This one is a Milestone Project from the Python Bootcamp course on Udemy. Here I made a card game of Black Jack in Python.

In [1]:
```
# Getting initial setup, random for future actions
import random
# Setting up the cards and card-value dictionary
suits = ('Hearts', 'Diamonds', 'Spades', 'Clubs')
ranks = ('Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'J
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}
```

In [2]:
```
# This will be the card class for each card that is in a hand
# It has its attributes and the string method that returns the full name of the car
class Card:
    def __init__(self,suit,rank):
        self.suit = suit
        self.rank = rank
        self.value = values[rank]
    def __str__(self):
        return self.rank+' of '+self.suit
```

In [3]:
```
# This will be the deck class for the player and dealer
# This is basically a list of cards that each one has
# Main methods are taking a card from the top of the deck and shuffling the deck
class Deck:
    def __init__(self):
        self.deck = [] # starting as an empty list so each instance is the same
        for suit in suits: # filling up the deck with cards
            for rank in ranks:
                self.deck.append(Card(suit,rank))
    def take_card(self):
        return self.deck.pop()
    def shuffle(self):
        random.shuffle(self.deck)
```

In [4]:
```
# Creating a class Person that can be either the player or dealer
# Dealer can be Person as a default
# Player needs additional methods and attributes
# Class Player is defined later and is going to inherit from this class
# The class needs aces as a sepearate attribute,
# because aces can have two values (one and eleven)
# So we can adjust for the aces in the hand as a method
class Person:
    def __init__(self):
        self.hand = [] # list of cards in the hand
        self.value = 0 # current value of the hand
        self.aces = 0 # number of aces
    def take_card(self,new_card):
        self.hand.append(new_card)
        self.value += values[new_card.rank]
        if new_card.rank == 'Ace':
            self.aces += 1
    def adjust_for_aces(self):
```

```
        while self.value > 21 and self.aces:
            self.value -= 10
            self.aces -= 1
```

In [5]:
```
# The player class with inheritance from class Person
# There are additional attributes and methods for the player
class Player(Person):
    def __init__(self):
        super().__init__()
        self.name = input('Choose a name: ')
        self.balance = 0 # initial balance
        self.bet = 0 # initial bet
    def top_up_balance(self,amount):
        self.balance += amount
    def win_bet(self):
        self.balance += self.bet
    def lose_bet(self):
        self.balance -= self.bet
```

In [10]:
```
# Defining some starting functions
def initialize(player):
    print(f'Welcome {player.name}! Currently you have ${player.balance} in your ban
    while True:
        amount = input('Enter the amount to top up your balance: ')
        try:
            if int(amount) > 0:
                amount = int(amount)
                break
            else:
                print('The amount has to be positive.')
        except:
            print('Please enter a valid amount.')
    player.top_up_balance(amount)
'''def ready(player):
    while True:
        ans = input(f'Well done {player.name}! Now you have ${player.balance} in yo
        if ans == 'Y':
            return True
        elif ans == 'N':
            return False
        else:
            print('Type the correct capital-case letter')'''
def bet(player):
    while True:
        try:
            player.bet = int(input('How much do you want to bet? '))
        except:
            'Enter an integer amount!'
        else:
            if player.bet < player.balance:
                print('Bet accepted')
                break
            else:
                print('Insufficient funds')
def show_first(player,dealer,deck): # showing the cards at the start
```

```
    for i in range(2):
        player.take_card(deck.take_card())
        print(f'You have {player.hand[i]}')
        dealer.take_card(deck.take_card())
    print(f'Dealer has {dealer.hand[1]}')
```

In [7]:
```
#Gameplay and end-game functions
def hit_player(player,deck):
    global round_on
    global round_num
    player.adjust_for_aces()
    while True:
        if player.value > 21: # immediately catching if player has too much
            print('You lose, you are busted')
            round_on = False
            round_num +=1
            player.lose_bet()
            show_last(player,dealer,deck)
            break
        ans = input(f'Do you wish to hit? Y/N ')
        if ans == 'Y':
            player.take_card(deck.take_card())
            player.adjust_for_aces()
            print(f'You took {player.hand[-1]}')
        elif ans == 'N':
            break
        else:
            print('Type the correct capital-case letter')
def hit_dealer(dealer,deck):
    dealer.adjust_for_aces()
    while dealer.value <17:
        dealer.take_card(deck.take_card())
        dealer.adjust_for_aces()
def show_last(player,dealer,deck): # showing cards at the end
    print("Player has: ")
    print(*player.hand, sep=', ')
    print("Dealer has: ")
    print(*dealer.hand, sep=', ')
```

In [19]:
```
# Beginninng the game with a setup
game_on = True
round_num = 1
player = Player()
initialize(player)
#start = ready(player)
# While the game is on - play the current game
while game_on:
    #if not start:
        #break
    dealer = Person()
    cur_deck = Deck()
    cur_deck.shuffle()
    round_on = True
    # Renew the player hand
    player.hand = []
```

```
    player.value = 0
    while round_on:
        print(f'This is round number {round_num}.')
        # Prompt for a bet and show the table
        bet(player)
        show_first(player,dealer,cur_deck)
        # Start the round proper
        hit_player(player,cur_deck)
        # Considering various scenarios
        p = player.value
        if p <= 21:
            # If player isn't busted then dealer can enter the round
            hit_dealer(dealer,cur_deck)
            d = dealer.value
            #The round is over and comparing starts
            show_last(player,dealer,cur_deck)
            if d > 21: # dealer has too much
                print('You won, dealer is busted')
                round_on = False
                round_num +=1
                player.win_bet()
            elif d < p:
                print('You won')
                round_on = False
                round_num +=1
                player.win_bet()
            elif d == p:
                print('This is a draw/push')
                round_on = False
                round_num += 1
            elif d > p:
                print('You lost')
                round_on = False
                round_num +=1
                player.lose_bet()
            else:
                round_on = False
                round_num +=1
                print('Unexpected result')
        else: # that is the player has too much, which we caught before
            break
    # Info after a round is over
    print('Your balance is: ',player.balance)
    cont = input('Do you want to continue? Y/N ')
    if cont == 'N':
        game_on = False
```

```
Choose a name: name
Welcome name! Currently you have $0 in your bank, you can top up
Enter the amount to top up your balance: 10
This is round number 1.
How much do you want to bet? 1
Bet accepted
You have Five of Spades
You have Seven of Spades
Dealer has Ace of Diamonds
Do you wish to hit? Y/N Y
You took Three of Hearts
Do you wish to hit? Y/N N
Player has:
Five of Spades, Seven of Spades, Three of Hearts
Dealer has:
King of Clubs, Ace of Diamonds
You lost
Your balance is:  9
Do you want to continue? Y/N Y
This is round number 2.
How much do you want to bet? 2
Bet accepted
You have Two of Diamonds
You have Three of Spades
Dealer has Jack of Clubs
Do you wish to hit? Y/N Y
You took Eight of Diamonds
Do you wish to hit? Y/N Y
You took Ace of Hearts
Do you wish to hit? Y/N N
Player has:
Two of Diamonds, Three of Spades, Eight of Diamonds, Ace of Hearts
Dealer has:
Queen of Spades, Jack of Clubs
You lost
Your balance is:  7
Do you want to continue? Y/N Y
This is round number 3.
How much do you want to bet? 1
Bet accepted
You have Two of Hearts
You have King of Hearts
Dealer has Jack of Diamonds
Do you wish to hit? Y/N Y
You took Seven of Clubs
Do you wish to hit? Y/N N
Player has:
Two of Hearts, King of Hearts, Seven of Clubs
Dealer has:
Five of Diamonds, Jack of Diamonds, Queen of Hearts
You won, dealer is busted
Your balance is:  8
Do you want to continue? Y/N N
```