## Project Name: Black Jack

## Overview

Our project is a simple one player card game. The user of the game plays against the dealer (the computer) in a game of Black Jack. The objective of the game is to collect a hand of cards whose value add up to as close to 21 as possible without going over 21. At the start of the game, the player and the dealer are dealt two cards. The player then can choose to get another card (hit) or to keep his/her hand of cards as is (stand). The player can continue to hit as long as he/she wants. If the player’s hand adds up to more than 21, the player goes bust and the dealer automatically wins. If the player stands (without going bust), the dealer updates his hand using the following rules. If the dealer’s hand adds up to 16 or less, the dealer must get another card (hit). If the dealer’s hand adds up to 17 or more, the dealer must keep the hand (stand). After the dealer stands (or goes bust), the two hands are compared. Whichever hand is closest to 21 without going over is the winner.

The value of a hand is calculated by adding up the values of all the cards. Number cards have the value of their number. An ace has a value of 11 or 1 (whichever is better in the given situation). All other face cards have a value of 10. A two card hand that consists of an ace and a 10 valued card, called Black Jack, wins over any other hand. If the dealer’s and the player’s hands are identical in value, the game is considered a tie.

## Functional Requirements:

FR1:At the start of the game, the player and the dealer are dealt two cards each from the top of a shuffled deck.

FR2: After getting the initial two cards, the player can choose to hit or stand.

FR3: If the player chooses to hit, another card is added to the player’s hand.

FR4: The player can continue to hit as long as the player’s hand value is less than 21.

FR5: The player is forced to stand on 21.

FR6: If the player’s hand is over 21, the player loses.

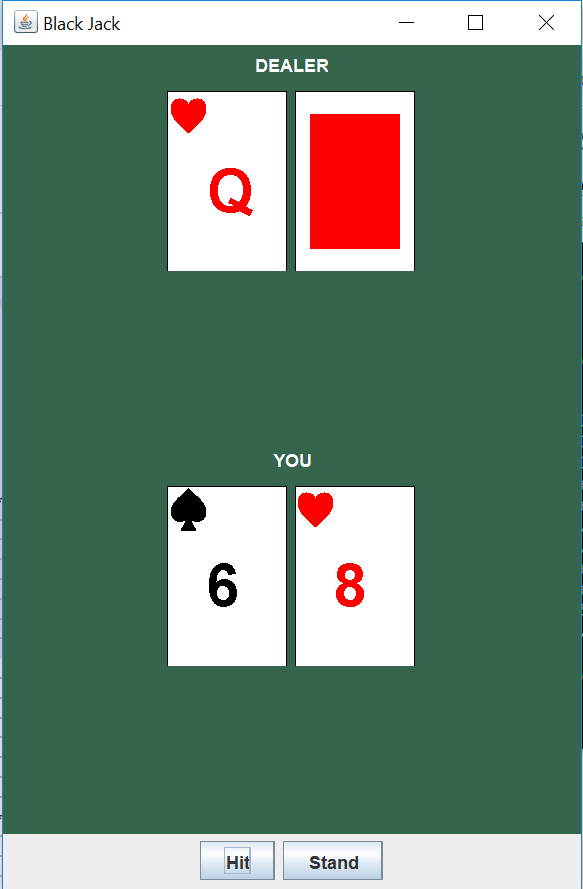
FR7: When the player chooses to stand, the dealer’s hand is updated as follows:

1. If the dealer’s hand adds up to 16 or less, another card is added to the dealer’s hand.
2. If the dealer’s hand adds up to 17 or more, the dealer cannot get any more cards.
3. If the dealer’s hand adds up to more than 21, the dealer loses.

FR8: When the dealer stands, both hands are compared in value and the results are announced.

1. A Black Jack hand (an ace with a 10 valued card) wins over any other hand.
2. If both hands are equivalent in value, the game is a tie.
3. If the dealer's hand value is greater than the player's hand value, the dealer wins.
4. If the player’s hand value is greater than the dealer’s hand value, the player wins.

## User Interface Prototype:



The prototype above shows the initial state of the game.

## User Interface Requirements:

UIR1: When the user clicks the “Hit” button, FR3 is executed.

UIR2: When the user clicks the “Stand” button, FR7 is executed.

UIR2: All cards in the player’s hand are shown face up.

UIR3: At the start of the game, one card in the dealer’s hand is shown face up, the other one is shown face down.

UIR4: When the user clicks the “Stand” button, the “Hit” and “Stand” buttons get disabled.

UIR5: At the end of the game, all of the dealer's cards are shown face up.

## Project Plan

| Requirement | Deliverable 1 | Deliverable2 | Deliverable 3 |
| --- | --- | --- | --- |
| FR1 | x |  |  |
| FR2 | x |  |  |
| FR3 | x |  |  |
| FR4 |  | x | x |
| FR5 |  | x | x |
| FR6 | x |  |  |
| FR7 | x |  |  |
| FR8 |  |  | x |
| UIR1 |  | x |  |
| UIR2 |  | x |  |
| UIR3 |  | x |  |
| UIR4 |  | x |  |
| UIR5 |  |  | x |