CIS-7 Project Documentation

Team GoodLane Blackjack Program

Team Members

- Ryan Lane
- Danny Goodlow

Project Information

Problems being solved:
This program attempts to solve the problem of calculating the chance of winning a game of Blackjack for each hand.

Solutions:
This program takes an object-oriented approach to the problem, using Hand and Deck classes to help simulate the game.

Algorithm and Calculations:
The program utilizes vectors to represent a set of cards in a hand, as well as a relational structure to represent the amount of each card number in a deck. The program calculates the probability of drawing particular cards from a deck by summing the amounts of the query cards in the deck and dividing by the total number of cards. A random card is drawn from a deck by picking a random position in the deck up to the total number of cards, then using the amounts of each card to select the random card from an ordered deck. This should be functionally the same as shuffling.

Objectives and User Interaction:
The program's objective was to implement a blackjack simulation showing the probability and chance of winning a game of Blackjack. The program displays the dealer and player hands to the user, as well as their sums and the probability of winning on a stand. The user may enter "H" or "S" to decide whether to hit or stand, then the program advances to the next hand or the dealer's turn.

Use of Discrete Structures:
Discrete structures were implemented in the C++ program by using the algorithm of probability. The concept of relations is also used to represent a deck of cards with each card number related to an amount of that card in the deck.

Limitations:

This program lacks an extensive method of determining the winning probability, instead using a simple approximation that may be inaccurate in some cases.

Recommendation:
Refactor the winning-chance calculation function to take the dealer’s hole card into account and find probabilities for more than one hit by the dealer.

Pseudocode and Flowchart

Pseudocode

```
1. Start Program

2. Deal hand to user and dealer

Set up a deck, with following requirements:

        ·    - 52 cards

        ·    - one through ten, of each suit, value what's on card

        ·    - ace, of each suit, value = 1 or 11

        ·    - jacks/queens/kings, of each suite, value = 10

3. User can hit or stay, if hit user gets another card

4. If user hits, they get another card

5. If user stays add the users cards up

6. Dealer must hit if his total is less than 17
```

7. Compare user’s hand and dealer’s hand

Write a method for players turn:

- - ask them hit/stay?
- - deal them card 2 if hit
- - if stay, move on to dealers turn
- - if broke or blackjack deliver message

Write a method for dealers turn:

- - if hand value < 17, hit
- - check if hand > player hand
- - if so, dealer wins, end of game
- - if broke or blackjack deliver message

8. If user has closer amount to 21 or 21 exactly user wins

9. If dealer has closer amount to 21 or 21 exactly dealer wins

10. Bust! User loses

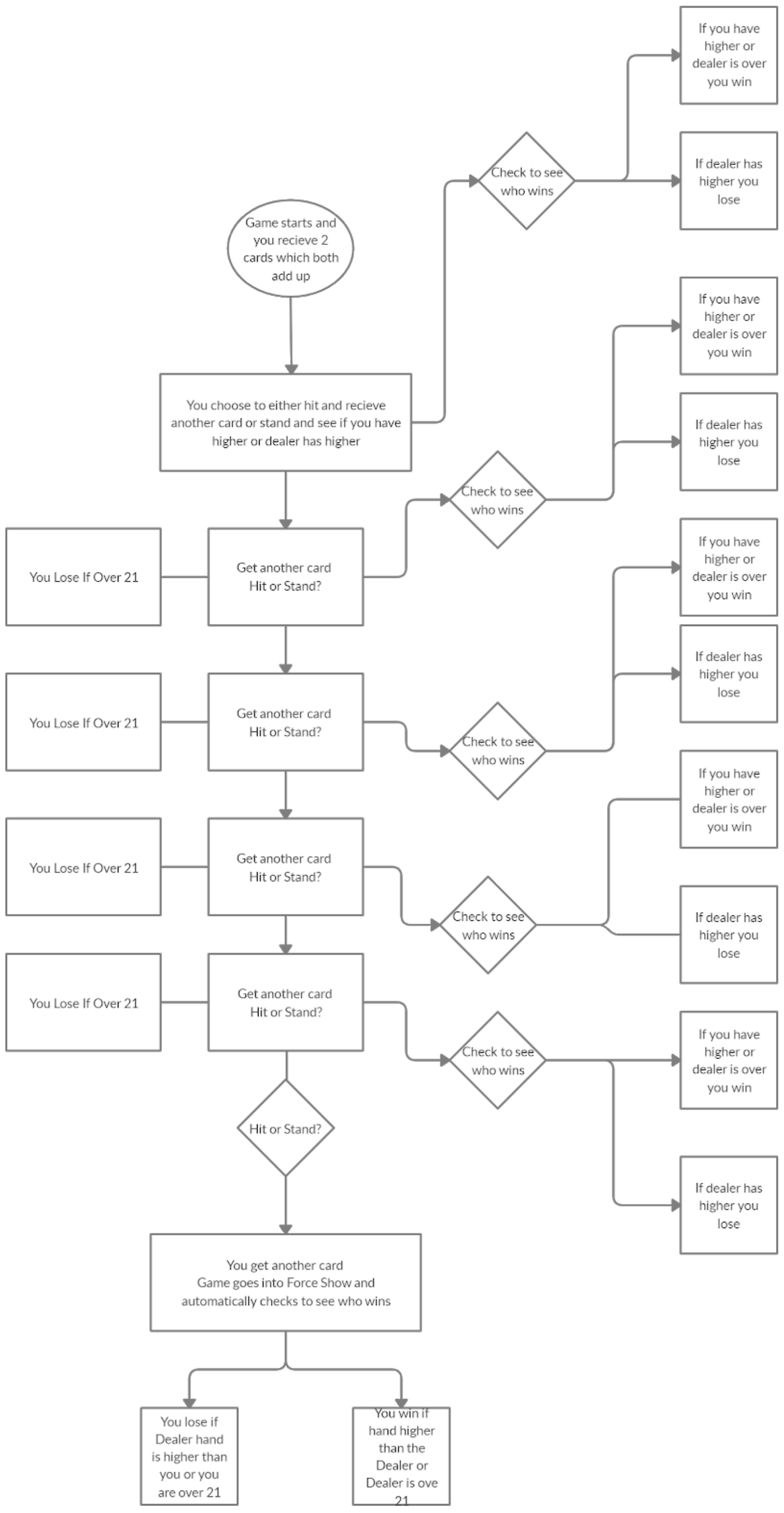
11. Bust! User wins

12. User wins

13. User loses

14. End

Flowchart



Game starts and you recieve 2 cards which both add up
You choose to either hit and recieve another card or stand and see if you have higher or dealer has higher
Check to see who wins
If you have higher or dealer is over you win
If dealer has higher you lose
You Lose If Over 21
Get another card Hit or Stand?
Hit or Stand?
You get another card Game goes into Force Show and automatically checks to see who wins
You lose if Dealer hand is higher than you or you are over 21
You win if hand higher than the Dealer or Dealer is ove 21