

Blackjack

Introduction

- In Blackjack, the aim is to beat the dealer (in this case, the computer) by having a higher card total and/or having a value of 21 or less.
- After the dealer is done with their turn, both parties show their hand and effectively their total card values. Whoever has the highest card value wins as long as it is under 21. If one has a value of over 21, they automatically lose.

Design and Implementation

- Initially, each party is dealt two hands. Additionally, you can only see one of the dealer's hands during the initial turn.

Let's play Blackjack!

Dealer's Hand:
---Hidden Card---
Ten of ♥Hearts♥

Your Hand:
Six of ♣Clubs♣
Nine of ♥Hearts♥

- After seeing your initial value, you have the option to either draw another card (a “hit”) or just hold the cards that you have in your hand (a “stand”)

```
def hit_or_stand(deck, hand):
    global playing

    while True:
        choice = input(
            '\nHit or Stand? (Enter "Hit" / "H" or "Stand" / "S"): '
        )
        if choice == 'Hit' or choice == 'H':
            hit(deck, hand)
        elif choice == 'Stand' or choice == 'S':
            print("You stand. Dealer's turn: ")
            playing = False
        else:
            print("Incorrect input. Please enter 'Hit' / 'H' or 'Stand' / 'S: ")
            continue
        break
```

- Once you stand, the dealer can “hit” if their total card value is less than 17. Once they reach or pass that value, the dealer cannot draw any more cards.

```
while dealer_hand.value < 17:  
    print("\nDealer hits again...\n")  
    hit(deck, dealer_hand)
```

- After the dealer is done with their turn, both parties show their hand and effectively their total card values. Whoever has the highest card value wins as long as it is under 21. If one has a value of over 21, they automatically lose.

Conclusions

- I personally learned how to implement functional skills into making a fun game. I also specifically learned about how one could use an asterisk to unpack an iterable into arguments in function calls.

```
def initial_deal(dealer, player):  
    print("Dealer's Hand:")  
    print("---Hidden Card---")  
    print(dealer.cards[1])  
    print("\nYour Hand:", *player.cards, sep="\n")
```

- For example, `*player.cards` passes all of the items that were appended into the player's hand list without knowing how many items were in the list.
- The best features of the project is the randomness of the dealing and the draws making every game seem unpredictable and fun. One shortcoming of the project is that it's currently only one player against the dealer and not “player vs player” or a group of players against the dealer.
- In hindsight, I would have found a way to make the card values in the shape of cards and not actual text.
- Some additional features I might want to add are enabling a player to go against another player, having multiple players go against the dealer, and having cards with the value instead of just plain text to make the game more immersive.