Chris Flanagan

Project Write-Up

5/8/19

For my project, I chose to create a game of Blackjack in Java. A user is able to play against the system which acts as a dealer. The game will continue until there is a winner.

I chose to design a game of blackjack in java because I thought it would be challenging, but not exceed my current java coding capabilities. This program includes methods, constructors, arrays, for/while loops, as well as multiple data types for variables and methods. This program encompasses four java classes, as well as an enum.

The objective of the game is to beat the dealer in one of the following ways: Get 21 points on the player's first two cards without a dealer blackjack, reach a final score higher than the dealer without exceeding 21, or let the dealer draw additional cards until their hand exceeds 21 (known as a “bust”). A player will also bust if their hand exceeds 21. Players are each dealt two cards. The dealer is also dealt two cards, normally one up and one down (hidden). The value of cards two through ten is their face value (2 through 10). Face cards (Jack, Queen, and King) are all worth ten. A hand's value is the sum of the card values. Aces can be worth one or eleven Players are allowed to ask for a “hit” and draw an additional card to improve their hand, or if they are satisfied with their hand they can “stay” and not receive a card. The dealer will reveal his card and either hit or stay as well unless his total is greater than 16 and he must stay. This will continue until wither a winner is declared.

A user can interact as a player and play against the program which acts as a dealer. The player will input the amount of money they would like to wager on the game. The system deals two cards to the player and the dealer. One of the dealer’s cards is face down, but both of the

players cards are visible. The system will ask you to hit or stay. The dealer will also hit or stay. This will continue until either the dealer or the player wins by exceeding the value of the other's cards and winning the game, or until either the dealer or the player exceed 21 and lose. The system will then tell you your total winnings or losses, ask if you would like to play again. The system will keep track of your total winnings and losses over multiple game, and display them after each game, until the user leaves the table.

This java program consists of 4 classes and an enum. An enum is similar to a class, but it represents a group of constants. In the case of this project those constants are Hearts, Spades, Clubs, and Diamonds. The four classes are named Player, Card, Deck, and Game. The enum is named Suit.

The Card class uses a toString method to return the string value of each card. The Deck class has methods to shuffle the deck, tell whether the deck is shuffled or not, deal the next card to the player, and print the deck. The player class has several methods intended for allowing th player to play the game, such as a getWager method, getHandSum method, and a printHand method. The Game class runs the actual game of Blackjack, constructs the flow of game and guides the player through the game.

The problem my project solves is the ability for people to play the game of Blackjack from a computer. This project offers a simple way for users to play and wager hypothetical money on a Blackjack game.

Blackjack is a very popular game at casinos all over the world. Many casinos are switching to computerized versions of their games which is where I got the inspiration to create a Blackjack game in java. There have been many other versions of the game Blackjack written in many languages. My system allows users to play with ease and simplicity.

The system is very easy to use. All users must do is enter an “h” for when they wish to hit, and an “s” for when they want to stay. Users will also have to enter a dollar mount for the amount they would like to wager on the hand they are currently playing. The system will display instructions for the user as the game continues. When the hand is complete the system will ask if you would like to play again. The user will enter “yes” to play again and “no” to complete the game.

In conclusion, I designed a version of the popular casino game Blackjack in Java. The system allows users to easily interact with the system as a player while playing against the system who acts as a dealer. The player will enter his wager and begin the game. Classic Blackjack rules will commence and the player will choose to hit or stay after seeing one of the dealer’s cards and both of his. If the player losses they will be notified and told their total winnings. If the player loses they will be notified and told their total losses. The game will continue until the player leaves the decides to not play anymore.

Card
Suit: int myNumber: int String[] numString
Card(suit,aNumber): getNumber() toString()

Deck
card[] myCard numCards: int
shuffle(): void printDeck(int numToPrint): void dealNextCard()

Player
String name Card[] hand numCards: int maxCardNum: int wagerTotal: int getWager: int
emptyHand(): void addCard: Boolean getHandSum(): int printHand: boolean

Game
main

Suit (enum)
Hearts Diamonds Spades Clubs