Version 1.0

winter 2019

syst 17796

Deliverable 3

# Overview

## Blackjack - How to Play

Our Goal for this project is to create the game of blackjack as accurate and as fun as possible. Our goal of our game is to simulate what it would be like to play blackjack at a casino. Our final version of the game will have the ability to bet and win just like the real game of blackjack. Here is a link to the rules and as well as how to play blackjack.

The game will be written in java with no plans on making a GUI and make it a text-based game of blackjack. The current state of our code is pretty much at its bare bones right with only being able to generate and shuffle the of cards and deal the players hand. We also added a rule to make thing interesting. The rule is the first dealt Ace a player receives will be counted as an 11 and every other Ace will be used as intended. We add this to minimize the chance of getting two aces for an instant black jack.

## Project Scope

Describe the names and roles of each team member. Describe the technical scope of the project by talking about the interface and how you will know when the project is complete.

1. Luigi Agostino will be focusing on the documentation of the project as well as coding.
2. Spencer Gagnier will be the lead programmer of this project and will be the final say to changes made by him or other people in the group.

Our project will be complete when we can make the user bet on his hand see whether he wins or losses and payout or take away from the pool of money the player has.

## High-Level Requirements

1. Ability to shuffle and deal a hand to the house and the player.
2. Have a pot that can be won by the player
3. Have the player be able to bet on the hand he has
4. Can play again if the player choses too

## Implementation Plan

**GITHUD URL:** [**https://github.com/Squigie16/BlackJack**](https://github.com/Squigie16/BlackJack)

Each member must at least make 2 changes a week to the code. And every class discuss the issues or development of the project. All UML, and text files will be with Luigi Agostino on his USB in a project folder. We will be using the IDK NetBeans.

## Project Background and Description

When the game starts up you will be asked for your name. Once the user enters his/hers name they will be given $10,000 and must make the opening bet. Once the bet is made the Cards will be dealt; one to the player one to the dealer face down and then one to the player and another to the to the dealer this time face up. Once all the cards are dealt the player will be given the option to hit or stand. However, once the player goes over 21 before the face down card is turned over, they lose. Once they stay/stand, they dealer will reveal his hand as well as the players hand and the closets to 21 will get the win. If there is a tie it is a split pot. The player will win if he gets to $20,000. The player will lose if he loses all his money.

## Design Considerations

## 

**(Figure 1.1)**

In figure 1.1 you can see we have multiple classes as well as a couple of objects. We have the card object as well as the CardHand class. The Card object which holds the number of cards (Deck). The CardHand holds the players cards when the cards are dealt. If you look at the object, we broke up the cards into suit and Value and later randomize them in the card object. We also made sure that there are only 4 suits as well as make sure that each suit only has a total of 13 cards (Ace-King).

The class we have added is the Winner/Losing class to determine the winner of game as well as determining the loser of the game. It calls the player object as well as the player hand. It checks the and against the Dealers hand which is in the CardHand Class, against the players hand which is also in the CardHand class. It checks which player has the highest hand as well as checking if either the player or Dealer has Blackjack. The Player object also holds the players Amount of money and also checks if they meet the win condition in the Winner Class.

## Test Results Table

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Use Case | Test Method (ClassName.methodName) | Status (Date) |
| Player is dealt 2 cards. | “Regular Play” |  | Pass (April 10, 2019) |
| Dealer is dealt one card face up and one face down. | “Regular play” |  | Pass (April 4, 2019) |
| Player wins when he has black jack | “Regular Play” |  | Pass (April 12, 2019) |
| Dealer and player Tie | “Regular Play” |  | Pass (April 16, 2019) |
| Dealer Loses by going over 21 | “Regular Play” |  | Pass (April 16, 2019) |
| Player loses by going over 21 | “Regular Play” |  | Pass (April 16, 2019) |
|  |  |  |  |

## 