

BlackJack

by Maddy Fung



How to play

Guidelines:

- each player is given 2 cards
- the Dealer is given 2 cards, one face up the other face down/hidden
- each player is playing against the Dealer
- on each turn a player can either “Hit” or “Stay” accordingly so that they can get the closest to 21 without going over
- Ace and Face cards have special values
- the Dealer reveals all cards once everyone has gone and keeps drawing cards as long as the Dealer’s hand is > 17

Main objective:

- closest to 21 wins!

Most powerful hand:

Face card + Ace =

BlackJack

(has a value of 21)

Remember:

- Ace cards can be either 1 or 11
- Face cards are worth 10

“Hit” : draw another card and add it to your hand

“Stay” : do not draw any more cards

Let's Play BlackJack!

link for code: <https://codehs.com/sandbox/id/blackjack-qWqy8K>





Components: Card Class

```
class Card:
    def __init__(self, suit, value):
        self.suit = suit
        self.value = value
        self.valueInt = 0
        self.face = False

        if self.value == 'Jack' or self.value == 'Queen' or self.value == 'King':
            self.face = True
            self.valueInt = 10
        elif self.value == 'Ace':
            self.valueInt = 1
        else:
            self.valueInt = int(self.value)

    def __str__(self):
        return f'{self.value} of {self.suit}'

    def setValue(self, num):
        self.valueInt = num
```

Player Class

```
class Player:
    def __init__(self, number):
        self.number = number + 1 #int
        self.blackjack = False
        self.handVal = 0
        self.bust = False
        self.hand = [] #hand list of cards
```

- the Dealer is a Player object

```
def drawCard(self):
    num = random.randint(0, len(deck)-1)
    card = deck[num]
    self.hand.append(card)
    deck.remove(card)
    self.calculateHandValue()
    return card

def blackjack(self): #checks if blackjack
    if (self.hand[0].value == 'Ace' and self.hand[1].valueInt == 10 )
    | pr (( self.hand[0].valueInt == 10 ) and (self.hand[1].value == 'Ace')):
        self.blackjack = True

def setHandVal(self, num):
    self.handVal = num

def calculateHandValue(self):
    val = 0
    for i in self.hand:
        val = val + i.valueInt
    self.handVal = val
    return self.handVal

def hit(self):
    c = self.drawCard()
    print(f'          {c}')
    if self.hand[-1].value == 'Ace':
        aceVal = int ( input('What is the value of the ace?') )
        self.hand[-1].valueInt = aceVal

def isbust(self):
    if self.handVal < 21:
        self.bust = True
    else:
        self.bust = False
```



Planning out the code

1. create the Card objects and add them to a list called deck[]
2. add players to list called playerList
3. Dealer takes their initial 2 cards
4. go around the table for each Player's turn
 - a. each Player is drawn their initial 2 cards. If there is one Face card and one Ace then Player has Blackjack and is added to a list blackJack[]
 - b. while loop that keeps running as long as the player's hand value is under 21
 - c. Player is given choice of "hit" or "stay" and the while loop breaks if the player busts (goes over 21)
5. Players who did not bust are added to list notBusted[]
6. Dealer reveals both cards and keeps drawing cards until their hand is >17 or busts
7. Results
 - a. if there are players in blackJack[] then they are printed as BlackJack winners and bets are doubled as long as the Dealer does not also have Blackjack
 - b. if Dealer did bust, then print all the Players that did not bust (winners)
 - c. if Dealer didn't bust, print all the winners that had a higher hand value than Dealer



Difficulties while coding

- keeping track of the Ace cards
- using the Player class methods in different ways for the Dealer
- knowing where to calculate the total hand value

Future improvements

- using a visual developer tool like Unity
- implement all exceptions and fix any existing bugs
- implementing all special rules
 - ex. Splitting Hands, the other player's results affecting an individual player's rewards