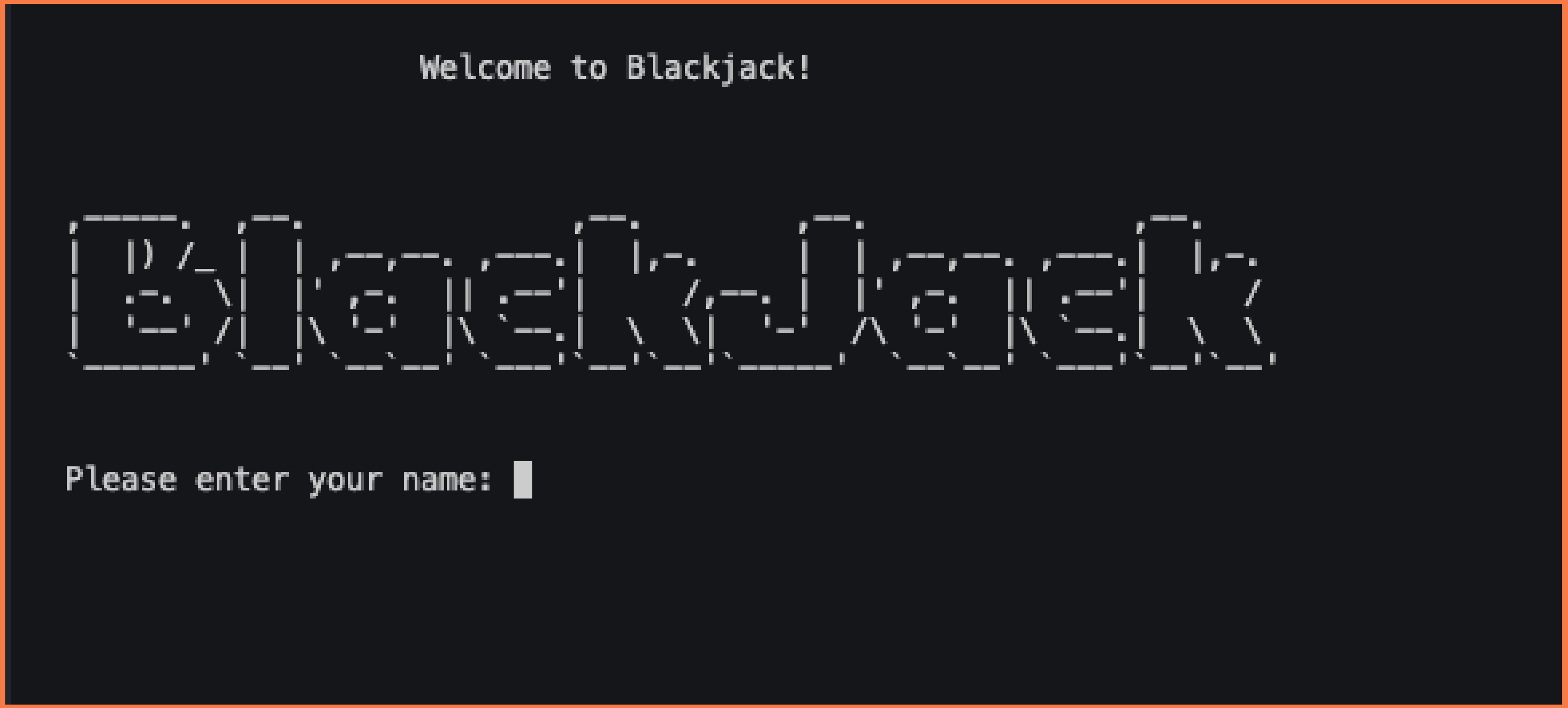
BLACKJACK IN CLI

Divij Seth



```
Welcome to Blackjack!

Please enter your name:
```

Core Features

Allow a graphical user interface for each round of game play that enhanced attention and readability.

Betting functionality, which allows players to maximise their winnings, and grow their balance.

Ability to deposit and withdraw your balance, because what good are winnings, if you cannot take them home.

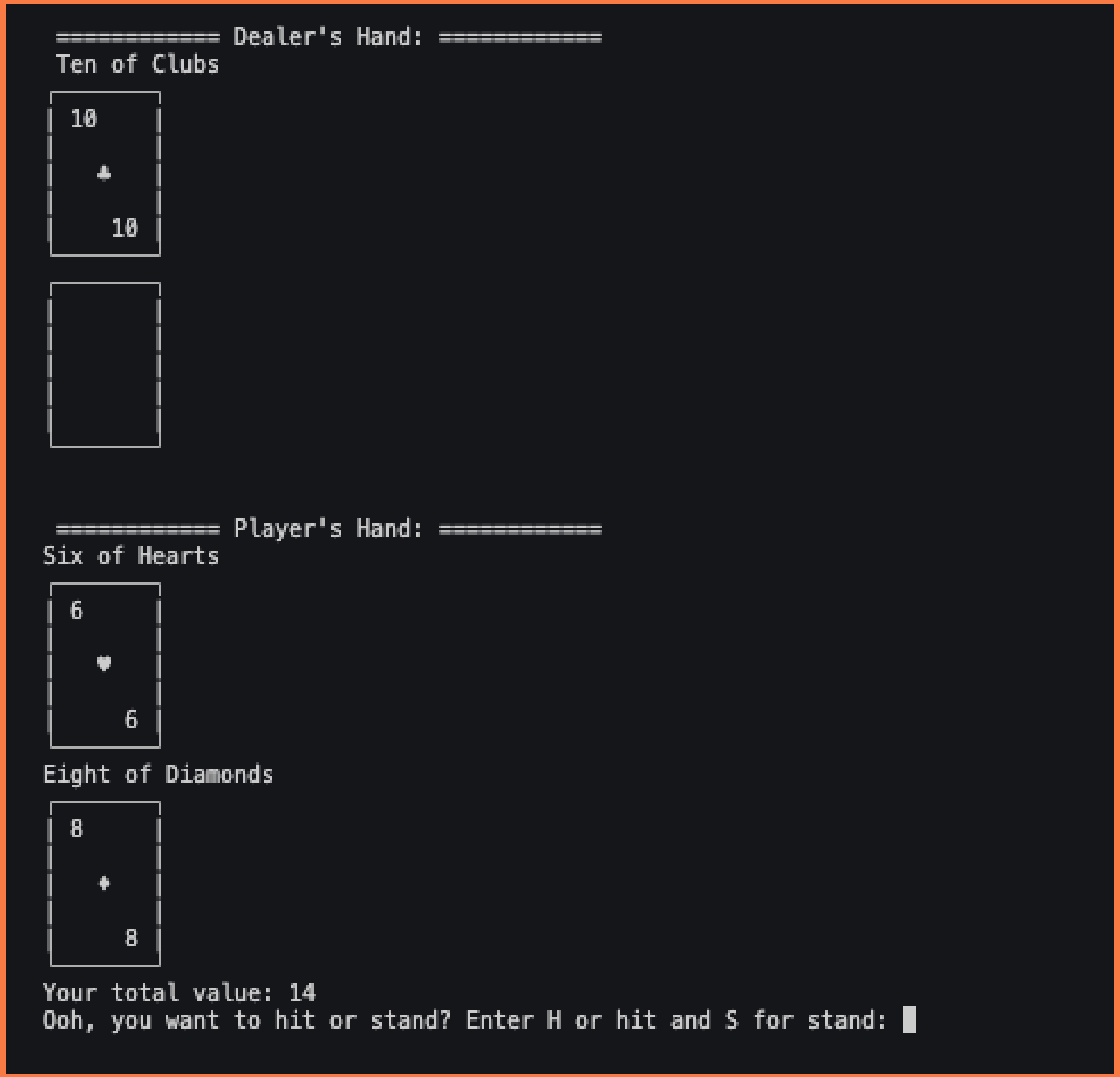
State of the art dealer logic, to keep the players on their toes!

Control the way you play, and make better in game decisions with a hit or stand feature.

The GUI

Visual Element that is sure to keep you entertained, and engaged.

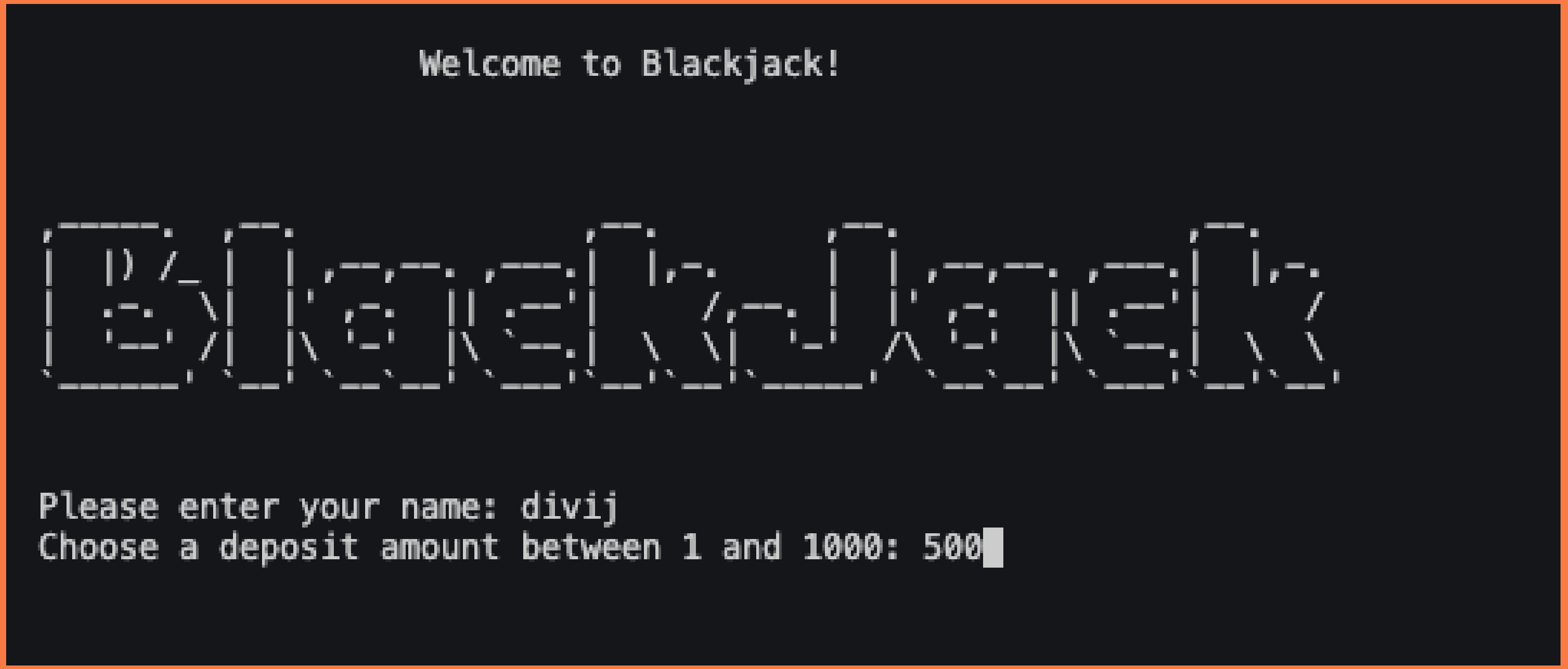
- Gui display of cards that corresponds to card and its values from the deck, that gets dealt to the player and the dealer.
- Also shows the value of each hand, and a readable description of the card to cater to all audiences, novice to pros.



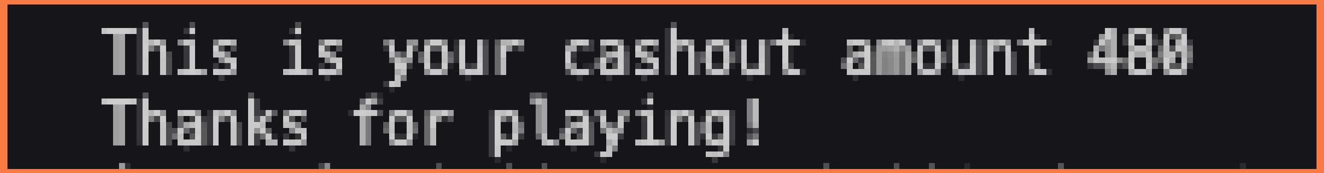
```
 ============ Dealer's Hand: ============
 Ten of Clubs
┌───────┐
| 10    |
|       |
|   ♣   |
|       |
|    10 |
└───────┘
┌───────┐
|       |
|       |
|       |
|       |
|       |
└───────┘


 ============ Player's Hand: ============
Six of Hearts
┌───────┐
| 6     |
|       |
|   ♥   |
|       |
|     6 |
└───────┘
Eight of Diamonds
┌───────┐
| 8     |
|       |
|   ♦   |
|       |
|     8 |
└───────┘
Your total value: 14
Ooh, you want to hit or stand? Enter H or hit and S for stand: █
```

Deposit and Withdrawals

Got cash on hand, want to make more? Dive in and secure your winnings.



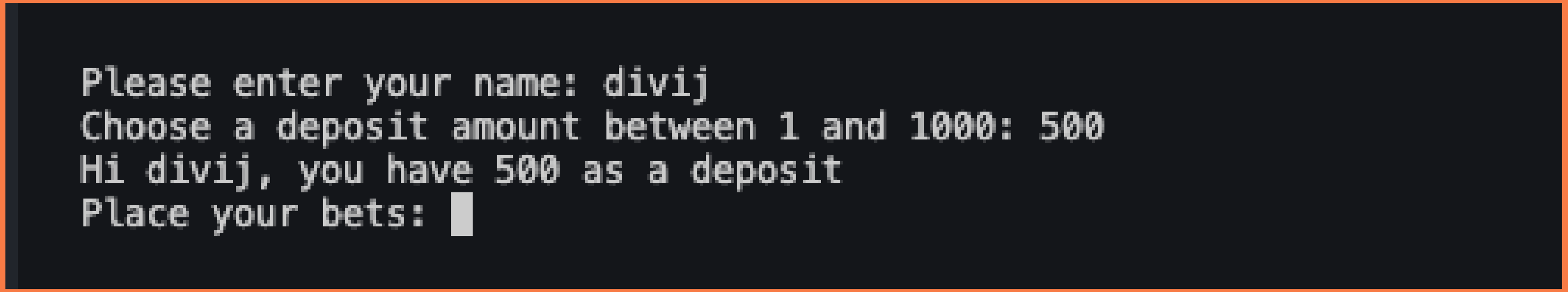
```
                  Welcome to Blackjack!

Please enter your name: divij
Choose a deposit amount between 1 and 1000: 500
```



```
This is your cashout amount 480
Thanks for playing!
```

With this feature, you can deposit upon opening, the game, and this balance will remain with you to grow and cashout if you wish, once you have decided not to play anymore.

Betting Functionality

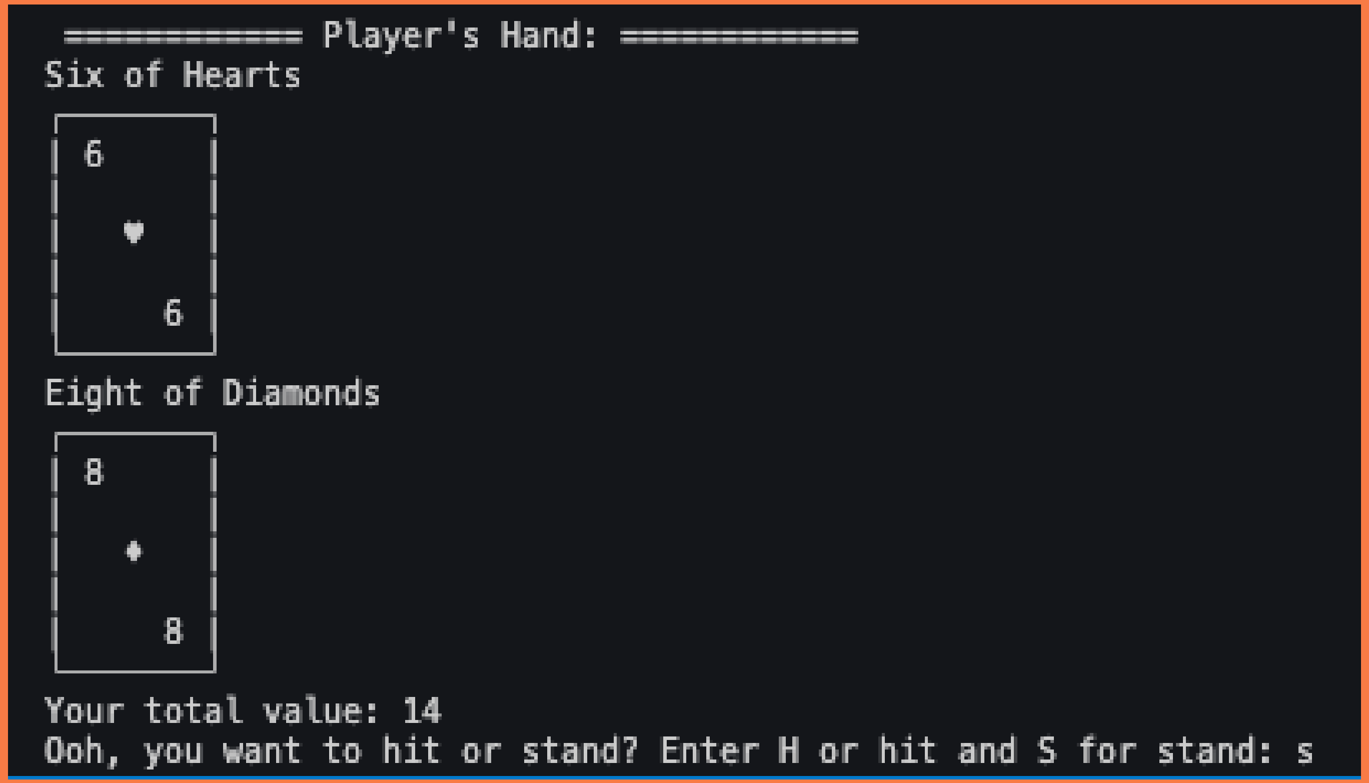


```
Please enter your name: divij
Choose a deposit amount between 1 and 1000: 500
Hi divij, you have 500 as a deposit
Place your bets: 
```

With this feature, you may bet your money that you have chosen to deposit, each bet must be less than the deposited amount, upon winning, your bet value is doubled, as per BlackJack rules and added to your balance. However this works in reverse too, when you lose, the bet amount is subtracted from your balance. Upon a draw, balance values stay the same, so you can try again to increase your money.

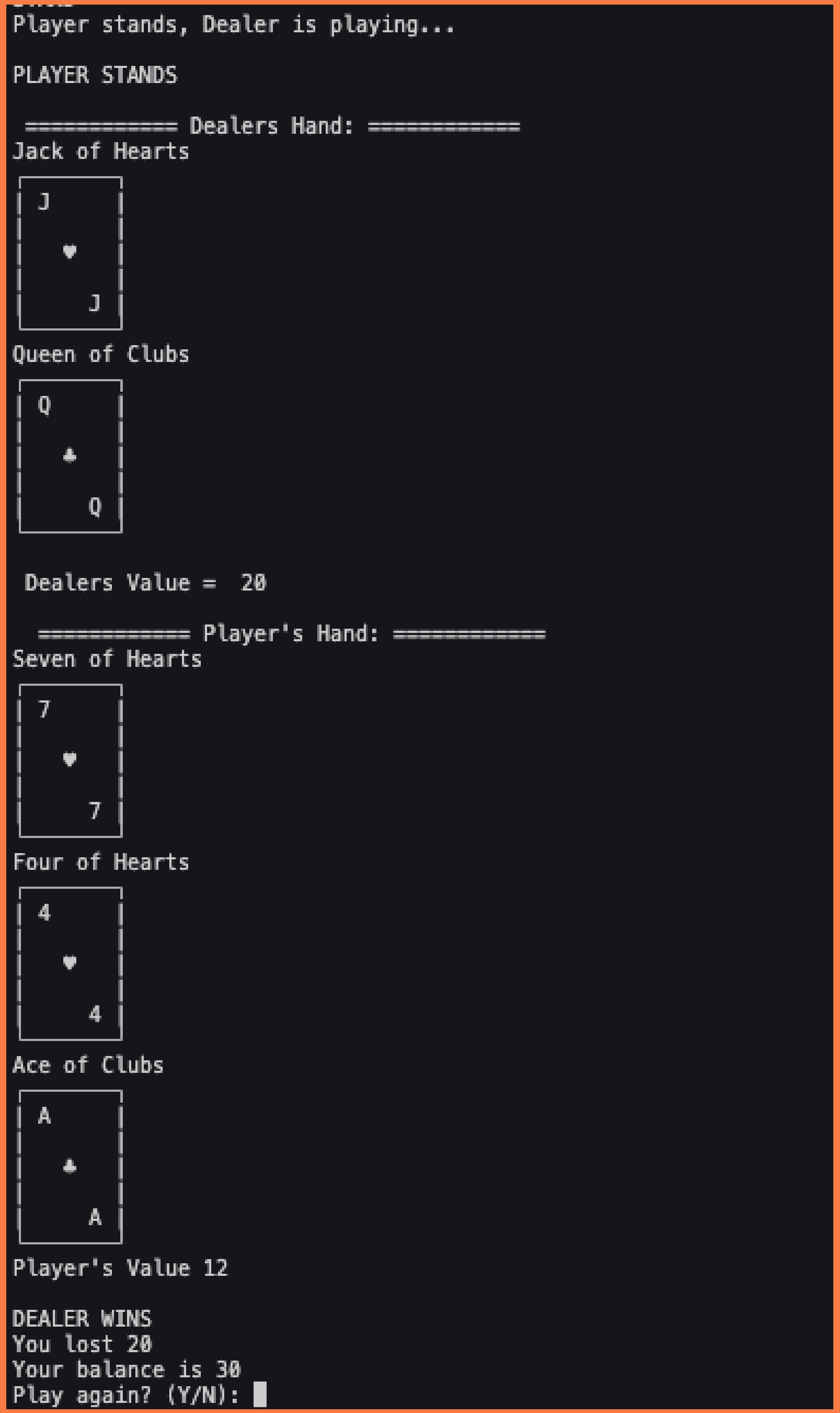
Control the way you play

Hit or Stand Feature



```
 ============ Player's Hand: ============
Six of Hearts
 ┌───────┐
 | 6     |
 |       |
 |   ♥   |
 |       |
 |     6 |
 └───────┘
Eight of Diamonds
 ┌───────┐
 | 8     |
 |       |
 |   ♦   |
 |       |
 |     8 |
 └───────┘
Your total value: 14
Ooh, you want to hit or stand? Enter H or hit and S for stand: s
```

Maximise your chances of winning against our dealers.



```
Player stands, Dealer is playing...

PLAYER STANDS

 ============ Dealers Hand: ============
Jack of Hearts
 ┌───────┐
 | J     |
 |       |
 |   ♥   |
 |       |
 |     J |
 └───────┘
Queen of Clubs
 ┌───────┐
 | Q     |
 |       |
 |   ♣   |
 |       |
 |     Q |
 └───────┘

 Dealers Value =  20

  ============ Player's Hand: ============
Seven of Hearts
 ┌───────┐
 | 7     |
 |       |
 |   ♥   |
 |       |
 |     7 |
 └───────┘
Four of Hearts
 ┌───────┐
 | 4     |
 |       |
 |   ♥   |
 |       |
 |     4 |
 └───────┘
Ace of Clubs
 ┌───────┐
 | A     |
 |       |
 |   ♣   |
 |       |
 |     A |
 └───────┘
Player's Value 12

DEALER WINS
You lost 20
Your balance is 30
Play again? (Y/N):
```

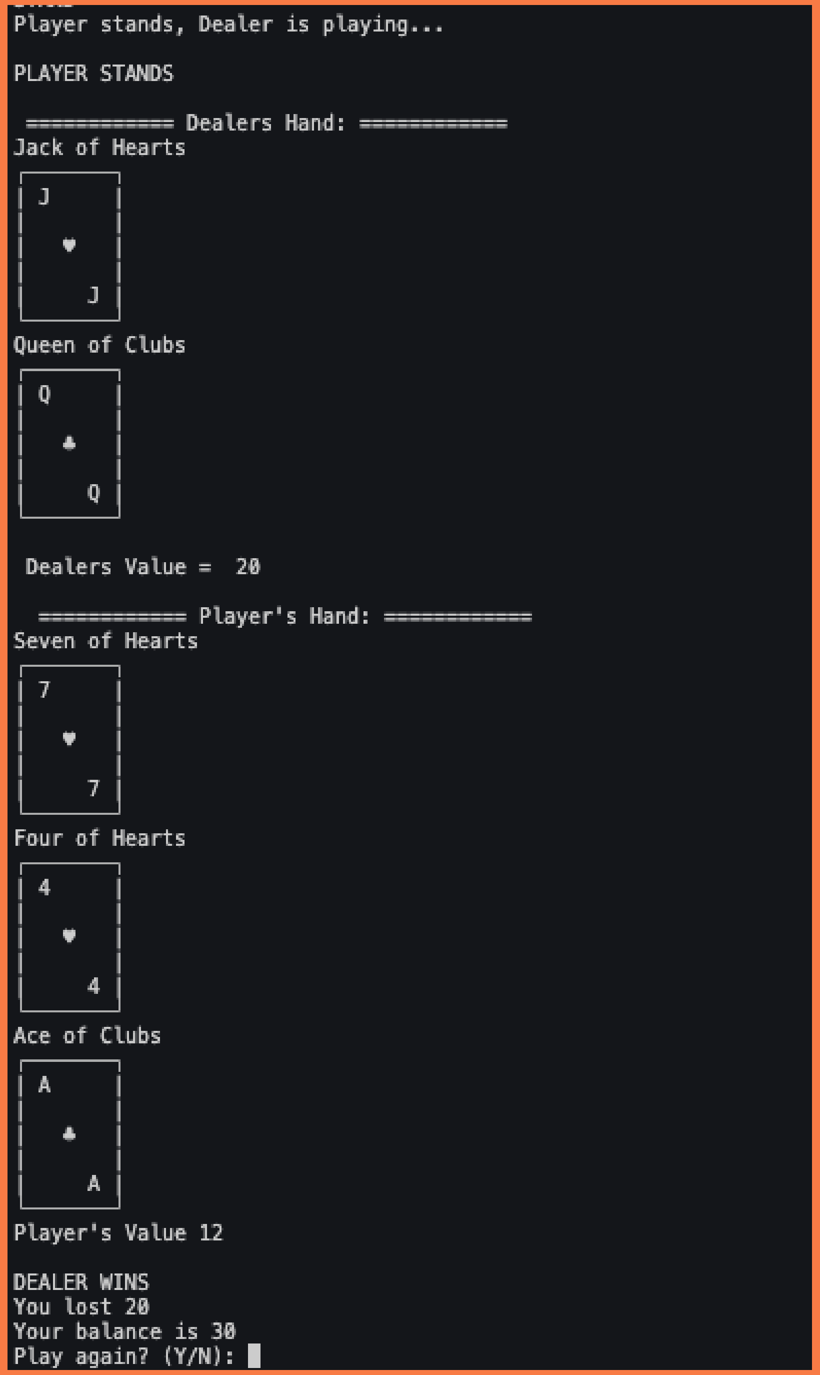
- The player has a choice of hitting or standing after a round is dealt, they can use this to make smarter decisions and move closer to the value of 21.
- Once hit, another card will be drawn at from the shuffled deck and added to the player's hand.
- If the player wishes to stand, the dealer will reveal his cards and keep hitting until they beat the player, bust, or draw.

Competitve Dealers

Get your casino fix with dealers logic that mimics a real life dealer.

In this feature, the player is always kept on their toes, with the dealer logic implementation. The dealer will use the value of your hand to try and beat you once you have stood. Dealer will keep hitting until the value of their hand is higher than the player, or equal. Dealer is less risk averse.

However, if the dealer already has a hand of 18 or higher, they will have to stand, increasing your chances of winning, the exact same way as in a casino.



```
Player stands, Dealer is playing...

PLAYER STANDS

 ============ Dealers Hand: ============
Jack of Hearts
 ┌───────┐
 │ J     │
 │       │
 │   ♥   │
 │       │
 │     J │
 └───────┘
Queen of Clubs
 ┌───────┐
 │ Q     │
 │       │
 │   ♣   │
 │       │
 │     Q │
 └───────┘

 Dealers Value =  20

  ============ Player's Hand: ============
Seven of Hearts
 ┌───────┐
 │ 7     │
 │       │
 │   ♥   │
 │       │
 │     7 │
 └───────┘
Four of Hearts
 ┌───────┐
 │ 4     │
 │       │
 │   ♥   │
 │       │
 │     4 │
 └───────┘
Ace of Clubs
 ┌───────┐
 │ A     │
 │       │
 │   ♣   │
 │       │
 │     A │
 └───────┘
Player's Value 12

DEALER WINS
You lost 20
Your balance is 30
Play again? (Y/N): █
```

Additional Features

Check Win Logic

When cards are dealt, there is a check to see if either player or dealer has blackjack. Then finally after a round is played, it can end with a bust, win, lose or a push. Implemented to mimic a real game of blackjack.

Play Again

For players that are feeling hot! Choose to play again to increase your balance, or chase your losses(Don't do that). This feature puts the control back to the player. Be warned, there is a check to see if you have enough balance left.

For developers, by developers

Lets dive into the developer side of things...

Check for Win logic

Dealer's Logic

OOP Principles

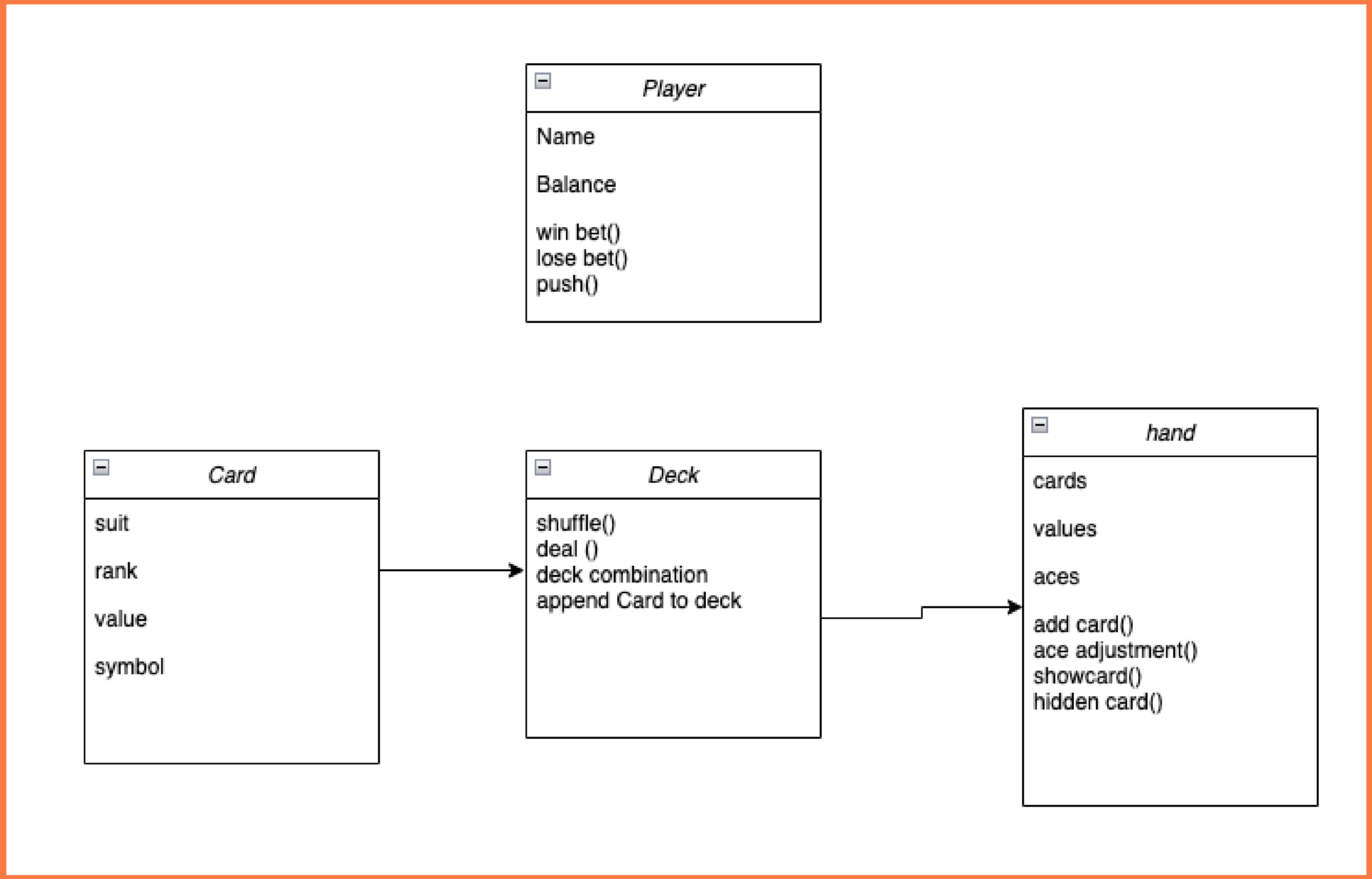
PEP 8 Styling

DRY

Wide range of concepts

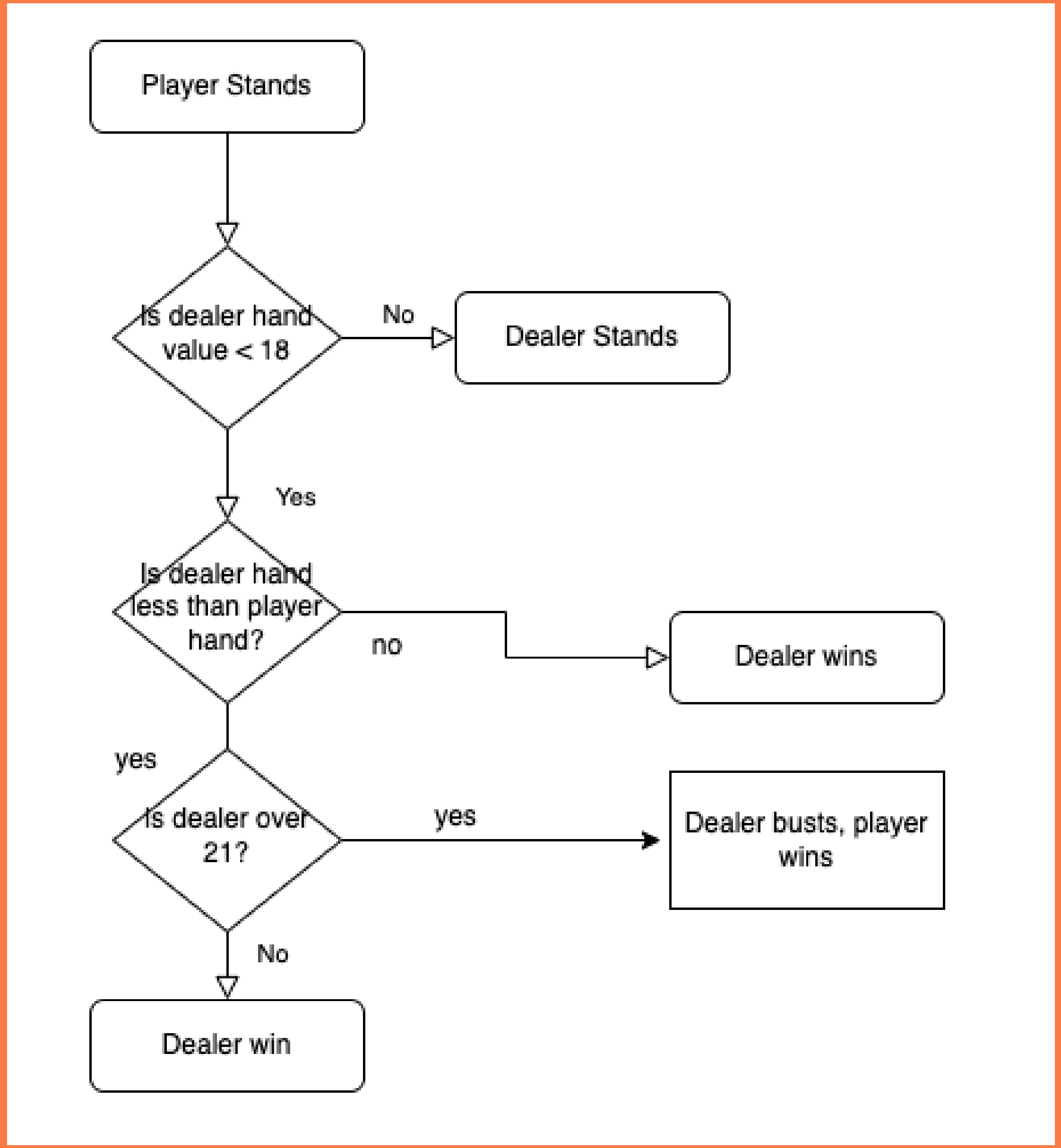
OOP Principles

Classes Used



Dealer Logic

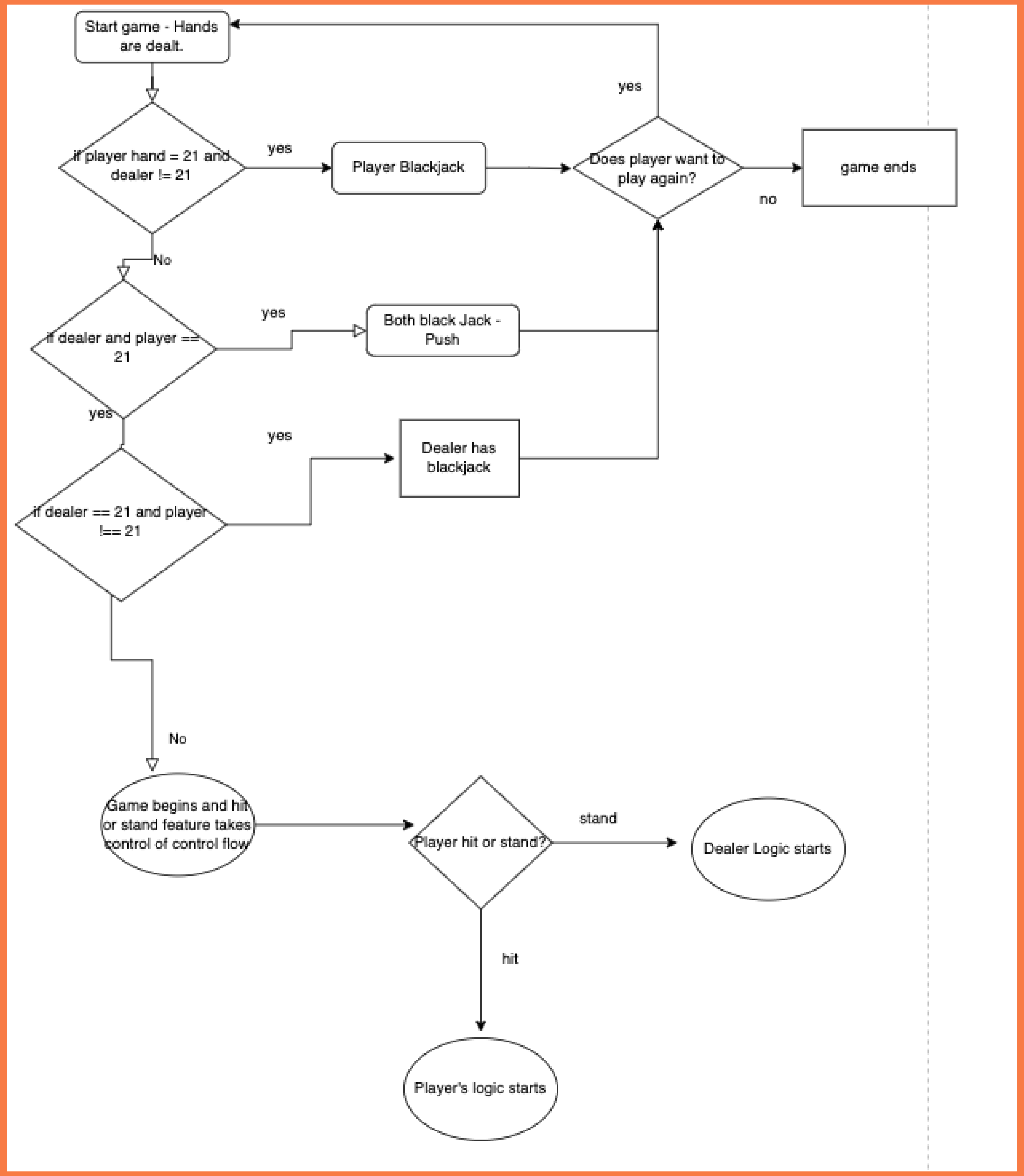
How the dealer plays


Player Stands
Is dealer hand value < 18
No
Dealer Stands
Yes
Is dealer hand less than player hand?
no
Dealer wins
yes
Is dealer over 21?
yes
Dealer busts, player wins
No
Dealer win

This logic is implemented once the player has decided to stand, the dealer will check to see if the value of their hand is less than 18 before continuing. Then it will check if the value is less than the players hand, if so, the dealer is obligated to keep hitting, until it either beats the player or busts.

Check for win and outcomes

With this logic, all outcomes are captured.


Start game - Hands are dealt.
if player hand = 21 and dealer != 21
yes
Player Blackjack
Does player want to play again?
yes
no
game ends
No
if dealer and player == 21
yes
Both black Jack - Push
yes
yes
Dealer has blackjack
if dealer == 21 and player !== 21
No
Game begins and hit or stand feature takes control of control flow
Player hit or stand?
stand
Dealer Logic starts
hit
Player's logic starts

- When the game starts, there is a check to see if either players or both have blackjack before progressing to hit or stand feature.
- If there is no blackjack and player wishes to hit, they can control their flow to reach a desired outcome pre-defined with win, loss or tie checking against dealer values. If they choose to stand, the dealer logic outlined previously takes over. In all scenarios there is an outcome and the player will always be prompted to play again if they wish, depending on if their balance has not reached zero.

Challenges, favourite parts and ethics.

- Scoping the project and its required time to implement was a major challenge, I underestimated the time it would take to implement certain factors, and didn't take into account debugging time. A major lesson learned for next time.

- Favourite part was designing logic in a way that captured all the elements of a real life game of blackjack, while capturing all the concepts we had learned.

- Ethical challenges arose in regards to a high amount of implementations of the same game found online, since this game covers all aspects of the concepts learned during python, it is a common project. There are various types of implementations which leave little room to solve this challenge uniquely. However since this project covered all aspects of the concepts we learned, it is one of the main reasons why I chose to make this game.