Project 2

Title

Blackjack

Course

CSC-17A

Section

46097

Date

July 26th, 2019

Author

Brandon Sanchez

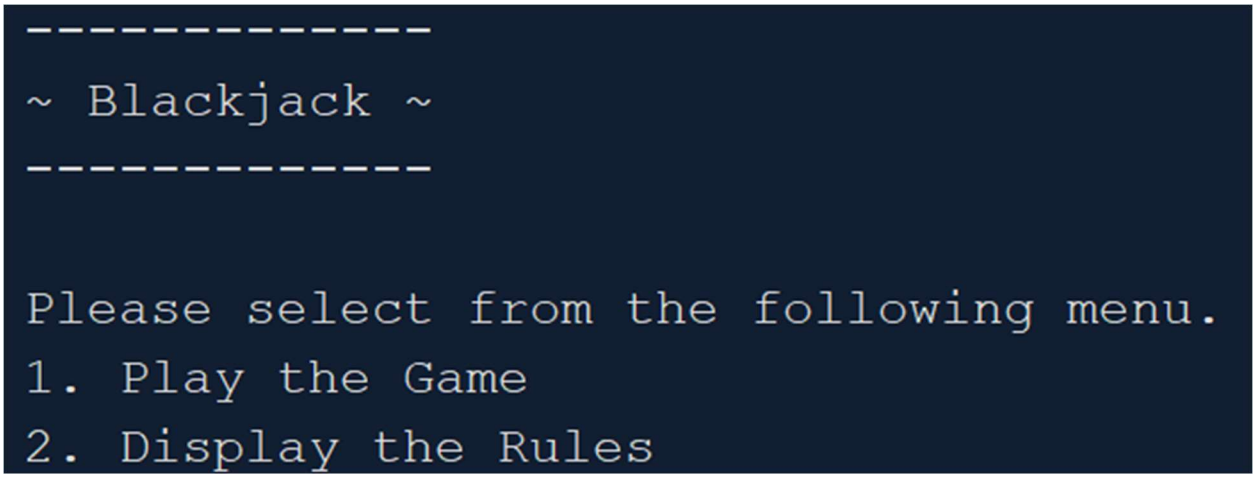
Introduction

Blackjack is a staple card game that is played across the world. It is simple enough for anyone to understand in one sitting yet nuanced enough for people to always come back for more. If my friends and I are playing a card game, there is a good chance that it will be Blackjack. The only table game I have played at a casino is Blackjack because that it how much I adore the game.

Rules

Available to see at the start of the game. Also, picture of rules included with sample IO.

Gameplay with Sample IO



The user is greeted with this menu when they start the game.

```
---------------------
~ Rules of BlackJack ~
---------------------
1. The game is played with a random number standard 52-card decks (No jokers).
2. The goal of the game is to have more points than the dealer but not exceed 21 points.
3. If either the player or dealer exceeds 21 points they 'bust' or lose the game.
4. The number on the card is the value in the game.
5. Face cards are worth 10 points and aces are worth 1 or 11.
6. The player and the dealer are both given 2 cards each.
7. At first the dealer only shows one card.
8. The player can either choose to stay (add no cards) or hit (add a card).
9. After the player stays the dealer shows their 2nd card.
10. If the dealer has 16 points or less, he draws until he busts or exceeds 16 points.
11. If no busts occur, the player wins with higher points than the dealer or loses with lower points than the dealer.
12. Same points = draw.
13. No doubles or splits.
```

If the user inputs 2 then it will display the rules.

```
Please enter your first name. Max 10 characters. Do not start with a number.
Brandon
```

The user then inputs their name within the guidelines.

```
-------------
Initial Cards
-------------
Dealer Card 1: Nine of Hearts (9)
Player Card 1: Five of Diamonds (5)
Player Card 2: Eight of Spades (8)
--------------
Initial Totals
--------------
Dealer Total: 9
Player Total: 13

Would you like to hit ('h') or stay ('s')?
```

The initial cards/totals are outputted, and the user decides if they want to hit or stay.

```
--------------------
New Card / New Total
--------------------
Player Card 3: Ten of Diamonds (10)
Player Total: 23
```

If the player hits, then it will display a new card and add it to their total.

```
Bust! You lost. Better luck next time.

Press Enter to See Player History.
```

In this case the player busted because their total was greater than 21.

```
You were dealt an Ace. Do you want it to be worth '1' or '11'?
```

If the player is dealt an ace, then they can choose the value.

```
Player Card 3: Ace of Hearts (11)
Player Total: 26

Your ace value has been changed to '1' to prevent busting.

--------------------
New Card / New Total
--------------------
Player Card 3: Ace of Hearts (1)
Player Total: 16
```

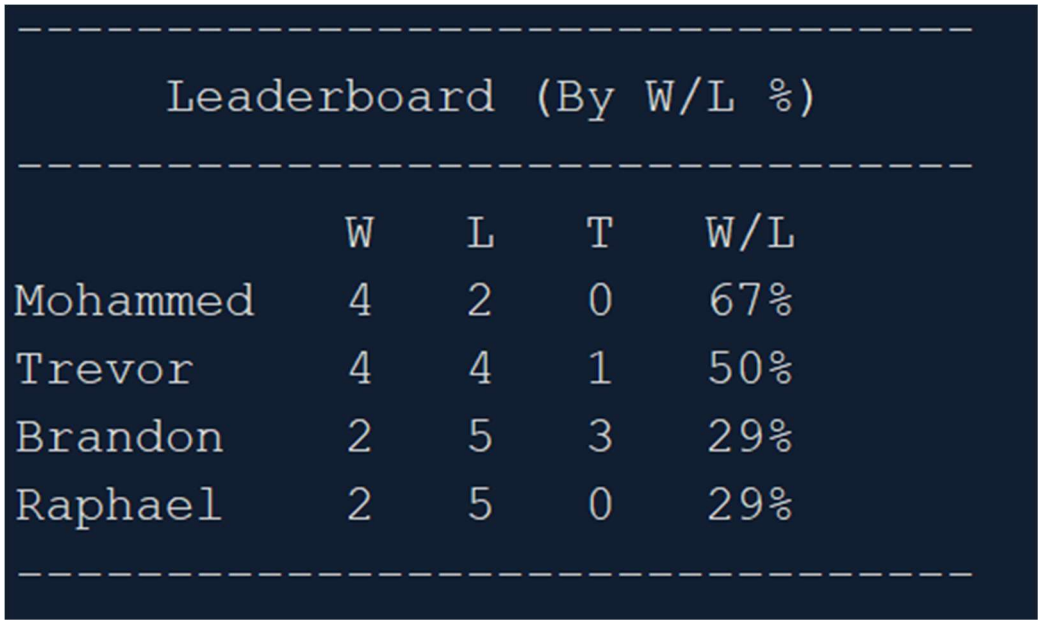
If the player has an ace worth 11 and they bust, the value will be reduced to 1 to prevent a bust.

```
------------
Card History
------------
Dealer Card 1: Nine of Spades (9)
Dealer Card 2: Three of Hearts (3)
Dealer Card 3: Eight of Clubs (8)
Player Card 1: Ten of Hearts (10)
Player Card 2: Five of Spades (5)
------------
Final Totals
------------
Dealer Total: 20
Player Total: 15

Dealer has the greater total. You lost. Better luck next time.
```

If the player doesn't bust, they will see this screen with the final results.

```
---------------
Player History
---------------
Wins:  2
Losses:  5
Ties:  3


-------------------------------------------
Would you like to play again? Type 'y' or 'n'.
```

Whether busting or not, the player gets to see their match history and is prompted to play again.



Once the player hits ‘n’ after playing however many times they like, the program will end with a leaderboard so the player can compare their results with others. Saves their stats in a file and will be remembered for later playthroughs.

Development Summary

Lines of Code: 1052

I decided to carry on with the same Blackjack game project because I enjoyed working on it in CSC-5 and for project 1. The first thing I set about doing was changing all of the structures into classes. This took the majority of the time. It seems simple to change them but there are a lot of little things that change between the two. I dealt with many glitches during the process that took a lot of time to fix. I think the way my classes are setup are pretty logical and the leaderboard class could be used for many other games. The player function that holds cards and stats could also be modified to work for other card games. My favorite part of the game is the leaderboard that keeps track of each player. If your name is already in the file it will add on to your already existing wins/losses/ties. After your first playthrough it will keep track of your stats and all you have to do is play again to have the stats saved. Sadly, gameplay wise nothing has changed. There is simply not enough time for me to add all the features I want to implement because changing the base code

takes so long. Hopefully if there is a game project in either CSC-17B or CSC-17C, I will continue with Blackjack and continue to make it better.

Future Goals

- Add doubles and splits
- Include a betting system
- Let user change ace value at any point

Pseudocode

Create classes.
Create function prototypes.
Set random number seed.
Declare variables and objects.

Fill leader object with player data from file.
Display menu.
If 1, play the game.
If 2, display rules then play the game.

Randomize dealer card 1.
If card = face card, card value = 10.
If card = ace, card value = 1.
Display dealer card 1.

Randomize player cards 1 & 2.
If card = face card, card value = 10.
If card = ace, card value = 1 or 11.
Display player cards 1 & 2.
If they are both worth 11, reduce 1 to a value of 1.

Display initial totals.
Ask for ‘hit’ or ‘stay’
If hit, repeat card process.
If bust, check for aces.
If there is an ace worth 11, reduce value
Else player loses, end game show history/prompt user to play again.
If stay, draw for dealer until bust or total > 16.

Display card history.
Display player/dealer totals.
Show who won.
Display player’s win history.
Prompt user to play again.

If user wants to play again loop them to start.
If user doesn’t want to play, output leaderboard and end run.