Blackjack

Introduction

Blackjack is a popular card game played between a player and a dealer. The objective of the game is to have a hand value as close to 21 as possible without exceeding it. The player competes against the dealer, and the player with a hand value closest to 21 without going over wins the round.

Design and Implementation

The code starts by initializing the deck, which consists of cards with different ranks and suits, along with their corresponding values. The deck is represented as a dictionary in the **deck** variable.

The game progresses by dealing two cards to both the player and the dealer. The **deal_cards()** function is responsible for this task. It selects two random cards from the deck and assigns them to the player and dealer. The selected cards are removed from the deck. The player's and dealer's cards are then stored in the **player_cards** and **dealer_cards** lists.

The player's total score is calculated based on the card values, and the information about the dealer's cards is partially revealed to the player. The dealers first card is not revealed to the player due to the rules of blackjack.

```
Preston Victory@DESKTOP-B8E0IDK MINGW64 ~/OneDrive/Desktop/NucampFolder/Python/1-Fundamentals/Blackjack Project
$ python blackjack.py

Welcome to the game of Blackjack!

What do you want the buy in to be? 1000

Good Luck!
Your total chip count is $1000.
How much would you like to bet this round? 50
The dealer cards are __ and 7 of Diamonds
Your cards are 6 of Diamonds and 5 of Spades with a total score of 11

What would you like to do? You can stay or hit?
```

The player is then given the option to either hit or stay. If the player chooses to hit, an additional card is drawn from the deck, and the total score for the player is updated accordingly. If the player's total score exceeds 21, they bust, and the dealer wins the round.

```
Welcome to the game of Blackjack!

What do you want the buy in to be? 1000

Good Luck!
Your total chip count is $1000.
How much would you like to bet this round? 50
The dealer cards are __ and 7 of Diamonds
Your cards are 6 of Diamonds and 5 of Spades with a total score of 11

What would you like to do? You can stay or hit? hit
You have drawn a Ace of Diamonds.
You now have a total of 22

You have busted. The dealer wins!

Would you like to play again?
```

If the player chooses to stay, the dealer's turn begins. The dealer draws cards until their total score is 18 or higher. If the dealer's total score exceeds 21, they bust, and the player wins the round. If both the player and dealer stay, their scores are compared, and the round winner is determined.

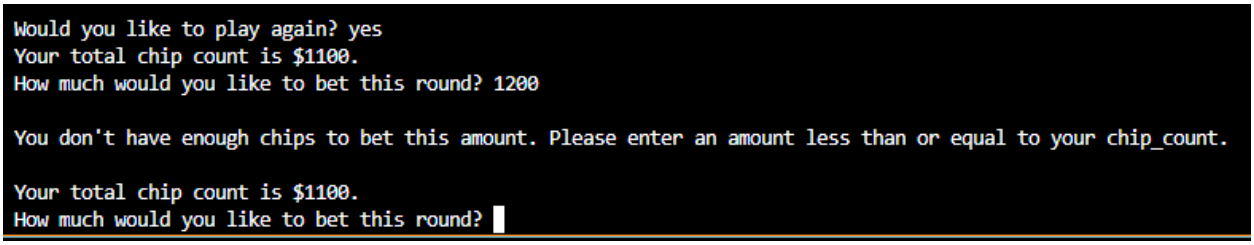
```
What would you like to do? You can stay or hit? stay
You have elected to stay.

The dealer has a total of 14 with Ace of Hearts and 3 of Spades.
The dealer has elected to hit and has drawn a 10 of Clubs.

The dealer now has a total of 24
The dealer has busted. You win!

Would you like to play again?
```

The code includes the functionality to allow the player to play multiple rounds and manage their chip count. The program will not let the player bet more than their chip count and the program ends when the chip count hits zero. The player is prompted to enter the buy-in amount and the bet amount for each round. The chip count is updated based on the outcome of each round.



Conclusion

Doing this project allowed be to build a game that could be playable from the ground up with very little instructions. It helped me learn to efficiently recall different concepts learned throughout the course and allowed me to work on my problem solving skills as a programmer. The best features of the project include being able to play as many rounds as you want as long as you have a chip count above 0 and that the dealer is able to be competitive to where the game is not one sided. I don't I would change anything in hindsight and do not plan to add any big additional features. I plan to complete my original portfolio idea Texas Holdem poker as that has a lot more concepts that are needed.