public Hand hand
public String name
public void Participant(String name)
public boolean wantToHit()
public void hit()
public void stop()
<<enumeration>>
Suit
CLUBS
DIAMONDS
HEARTS
SPADES
public Card
private Suit suit
private Value value
public String toString()
public void Card(Suit suit, Value value)
public Hand
public ArrayList<card> cards
public void Hand()
public void addToHand(Card card)
public void sumHand()
public boolean isBlackjack()
public void emptyHand()