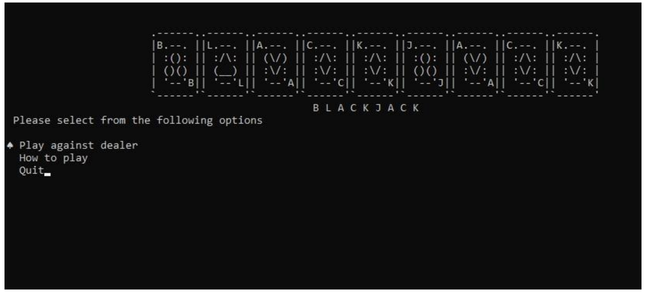
Blackjack

Created by Erkan Kaban

Walk-Through of Terminal app

After running the app in your terminal, you are introduced to a menu screen, this menu screen uses a external package pypi named "pick". Uses directional keys with a spade icon to show you which item you are and simply hit enter to quit, how to play or play against dealer.

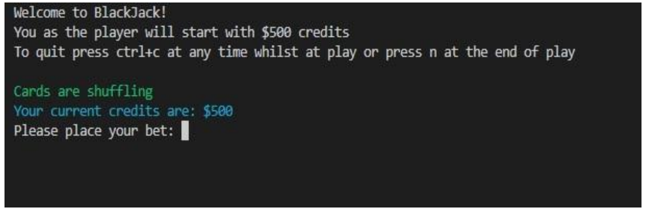
Main Menu Feature



```
.------..------..------..------..------..------..------..------..------.
|B.--. ||L.--. ||A.--. ||C.--. ||K.--. ||J.--. ||A.--. ||C.--. ||K.--. |
| :(): || :/\: || (\/) || :/\: || :/\: || :(): || (\/) || :/\: || :/\: |
| ()() || (__) || :\/: || :\/: || :\/: || ()() || :\/: || :\/: || :\/: |
| '--'B|| '--'L|| '--'A|| '--'C|| '--'K|| '--'J|| '--'A|| '--'C|| '--'K|
`------'`------'`------'`------'`------'`------'`------'`------'`------'
                              B L A C K J A C K
Please select from the following options

♠ Play against dealer
  How to play
  Quit
```

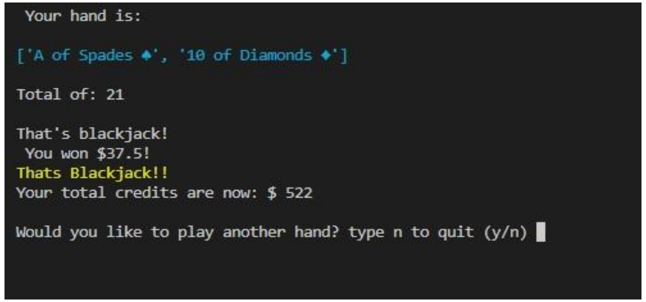
After you hit enter in the menu "Player against dealer" you are presented by the game, it presents some information to let you know what to do. In the Game Start we have a starting credit of 500 and we need to enter a value between 1-500

Bet Feature



```
Welcome to BlackJack!
You as the player will start with $500 credits
To quit press ctrl+c at any time whilst at play or press n at the end of play

Cards are shuffling
Your current credits are: $500
Please place your bet:
```

When a bet is placed, you are introduced to the cards you are dealt, with a total. If you hit 21 on your first two cards you win automatically gaining 2.5 times what your bet amount was.

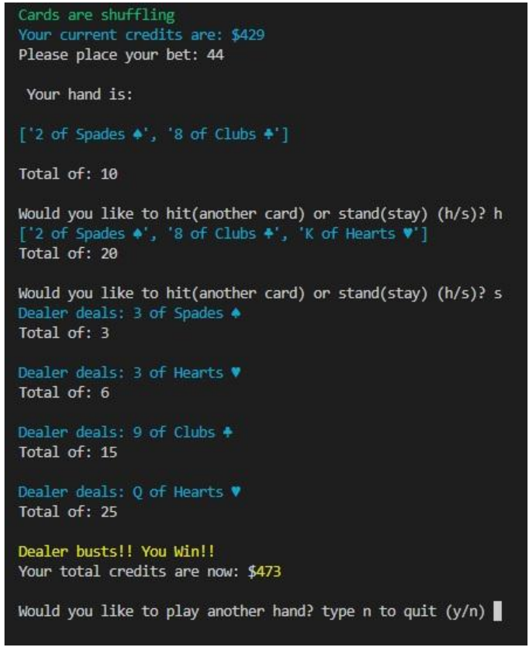


```
 Your hand is:

['A of Spades ♠', '10 of Diamonds ♦']

Total of: 21

That's blackjack!
 You won $37.5!
Thats Blackjack!!
Your total credits are now: $ 522

Would you like to play another hand? type n to quit (y/n)
```

Walk-Through - continuation

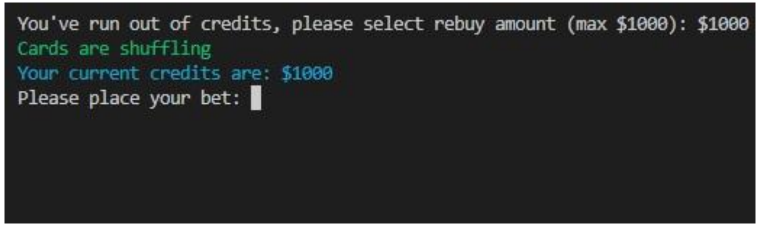
The **hit feature** - when you select h you are dealt a singular card this time and the total is added up at every sequence. If you reach a total of over 21 you are automatically lose.

The **stand feature** Once you are happy with the cards you have you hit the key 's' to stand, this is when the dealer plays up to a value of 17. The game compares your value to dealers and calculates winning or losses depending on the game at hand.

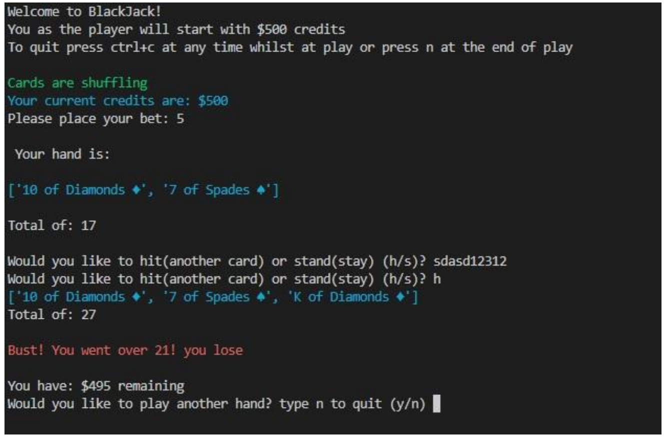


```
Cards are shuffling
Your current credits are: $429
Please place your bet: 44

 Your hand is:

['2 of Spades ♠', '8 of Clubs ♣']

Total of: 10

Would you like to hit(another card) or stand(stay) (h/s)? h
['2 of Spades ♠', '8 of Clubs ♣', 'K of Hearts ♥']
Total of: 20

Would you like to hit(another card) or stand(stay) (h/s)? s
Dealer deals: 3 of Spades ♠
Total of: 3

Dealer deals: 3 of Hearts ♥
Total of: 6

Dealer deals: 9 of Clubs ♣
Total of: 15

Dealer deals: Q of Hearts ♥
Total of: 25

Dealer busts!! You Win!!
Your total credits are now: $473

Would you like to play another hand? type n to quit (y/n)
```

Rebuy feature If you run out of credits and play another hand, the game will ask you if you'd like to rebuy, and allows you to enter an input from 1-1000 as your rebuy credits.



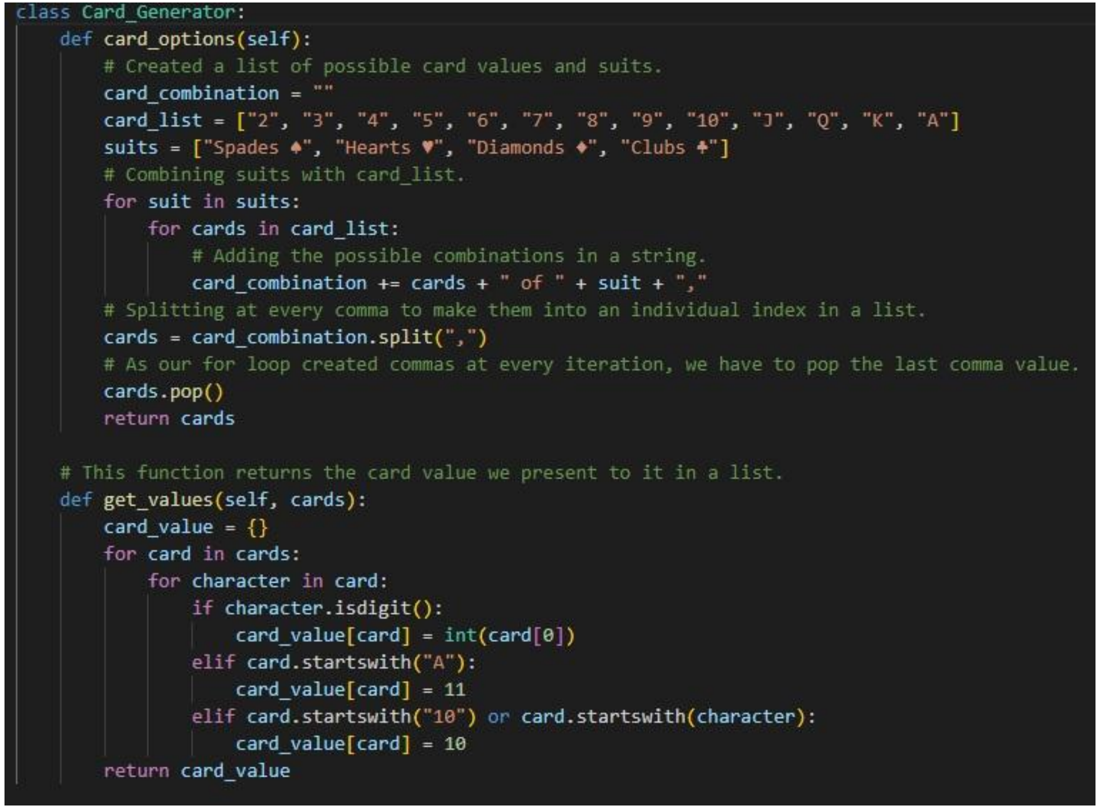
```
You've run out of credits, please select rebuy amount (max $1000): $1000
Cards are shuffling
Your current credits are: $1000
Please place your bet:
```



```
Welcome to BlackJack!
You as the player will start with $500 credits
To quit press ctrl+c at any time whilst at play or press n at the end of play

Cards are shuffling
Your current credits are: $500
Please place your bet: 5

 Your hand is:

['10 of Diamonds ♦', '7 of Spades ♠']

Total of: 17

Would you like to hit(another card) or stand(stay) (h/s)? sdasd12312
Would you like to hit(another card) or stand(stay) (h/s)? h
['10 of Diamonds ♦', '7 of Spades ♠', 'K of Diamonds ♦']
Total of: 27

Bust! You went over 21! you lose

You have: $495 remaining
Would you like to play another hand? type n to quit (y/n)
```

In this case, we hit and went over 21 automatically losing the game.

Walk-through - logic of terminal app

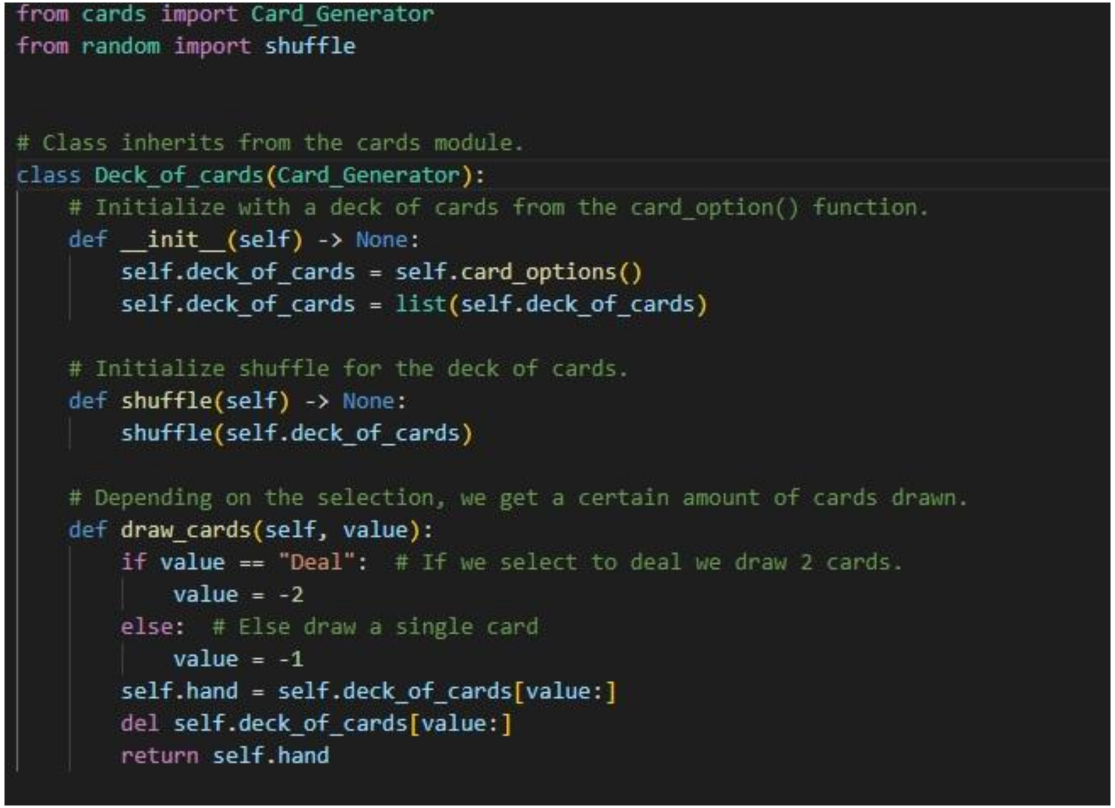


```
class Card_Generator:
    def card_options(self):
        # Created a list of possible card values and suits.
        card_combination = ""
        card_list = ["2", "3", "4", "5", "6", "7", "8", "9", "10", "J", "Q", "K", "A"]
        suits = ["Spades ♠", "Hearts ♥", "Diamonds ♦", "Clubs ♣"]
        # Combining suits with card_list.
        for suit in suits:
            for cards in card_list:
                # Adding the possible combinations in a string.
                card_combination += cards + " of " + suit + ","
        # Splitting at every comma to make them into an individual index in a list.
        cards = card_combination.split(",")
        # As our for loop created commas at every iteration, we have to pop the last comma value.
        cards.pop()
        return cards

    # This function returns the card value we present to it in a list.
    def get_values(self, cards):
        card_value = {}
        for card in cards:
            for character in card:
                if character.isdigit():
                    card_value[card] = int(card[0])
                elif card.startswith("A"):
                    card_value[card] = 11
                elif card.startswith("10") or card.startswith(character):
                    card_value[card] = 10
        return card_value
```

The first function card_options, I firstly needed to get the cards in a list/dictionary, I did this by using nested loops for the program to add each suits with each number/face card into a list and returning this variable cards to the program to contain them in a list.

Then continued on with another function to get the actual values of each card. I did this by applying all the cards in a list to this function named "cards" and with a series of if elif statements, we work out what each element in the list contained and returning the card value to present in the game.

Walk-through - logic of terminal app - continued



```python
from cards import Card_Generator
from random import shuffle


# Class inherits from the cards module.
class Deck_of_cards(Card_Generator):
    # Initialize with a deck of cards from the card_option() function.
    def __init__(self) -> None:
        self.deck_of_cards = self.card_options()
        self.deck_of_cards = list(self.deck_of_cards)

    # Initialize shuffle for the deck of cards.
    def shuffle(self) -> None:
        shuffle(self.deck_of_cards)

    # Depending on the selection, we get a certain amount of cards drawn.
    def draw_cards(self, value):
        if value == "Deal":  # If we select to deal we draw 2 cards.
            value = -2
        else:  # Else draw a single card
            value = -1
        self.hand = self.deck_of_cards[value:]
        del self.deck_of_cards[value:]
        return self.hand
```

Next we create another module, this time importing the cards class we explained earlier. We also create a class here with the card generator inherited as the values. This class was mainly created to handle the shuffling of the cards and when we draw cards whether we'd like to draw a single card or two, and to also physically remove the cards from the deck of cards.

The remaining, menu, blackjack_class and blackjack_main game I will explain in more detail in the presentation video as it's to long to display in a slide.

The blackjack class is where all the gameplay and logic happens and was separated to main_game module to help assist doing further tests to the blackjack class and its functions.

blackjack_main creates an instances of the blackjack class and runs the game for us.

Challenges, ethical issues and favourite parts

Challenges:

- Creating the card generator, and shuffle.
- Breaking everything down, simplifying.
- Once I got it working, getting everything and repeating things less.
- Tying all my classes functions and main menu together.

Future releases:

- Have a visual of the cards being dealt rather than having them displayed in text.
- Having each card coloured
- Having more than one player, play at once.
- Having a double feature, to double your bet during a game.

Ethical Issues:

During the planning phase of my program I did notice that there was a lot of packages that I could've used for my program that would've essentially did everything for me. So I only imported a very specific singular function and did the rest on my own, as not doing this would defeat the purpose of the project,

Not using too many external packages in regards to my blackjack program as there was several copies out there. Keeping it to a minimum.

Favourite parts:

- Creating and generating cards
- Having the player features to be able to work
- Having a dealer play automatically and compare with player
- Getting everything functioning to a level