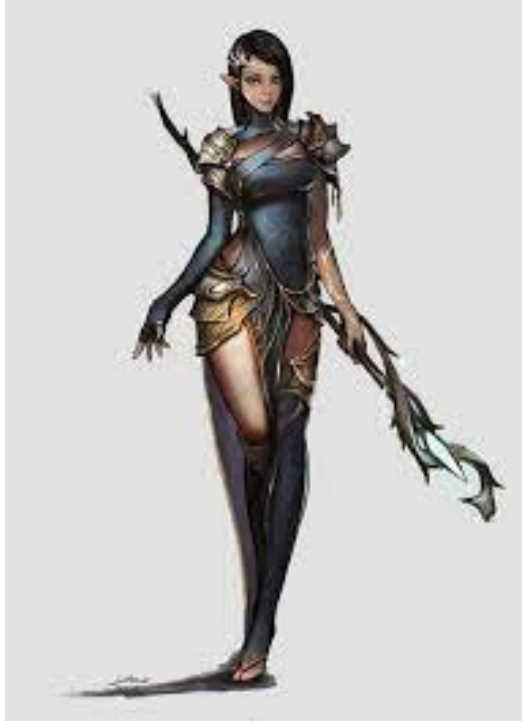
BLACKJACK GAME PRESENTATION

How we built our Blackjack game with Python

ABOUT US



GIANG

The worrior coder

SANDRA

The worrior designer

ABOUT BLACKJACK

- Blackjack is a card-based game played at casinos.
- The participants in this game compete with the dealer assigned by the casino.
- The magic number for Blackjack is 21.
- If a player gets an exact 21 = Win
- If a player gets above 21 = Loose

PLAYING PLAN

1. Create a deck of 52 cards
2. Shuffle the deck
3. Deal two cards to the Dealer and two cards to the Player
4. Show only one of the Dealer's cards, the other remains hidden
5. Show both of the Player's cards
6. Ask the Player if they wish to Hit, and take another card
7. If the Player's hand doesn't Bust (go over 21), ask if they'd like to Hit again.
8. If a Player Stands, play the Dealer's hand. The dealer will always Hit until the Dealer's value meets or exceeds 17
9. Determine the winner
10. Ask the Player if they'd like to play again

STEP BY STEP PLAN

Step 1: Imports and Global Variables

Step 2: Create a Card Class

Step 3: Create a Deck Class

Step 4: Create a Hand Class

Step 5: Write a function for taking hits

Step 6: Write a function prompting the Player to Hit or Stand

Step 7: Write functions to display cards

Step 8: Write functions to handle end of game scenarios

THE CODE

```
import random
suits = ('Hearts', 'Diamonds', 'Spades', 'Clubs')
ranks = ('Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Jack', 'Queen',
'King', 'Ace')
values = {'Two':2, 'Three':3, 'Four':4, 'Five':5, 'Six':6, 'Seven':7, 'Eight':8, 'Nine':9, 'Ten':10,
'Jack':10,
        'Queen':10, 'King':10, 'Ace':11}
playing = True
class Card:
    def __init__(self, suit, rank):
        self.suit = suit
        self.rank = rank
    def __str__(self):
        return self.rank + ' of ' + self.suit
```

THE CODE

```
i#creating Deck, shuffle function and single dealing
class Deck:
    def __init__(self):
        self.deck = []  # start with an empty list#
        for suit in suits:
            for rank in ranks:
                self.deck.append(Card(suit, rank))
    def __str__(self):
        deck_comp = '' #strating competition deck empty#
        for card in self.deck:
            deck_comp += '\n' + card.__str__() #add each card object;s strin#
        return 'The deck has' + deck_comp
    def shuffle(self):
        random.shuffle(self.deck)
    def deal(self):
        single_card = self.deck.pop()
        return single_card
```

THE CODE

```
i#creating a hand#
class Hand:
    def __init__(self):
        self.cards = []  # start with an empty list as we did in the Deck class
        self.value = 0   # start with zero value
        self.aces = 0    # add an attribute to keep track of aces
    def add_card(self,card):
        self.cards.append(card)
        self.value += values[card.rank]
        if card.rank == 'Ace':
            self.aces += 1
    def adjust_for_ace(self):
        while self.value > 21 and self.aces:
            self.value -= 10
            self.aces -= 1
```

THE CODE

```
# taking hits#
def hit(deck,hand):
    hand.add_card(deck.deal())
    hand.adjust_for_ace()
#player to take hits or stand#
def hit_or_stand(deck,hand):
    global playing
    while True:
        x = input("Would you like to Hit or Stand? Enter 'h' or 's'")
        if x[0].lower() == 'h':
            hit(deck,hand)  # hit() function defined above
        elif x[0].lower() == 's':
            print("Player stands. Dealer is playing.")
            playing = False
        else:
            print("Sorry, please try again.")
            continue
        break
```

THE CODE

```
#functions to display cards#
def show_some(player,dealer):
    print("\nDealer's Hand")
    print("<card hidden>")
    print(' ', dealer.cards[1])
    print("\nPlayer's Hand: ", *player.cards, sep= '\n')
def show_all(player,dealer):
    print("\nDealer's Hand:", *dealer.cards, sep="\n")
    print("Dealer's Hand =",dealer.value)
    print("\nPlayer's Hand: ", *player.cards, sep= '\n')
    print("Player's Hand = ", player.value)
def player_busts(player,dealer):
    print("Player busts!")
def player_wins(player,dealer):
    print("Player wins!")
def dealer_busts(player,dealer):
    print("Dealer busts!")
def dealer_wins(player,dealer):
    print("Dealer wins!")
def push(player,dealer):
    print("Dealer and Player tie! It's a push.")
```

THE CODE

```
while True:
    # Print an opening statement
    print("Welcome to my kickass Blackjack game.")
    # Create & shuffle the deck, deal two cards to each player
    deck = Deck()
    deck.shuffle()
    player_hand = Hand()
    player_hand.add_card(deck.deal())
    player_hand.add_card(deck.deal())
    dealer_hand = Hand()
    dealer_hand.add_card(deck.deal())
    dealer_hand.add_card(deck.deal())
    # Show cards (but keep one dealer card hidden)
    show_some(player_hand, dealer_hand)
    while playing:  # recall this variable from our hit_or_stand function
        # Prompt for Player to Hit or Stand
        hit_or_stand(deck, player_hand)
        # Show cards (but keep one dealer card hidden)
        show_some(player_hand,dealer_hand)
        # If player's hand exceeds 21, run player_busts() and break out of loop
        if player_hand.value >21:
            player_busts(player_hand, dealer_hand, player_chips)
            break
```

THE CODE

```
# If Player hasn't busted, play Dealer's hand until Dealer reaches 17
  if player_hand.value <= 21:
    while dealer_hand.value <17:
      hit(deck, dealer_hand)
    # Show all cards
    show_all(player_hand,dealer_hand)
    # Run different winning scenarios
    if dealer_hand.value > 21:
      dealer_busts(player_hand,dealer_hand)
    elif dealer_hand.value > player_hand.value:
      dealer_wins(player_hand,dealer_hand)
    elif dealer_hand.value < player_hand.value:
      player_wins(player_hand,dealer_hand)
    else:
      push(player_hand,dealer_hand)
```

THE CODE

```
# Ask to play again
  new_game = input("would you like to play again? Enter 'y' or 'n'")
  if new_game[0].lower() == 'y':
      playing = True
      continue
  else:
      print('Thanks for playing! ')
      break
```

THE GAME

Now lets play!

https://github.com/Gixi0612/IronhackLabs_GiangLe/blob/master/Module%201/Final%20Blackjack.ipynb

THANK YOU