Introduction:

The first semester of Binus International we were though about Python as our first programming language. Were though under the guidance of sir Bagus, we learn python for about 6 month, every meeting is really important and very educational. It was really hard because it a new thing for me, but with the help of sir Bagus and my friend I can understand.

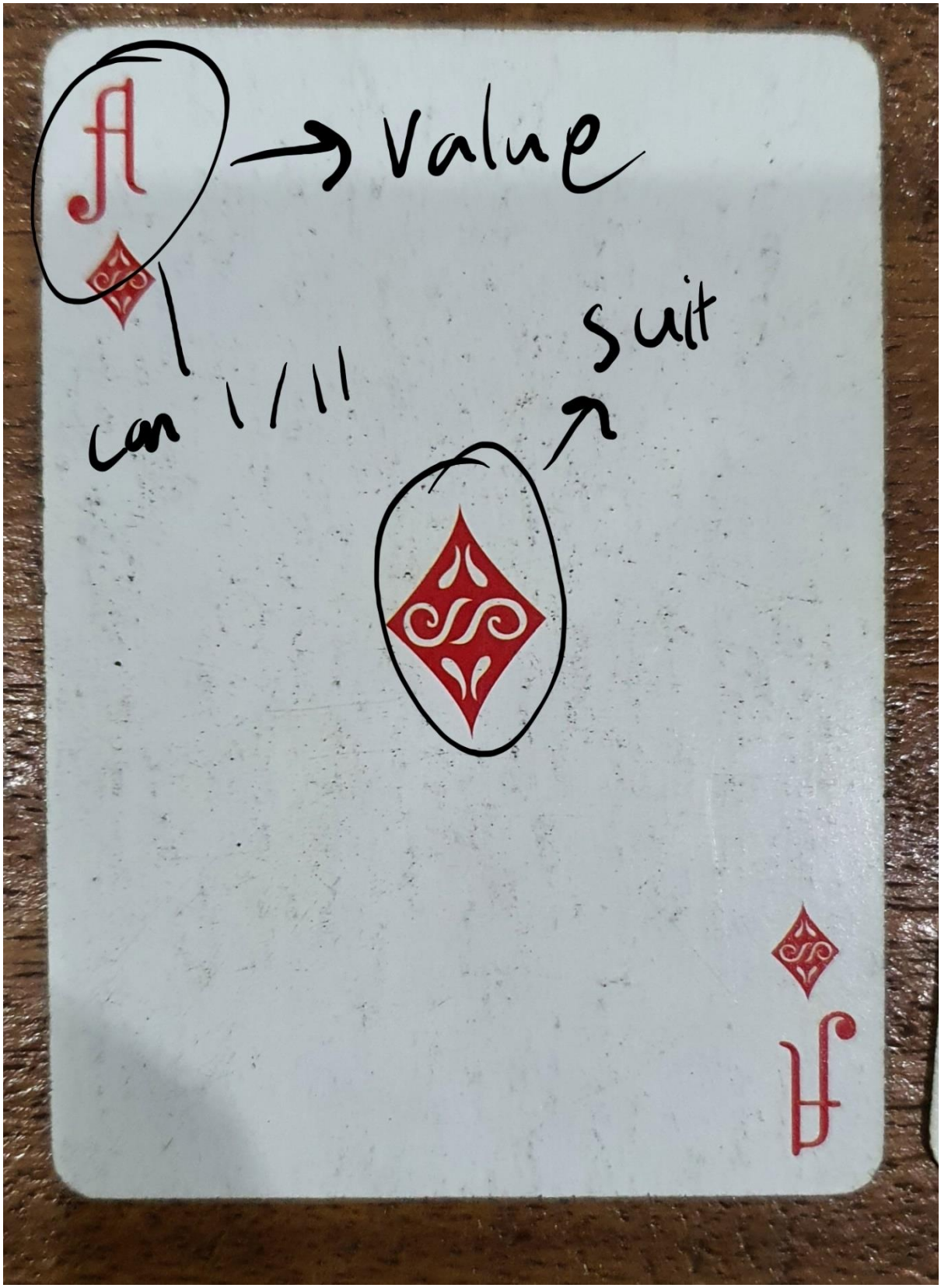
Project Specification:

With this come the end project, which is an individual project, we are need to make a not quite simple program , We need to use all the thing that are already been teach. One of my weakness are not really understand about OOP and classes, with this in mind I'm making a Blackjack. The rule of the game is getting to 21 value and don't go over board 21. The project start at 27 December 2020, the process was really hard.

The project of blackjack is for people who want to try blackjack in the form of Python. For me it's for helping me learn about class more and to be able to solve problem. The game of blackjack must run.

Solution Design :

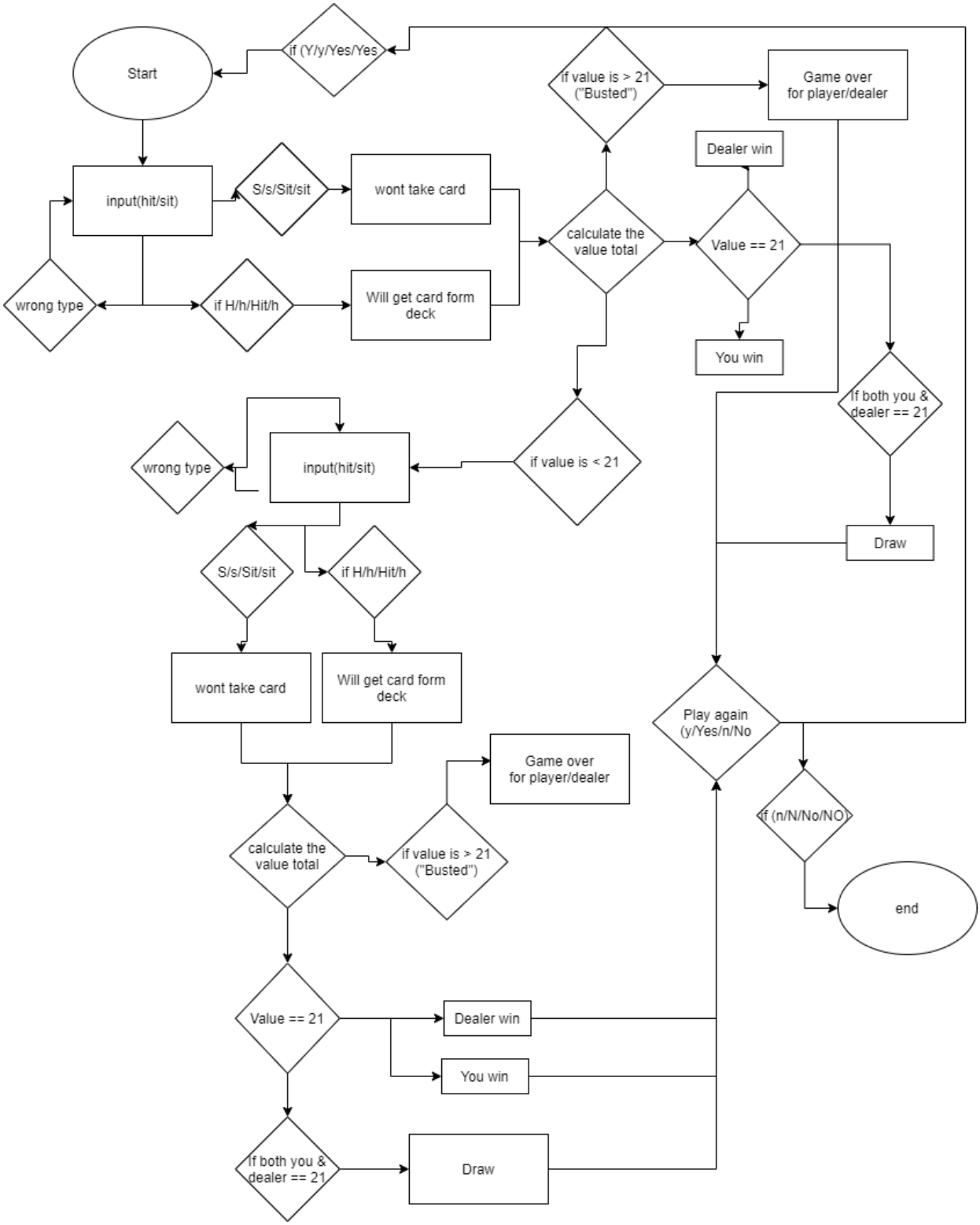
The program will be in 1 file and to play the game will need python any version, The game will be offline (you vs GPU) the objective is to be lucky and get 21 value of card first. Ace will give you 1 or 11 (automatic). Jack, Queen , and king will give 10. The plan is to make a class for card, deck, hand , and the game.



First step of making the class for card need to input value and the suit of the card

After making the card class , now we need to make the deck which consist of 52 card.

When programming BlackJack most important thing is that the dealer hard we can only see 1 card, and we need to make a away so that we can see our card to.



This flowchart is the idea of the code

The Fuction :

```
class Card:
    def __init__(self, suit, value):
        self.suit = suit
        self.value = value

    def __repr__(self):
        return " of ".join((self.value, self.suit))
```

- This is the class card that we will have to make before the deck class. In this class the card will have 2 part, Value (numeric number 1,2,3,4,5,6 and soon) , suit is the symbol on the card.

```
    def __repr__(self):
        return " of ".join((self.value, self.suit))
```

- This is for printing the right order, first come out “value” of “suit”

```
# inside the card deck
class Deck:
    def __init__(self):
        self.cards = [Card(s, v) for s in ["Spades", "Clover",
         "Hearts", "Diamonds"] for v in
                        ["A", "2", "3", "4", "5", "6",
                         "7", "8", "9", "O", "J", "Q", "K"]]

    def shuffle(self):
        if len(self.cards) > 1:
            RD.shuffle(self.cards)

    def deal(self):
        if len(self.cards) > 1:
            return self.cards.pop(0)
            # delet card that have been taken
```

- This deck class need to add the card value and suits
- Then we import random to shuffel the card

- The deal is to delet the card everytime we take a card, so there won't be any duplicates

```python
# you the player hand and the dealer hand
class Hand:
    def __init__(self, dealer=False):
        self.dealer = dealer
        self.cards = []
        # empty list
        self.value = 0

    def add_card(self, card):
        self.cards.append(card)

    def Calcu_Value(self):
        self.value = 0
        has_ace = False
        for card in self.cards:
            if card.value.isnumeric():
                self.value += int(card.value)
            else:
                if card.value == "A":
                    has_ace = True
                    self.value += 11
                else:
                    self.value += 10

        if has_ace and self.value > 21:
            self.value -= 10
            #making ace 1

    def get_value(self):
        self.Calcu_Value()
        return self.value

    def display(self):
        if self.dealer:
            print("hidden")
            print(self.cards[0])
        else:
            for card in self.cards:
                print(card)
            print("Value:", self.get_value())
```

- The class hand is to make a list for the hand
- First we make the value of our hand is 0 so we can add
- Def calcu_value is to calculate the total value of the card in our hands, it will also auto calculate if we have Ace, to change if we have < 21 will be automaticly convert to 11,but if it's > 21 then the card will be 1. The J,Q, and K will be 10
- Def display is to display the value of our card, and to only display 1 card of the dealer.

Edsel Putra Harren /2440081942/L1BC

```
class Game:
    def __init__(self):
        pass

    def play(self):
        playing = True

        while playing:
            self.deck = Deck()
            self.deck.shuffle()

            self.player_hand = Hand()
            self.dealer_hand = Hand(dealer=True)

            for i in range(2):
                self.player_hand.add_card(self.deck.deal())
                self.dealer_hand.add_card(self.deck.deal())

            print("Your hand is:")
            self.player_hand.display()
            print("Dealer's hand is:")
            self.dealer_hand.display()

            game_over = False

            while not game_over:
                player_has_blackjack, dealer_has_blackjack = self.check_for_blackjack()
                if player_has_blackjack or dealer_has_blackjack:
                    game_over = True
                    self.show_blackjack_results(player_has_blackjack, dealer_has_blackjack)
                    continue

                choice = input("Please choose [Hit / stay] ").lower()
                while choice not in ["h", "s", "hit", "stay"]:
                    choice = input("Please enter 'hit' or 'stay' (or H/S) ").lower()
                if choice in ['hit', 'h']:
                    self.player_hand.add_card(self.deck.deal())
                    self.player_hand.display()
                    if self.player_is_over():
                        print("You have lost!")
                        game_over = True
                else:
                    player_hand_value = self.player_hand.get_value()
                    dealer_hand_value = self.dealer_hand.get_value()

                    print("Final Results")
                    print("Your hand:", player_hand_value)
                    print("Dealer's hand:", dealer_hand_value)
```

```
                    if player_hand_value > dealer_hand_value:
                        print("You Win!")
                    elif player_hand_value == dealer_hand_value:
                        print("Tie!")
                    else:
                        print("Dealer Wins!")
                    game_over = True

                again = input("Play Again? [Y/N] ")
                while again.lower() not in ["y", "n"]:
                    again = input("Please enter Y or N ")
                if again.lower() == "n":
                    print("Thanks for playing")
                    playing = False
                else:
                    game_over = False

    def player_is_over(self):
        return self.player_hand.get_value() > 21

    def check_for_blackjack(self):
        player = False
        dealer = False
        if self.player_hand.get_value() == 21:
            player = True
        if self.dealer_hand.get_value() == 21:
            dealer = True

        return player, dealer

    def show_blackjack_results(self, player_has_blackjack, dealer_has_blackjack):
        if player_has_blackjack and dealer_has_blackjack:
            print("Both players have blackjack! Draw")

        elif player_has_blackjack:
            print("You have blackjack! You win")

        elif dealer_has_blackjack:
            print("Dealer has blackjack! Dealer wins")

    def dealer_hit(self):
        if self.dealer_hand.get_value < 15 :
            self.dealer_hand.add_card.append(self.deck.deal.pop(0))
            dealer_hand.get_value.display
```

- This is the Class game where the game will run
- Def play is to play, first we call the deck, then we shuffel it then we print the hand (player and CPU) Then make a function that check if we have blackjack or not.
- We add the option to play by pressing h/s or hit/stay and afther that the function will check the value if it's > 21 the you lose
- Then we add a loop to for playing again or not

```
def player_is_over(self):
    return self.player_hand.get_value() > 21

def check_for_blackjack(self):
    player = False
    dealer = False
    if self.player_hand.get_value() == 21:
        player = True
    if self.dealer_hand.get_value() == 21:
        dealer = True

    return player, dealer

def show_blackjack_results(self, player_has_blackjack, dealer_has_blackjack):
    if player_has_blackjack and dealer_has_blackjack:
        print("Both players have blackjack! Draw")

    elif player_has_blackjack:
        print("You have blackjack! You win")

    elif dealer_has_blackjack:
        print("Dealer has blackjack! Dealer wins")

def dealer_hit(self):
    if self.dealer_hand.get_value < 15 :
        self.dealer_hand.add_card.append(self.deck.deal.pop(0))
        dealer_hand.get_value.display
```

- Def player_is_over for ending the game and give the player a game over
- Def check_for_blackjack is for checking if the player/dealer have 21, if player have 21 the player win. If the dealer have 21 than he win. But if both have 21 player and dealer will have a draw
- Def show_blackjack_result this is to print the out come of who will win
- Def dealer_hit will give the dealer a change to hit a card if the value is < 15

```
164
165     def dealer_hit(self):
166         if self.dealer_hand.get_value < 15 :
167             self.dealer_hand.add_card.append(self.deck.deal.pop(0))
168             dealer_hand.get_value.display
169
170
171
172 g = Game()
173 g.play()
174
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
Your hand: 15
PS C:\Users\User> & C:/Users/User/Desktop/BlackJack_env/scripts/python.exe "e:/B
Your hand is:
Q of Spades
K of Clover
Value: 20
hidden
J of Diamonds
Please choose [Hit / stay] h
Q of Spades
K of Clover
2 of Clover
Value: 22
You have lost!
Play Again? [Y/N]
```

Source of help :

https://www.askpython.com/python/examples/blackjack-game-using-python

http://faculty.salina.k-state.edu/tim/software/blackjack/docblackjack.html

https://sites.google.com/site/alexcs110/home/final-project

https://stackoverflow.com/questions/2982133/python-blackjack

https://www.youtube.com/watch?v=yJz2at4Hco4&ab_channel=techBytesio

https://www.youtube.com/watch?v=IPMcV_IXtX4&ab_channel=EngineerMan

https://www.youtube.com/watch?v=s99L7ctOGkk&t=939s&ab_channel=TigerhawkT3