Project 1

Title: Blackjack

Course: CSC-5

Section: 46023

Due Date: 7/21/14

Author: Krishen Wadhwani

**1 - Introduction:**

Blackjack is a simple game that has become a common card game over the past few centuries. This game originated centuries ago with Miguel de Cervantes, who is most famous for writing *Don Quixote*. In another book, his plot includes a card game where the aim is achieve a total of 21 from a deck of cards. Ever since then, the game has evolved and has been introduced as one of the leading games that numerous casino’s utilize to attract tourists and gamblers in hopes of making some profit.

Often times though, many people do enjoy the game itself, and are not inspired by the potential to make money off of a simple card game. One opportunity to play and practice decision making for this game is through this program.

**2 – Game Play and Rules:**

The game of blackjack is fairly simple. In the game, the player is rivaling the dealer, which means that regardless of the number of players in the game, the only contest is between the dealer and each individual player. The first step is that the player and the dealer draws two cards each from a deck of 52 cards. When the cards are randomly

Then, according to the type of card that is drawn (king, 2, ace), a value is assigned to that card. The following table demonstrates how the types of the cards correlate with the values of the cards:

|  |  |
| --- | --- |
| Card Name | Card Value |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| 7 | 7 |
| 8 | 8 |
| 9 | 9 |
| 10 | 10 |
| Jack | 10 |
| Queen | 10 |
| King | 10 |
| Ace | 1 or 11 |

As shown above, the number cards represent the value that they are worth, whereas the face cards are all worth 10 points. Finally, the ace has an option available for how much it is worth. It can either be given a value of 1 or 11 points, depending on what the total value of the person’s hand is.

After the initial two cards are given, the values of the hand (the person’s cards) is calculated. The whole potential of the goal is to achieve a total point value of 21. If the person (dealer or player) gets a 21 total point value in the first hand, they have a blackjack and automatically win the game. If neither person has such a value, they are given the opportunity to draw another card, or “hit.”

Altogether, the total of the first two cards can never reach 21, even if the two are both ace cards (seeing as one can be an 11 while the other is a 1, for a total of 12). After the third card is drawn, this opportunity to reach over 21 becomes available. The next card is added to the original point value of the initial two cards. If either the dealer or the player reaches over 21, he/she will “bust” and automatically loses. Both the dealer and the player may hit as many times as desired, but in most situations neither will hit more than four or five times.

In most casinos and places where blackjack is played at commonly, the dealer is required to hit if their total card value is less than 17. The reason for this requirement is so dealer has the greatest advantage to win if he/she has more than 17 and there will be a larger chance to win if hitting with a number below 17.

Finally, after both people have drawn their hands, the two values are compared. If the dealer or player has a blackjack initially, as stated above, they automatically win. In the case where the dealer and player have the same hand, it is generally accepted that if the hand is above 17 points, that it will be counted as a draw. If the hands are below 17, the game will be counted as a loss for the player. If both the dealer and player bust, then the game is counted as a draw. Finally, if neither player nor the dealer achieves a 21 point value, the higher point value wins the game.

**3 - Key Features**

There were several key features that have been useful in the development of the program. Certain loops, statements and variables were accordingly used to make the program as efficient as possible while still accomplishing its goal.

**3.1 – Variable Usage**

**3.1.1 – Short**

Since the majority of the variables used to develop the program were whole numbers that existed between 1 and 35, it was unnecessary to use any variable of larger size value. Since the short allocates 2 bytes of space while allowing the number to be greater than 60,000, it was useful to include the numbers using shorts.

Lines Used: 28-50

**3.1.2 – Char**

The char variable is a variable that allows the usage of numbers as well as other characters. This was an important variable considering that a stack of cards includes not only number cards, but also various face value cards, such as the king, queen, jack, ace, and ten. The number ten, although it can be seen as an integer as well, was two characters, and thus it had to be represented with the letter “t.”

Lines Used: 31, 41

**3.1.3 – String**

Strings were only used once throughout the entirety of the project, for the hit feature. It allowed the user to input yes or no when deciding whether or not to hit to pick up a new card.

Lines Used: 44

**3.2 – Decision/Loop Constructs**

**3.2.1 – If/Else if/Else Statements**

“If/Else if/Else” statements were one of the most useful constructs that were used. This decision statement allowed for values to be checked. The first place where they were used was when correlating the randomly assigned value to a specific card name. When randomly assigning the value, the number 1 was correlated to an ace, and the two to another two. This repeated until the face values, where an 11 was correlated to jack, a 12 to a queen, and a 13 to a king.

In addition to being used to correlate the random number to a card, the blackjack value of the card was correlated the card name. In this case, the face cards were given values of 10, the ace was automatically assigned with a value of 11 (which can be changed later), and the number cards were given their respective number.

The “if” statement was also used alone in order to help when deciding to change the ace value. The statement checked to see if the total point value was over 21, and then to check if there was an ace that was randomly drawn during the game. If there were two ace cards initially drawn, then once ace value would automatically be set to one.

Another place where the “if/else” statement was used was to see if the player wanted to hit again, which when then accordingly allow the loop to be run again until the hit value was made to be a no.

The final place for which and “if/else if” statement was utilized was to determine the winner of the game, by comparing the value of the dealer to the value of the player. The various cases (explained in the game rules above) describe the different possibilities and outcomes for the two values to be compared. Accordingly, the statement would be used to determine if the player wins, loses, or draws.

Lines Used: 79-107, 113 – 124, 145, 226, 260 – 266, 283

**3.2.2 – Do/While Loops**

The “do/while” loop that was best utilized for the case if the number of games played. This is because, rather than the while loop, the check to see if the loop should continue or not is at the end, which would allow the number of games to be checked after each game, and accordingly stop it after the last game.

Lines Used: 64, 333

**3.2.3 – While Loops**

The “While” Loop was useful across various cases throughout the program. This loop was first correlated to determine the card value for each name. The loop allowed the use of this determination for each of the card variables.

Another important place where it was useful was for the check if the dealer or player would like to hit. In the case of the dealer, the dealer was forced to hit if his total was below 17. Accordingly, the dealer hit as many times as it took until either he bust, achieved 21, or had a total greater than 17.

Later, when the player was hitting, the while loop was checked to see if it should run again if the player input “yes” for hit. At the end of the loop, the user would be prompted again to hit if his/her total was less than 21. Otherwise hit was set to an alternate value and the loop was terminated.

**3.2.4 – For Loops**

This loop was especially useful in the cases where the user was not required to input any number of value for the loop to continue. The first loop was used to assign card names to each of the values of the cards. In this program, an array was used for each card name/value and then they were randomly picked for each value of the array. Initially, 10 card values were determined, but only the first two were initially used for the hand. This was useful in that the memory for the variables in the array were not prone to random memory changes and it reduced the code required to draw another random card.

Later, the loop was used for determining how many times the ace check ran. This allowed for the check for an ace to run only the specific number of times the cards hit. If there were two initial cards and 2 additional hits, then the loop should have checked four total values for these four cards. This way, the ace won’t be accounted for if the ace were a number within the array that hasn’t been contributed to the total.

This loop within the program was used twice for each the player and the dealer.

Lines Used: 75, 149, 165, 252

**3.3 – Arrays**

Arrays were especially useful for this particular game. These special types of variable sets allowed for one variable to be used to represent numerous more variables. The main utilization of variables for this project was in the assignment of values to each card number. The multiple card numbers are assigned by incrementing the variable by incrementing the index value. This allows for the prevention of mass repetition of code for different variables. Without arrays, the project would have expected to be well over 800 lines of code, whereas with them it was only 350.

Lines Used: All throughout program

**3.4 – Comments**

Comments were the most useful portion of this project in that they allowed for much easier code implementation. They outlined each and every step in the process and explained the reason for which certain pieces of code were utilized.

Lines Used: All throughout program