Russell Lilljedahl

Blackjack, Java

Plan

25 July 2018

Contents

[Table of Revisions 3](#_Toc520831772)

[1. Requirements 4](#_Toc520831773)

[2. Blackjack Rules 5](#_Toc520831774)

[3. Project Design 6](#_Toc520831775)

[4. Pseudocode 8](#_Toc520831776)

[5. Future Enhancements: 10](#_Toc520831777)

# Table of Revisions

Revision Name/Date Description

|  |  |  |
| --- | --- | --- |
| 0.1 | Russell Lilljedahl / 25 July 2018 | Initial |
| 0.2 | Russell Lilljedahl / 30 July 2018 | First testing |
| 0.3 | Russell Lilljedahl / 31 July 2018 | Clean up |
| 0.4 |  |  |
| 0.5 |  |  |
| 0.6 |  |  |

1. Requirements

Project is to recreate a black jack card game. It must implement a feature to keep the statistics of all games played.

# 2. Blackjack Rules

Blackjack Rules

https://www.blackjackinfo.com/blackjack-rules/

1. Blackjack may be played with one to eight decks of 52-card decks. This game will play with one deck that is shuffled after every hand.

2. Aces may be counted as 1 or 11 points, 2 to 9 according to their face value, and tens and face cards count as ten points.

3. The value of a hand is the sum of the point values of the individual cards. Except, a "blackjack" is the highest hand, consisting of an ace and any 10-point card, and it outranks all other 21-point hands.

4. After the players have bet, the dealer will give two cards to each player and two cards to himself. One of the dealer cards is dealt face up. The facedown card is called the "hole card."

5. Play begins with the player to the dealer's left. The following are the choices available to the player:

Stand: Player stands pat with his cards. This means that the player is done and it is the dealers turn.

Hit: Player draws another card (and more if he wishes). If this card causes the player's total points to exceed 21 (known as "breaking" or "busting") then he loses.

Forfeit: The player is done playing.

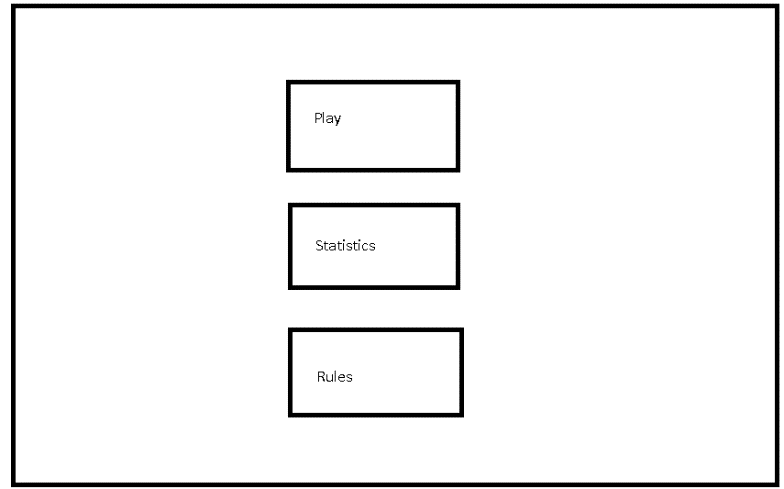
6. After each player has had his turn, the dealer will turn over his hole card. If the dealer has less than 16, then they will draw another card.

7. If the dealer goes over 21 points, then any player who didn't already bust will win.

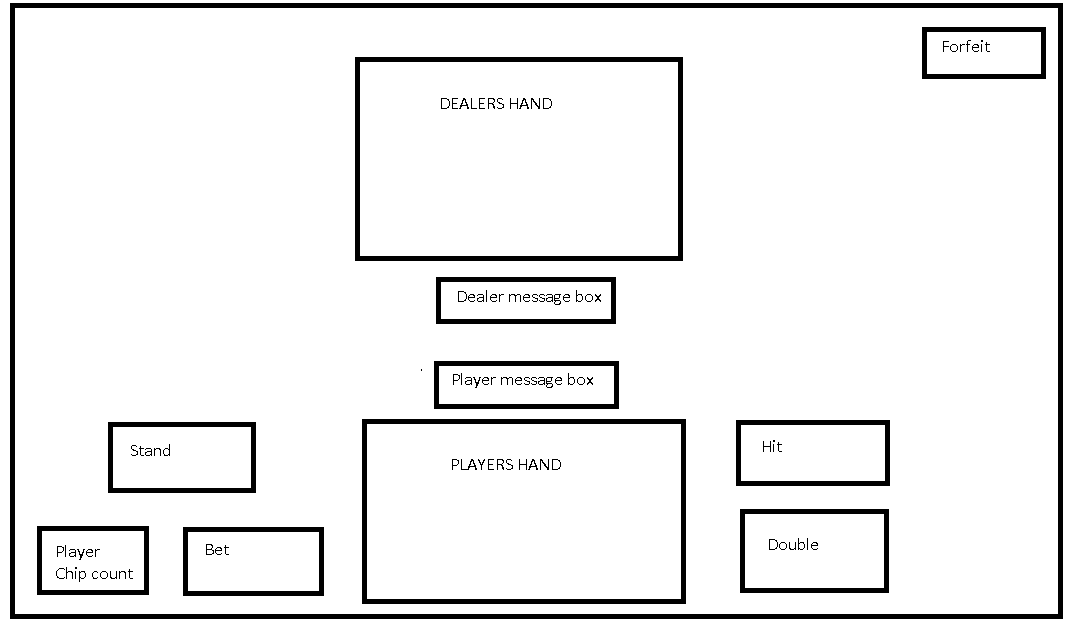
8. If the dealer does not bust, then the higher point total between the player and dealer will win.

# 3. Project Design

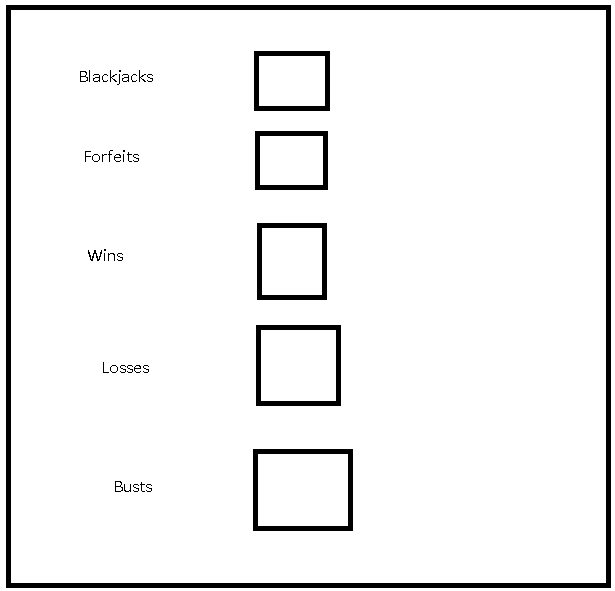
1. Main class:
   1. GUI



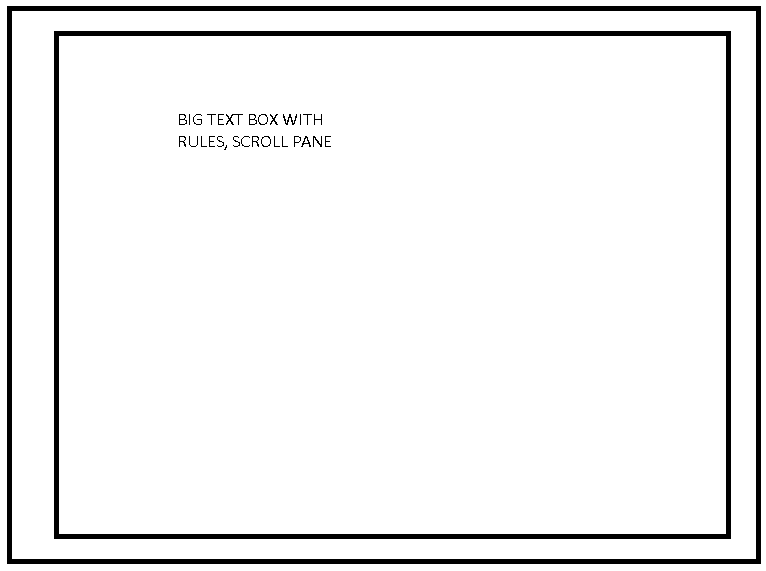
1. Play game class:
   1. GUI



1. Statistics class
   1. GUI



1. How to play Class
   1. GUI



# 4. Pseudocode

1. MainClass
   1. If statistics.txt exists
      1. Continue
   2. Else
      1. Create statistics.txt
      2. Set values
   3. MainMenuGUI
2. MainMenuGUI
   1. initGUI
   2. if playButton.isClicked
      1. go to PlayGameGUI
   3. if statisticsButton.isClicked
      1. go to StatisticsGUI
   4. if rulesButton.isClicked
      1. go to rulesGUI
3. PlayGameGUI
   1. initGUI
   2. Constructor
      1. Deck.CreateDeck
      2. Deck.ShuffleDeck
      3. Get money count, global.getChipCount
   3. Betting
      1. Place bet, max 50
      2. Click deal
   4. DealCards
      1. Player and dealer receives 2 cards.
         1. Players cards are shown, one of dealers is shown
      2. Click hit, forfeit, or stand
         1. If hitButton.isSelected, hit()
         2. If forfeitButton.isSelected, forfeit()
         3. If standButton.isSelected, stand()
   5. Hit()
      1. Player receives another card
      2. Re-tally scores
   6. Forfeit()
      1. Save data to global
      2. Confirm forfeit
      3. Exit to main menu
   7. Stand()
      1. Pass turn to dealer
4. Deck class
   1. Create deck
      1. List<Cards> cards = new ArrayList();
   2. Shuffle deck
      1. Collections.shuffle
   3. GetCard
   4. GetValue
5. NPC Interface
   1. Player class
      1. Hit()
      2. Stand()
      3. Bet()
   2. Dealer class
      1. Hit() // if card count is 15 or below
      2. Stand()
6. Save class
   1. Save data
7. Load class
   1. Load data
8. Statistics class (wins, losses, blackjacks, forfeits, busts, chipCount)
   1. Int chipCount
   2. Int forfeits
   3. Int blackjacks
   4. Int losses
   5. Int wins
   6. Int busts

# 5. Future Enhancements:

Add Insurance, split and double. Have to read up on these.

Card images.