ROULETTE TERMINAL APP (T1A3)

ROULETTE TERMINAL APPLICATION

- Wanted to create a basic roulette type game.
- The game is similar to roulette, where you can bet on ‘even’, ‘odd’, ‘red’, ‘black’, or any number from 0 to 36 inclusive.
- The payout rules are 18 to 1 if the number lands on the number that you bet on. All other bets pay out 1 to 1.

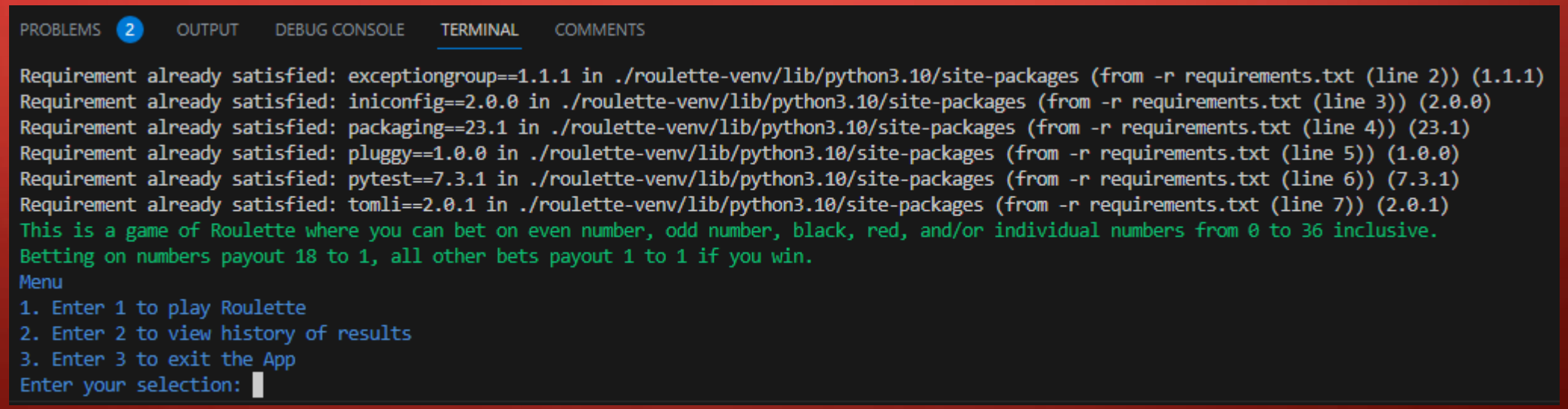
FEATURES

There are a total of nine features in this application. These are:-

- The navigation menu.
- The enter funds feature.
- The choose what to bet on feature.
- The bet amount feature.
- Generate random number feature.
- Display win amount feature.
- Play feature.
- Add result history feature.
- View result history feature.

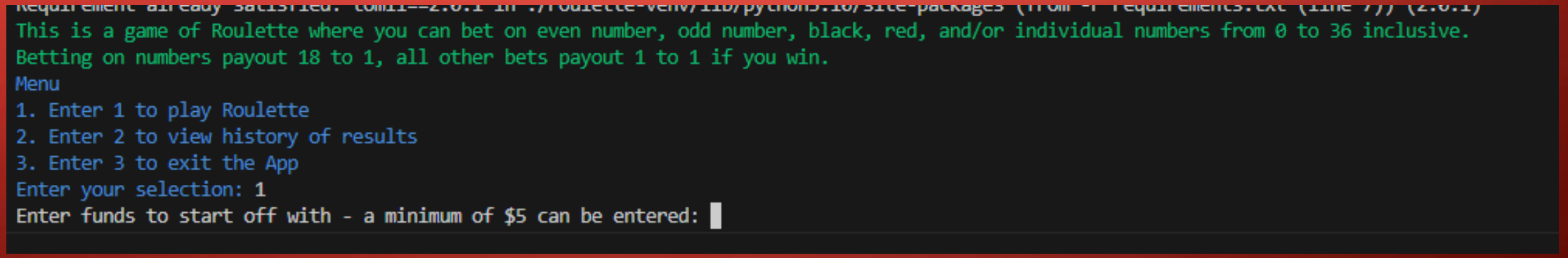
NAVIGATION MENU FEATURE

- You are given three options to start with, type '1' to play roulette, '2' to view the history of results, and '3' to exit the application.



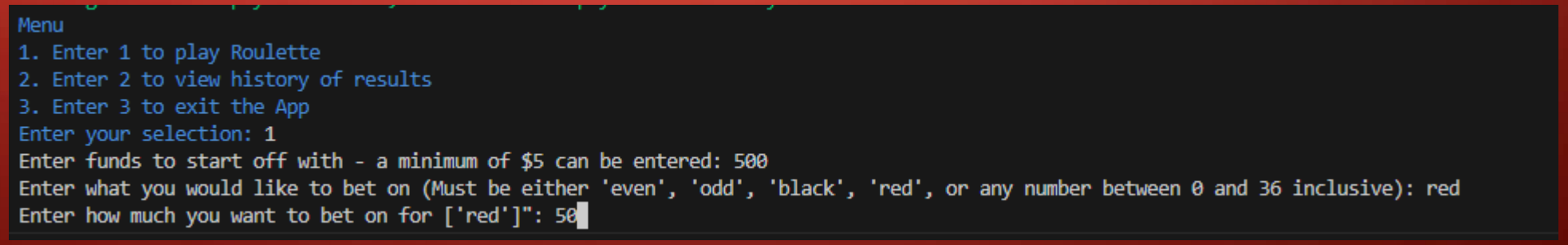
ENTER FUNDS FEATURE

- This feature allows the user to enter funds at the start of the game.
- A minimum of $5 can be entered, as bets are a minimum of $5.



CHOOSE WHAT TO BET ON AND BET AMOUNT FEATURE

- The choose what to bet on feature lets the user choose what to bet on and appends it to a list data structure.
- The bet amount feature lets the user enter the amount they wish to bet with, on what they have chosen.



GENERATE RANDOM NUMBER FEATURE

- This feature generates a random number and displays the number landed on and if it is red or black and even or odd.

```
Enter funds to start off with - a minimum of $5 can be entered: 500
Enter what you would like to bet on (Must be either 'even', 'odd', 'black', 'red', or any number between 0 and 36 inclusive): red
Enter how much you want to bet on for ['red']": 50
remaining funds $450
Enter 'yes' to start game or 'no' to continue placing bets: no
Enter what you would like to bet on (Must be either 'even', 'odd', 'black', 'red', or any number between 0 and 36 inclusive): even
Enter how much you want to bet on for ['red', 'even']": 50
remaining funds $400
Enter 'yes' to start game or 'no' to continue placing bets: yes
The number landed on is 5 red odd
You won $50 as the number was red
Your remaining funds are $500
Do you wish to exit game? (yes/no):
```

DISPLAY WIN AMOUNT FEATURE

- Displays the amount won if you won in the game, and the remaining funds.

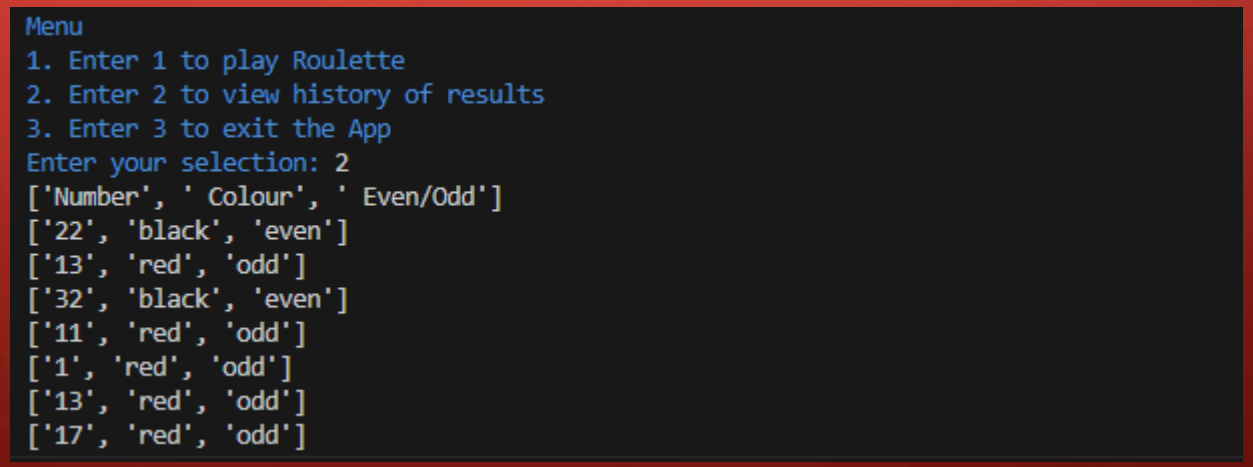
```
Enter funds to start off with - a minimum of $5 can be entered: 500
Enter what you would like to bet on (Must be either 'even', 'odd', 'black', 'red', or any number between 0 and 36 inclusive): red
Enter how much you want to bet on for ['red']": 50
remaining funds $450
Enter 'yes' to start game or 'no' to continue placing bets: no
Enter what you would like to bet on (Must be either 'even', 'odd', 'black', 'red', or any number between 0 and 36 inclusive): even
Enter how much you want to bet on for ['red', 'even']": 50
remaining funds $400
Enter 'yes' to start game or 'no' to continue placing bets: yes
The number landed on is 5 red odd
You won $50 as the number was red
Your remaining funds are $500
Do you wish to exit game? (yes/no):
```

PLAY FEATURE

- The play feature essentially controls the flow of the game.
- There are multiple conditional statements like ‘do you want to continue betting or start game?’, and ‘do you wish to exit game?’, etc.

ADD AND VIEW HISTORY FEATURE

- After generating the random number, the application adds the results to a ‘history.csv’ file.
- You can view the result history in the main navigation menu by entering ‘2’.



CODE LOGIC - MAIN.PY FILE

- This file contains the code necessary to run the application.
- The most important part of this file is the navigation menu code. I declared two local variables in the 'if conditional' – 'total_funds' and 'what_you_bet_on'.
- I reassigned the 'total_funds' variable as I wanted to display the remaining funds after exiting the game.

```
def nav_menu():
    # Navigation menu function
    print(f"{fg('blue')}Menu")
    print("1. Enter 1 to play Roulette")
    print("2. Enter 2 to view history of results")
    print("3. Enter 3 to exit the App")
    user_choice = input(f"Enter your selection: {attr('reset')}")
    return user_choice

while user_selection != "3":
    user_selection = nav_menu()

    if user_selection == "1":
        # This condition allows you to play roulette, the total_funds variable
        # is reset to 0 everytime the user starts the game.
        # total_funds variable is overridden as the user plays the game.
        total_funds = 0
        what_you_bet_on = []
        total_funds = funds()
        total_funds = play(what_you_bet_on, file_name, play_roulette, total_fund
        print(f"{fg('yellow')}You exited the game with ${total_funds}{attr('rese
    elif user_selection == "2":
        view_history(file_name)
    elif user_selection == "3":
        continue
    else:
        print("Invalid Input")

print("Thanks for playing Roulette")
```

FUNDS() FUNCTION

- Used a while loop to ask the user to enter funds of minimum $5. If the user enters less than 5, it asks the user again until they enter a valid input.

```
def funds(user_input=None):
    # Funds function - lets user put in initial funds to play with.
    # It lets the user put in funds greater than 5, as the minimum bet is 5.
    while True:
        if user_input is not None:
            user_input_value = user_input
        else:
            user_input_value = input("Enter funds to start off with - a "
                                     "minimum of $5 can be entered: ")

        try:
            total_funds = int(user_input_value)
            if total_funds < 5:
                total_funds = 0
                print("Please enter amount greater than 5")
            else:
                return total_funds
        except ValueError:
            print("Please type in numbers only")
        except Exception as e:
            print(e)
```

BET_SELECTION() FUNCTION

- Uses a while loop set to True.
- Uses input() method to ask user to enter what to bet on, and appends to a list if the bet is valid.
- Catches any errors like 'ValueError'

```
def bet_selection(what_you_bet_on):
    # This function lets the user choose what to bet on and appends
    # the bet to a list after each bet.
    while True:
        try:
            choose_bet = input("Enter what you would like to bet on "
                               "(Must be either 'even', 'odd', 'black', 'red',"
                               " or any number between 0 and 36 inclusive): ")
            if choose_bet == 'even':
                return what_you_bet_on.append(choose_bet)
            elif choose_bet == 'odd':
                return what_you_bet_on.append(choose_bet)
            elif choose_bet == 'black':
                return what_you_bet_on.append(choose_bet)
            elif choose_bet == 'red':
                return what_you_bet_on.append(choose_bet)
            for i in range(0, 37):
                if int(choose_bet) == i:
                    return what_you_bet_on.append(int(choose_bet))
            else:
                print("Invalid choice. You can only bet on 'even', 'odd', "
                      "'black', 'red', or any number between 0 and 36 inclusive"
        except ValueError:
            print("Invalid bet. Please type in 'even', 'odd', 'black', "
                  "'red', or any number between 0 and 36 inclusive")
        except Exception as e:
            print(e)
```

BETTING() FUNCTION

- Uses a while loop set to True.
- Asks the user to enter a bet less than total funds and greater than 5.
- Catches ValueError.

```
def betting(what_you_bet_on, total_funds):
    # This function allows the user to place bets greater than $5.
    while True:
        try:
            bet = int(input(f"Enter how much you want to bet on for "
                            f"{what_you_bet_on}\": "))
            if bet > total_funds:
                bet = 0
                print(f"Your bet has exceeded the total funds, please "
                      "enter a bet lower than {total_funds}")
            elif bet < 5:
                bet = 0
                print("Minimum bet is $5, please enter a bet $5 or higher")
            else:
                return bet
        except ValueError:
            print("Invalid bet. Please type in numbers only")
        except Exception as e:
            print(e)
```

PLAY() FUNCTION

- This function incorporates most of the other functions. It uses data returned from other functions.
- Example is the ‘total_funds’ variable is subtracted by the ‘next_bet’ variable, which was assigned to the ‘betting()’ function.

```
def play(what_you_bet_on, file_name, play_roulette, total_funds):
    # Play function - incorporates all the other functions and has
    # conditions to control the flow of the game.
    while play_roulette != "yes":
        if total_funds >= 5:
            bet_selection(what_you_bet_on)
            next_bet = betting(what_you_bet_on, total_funds)
            total_funds -= next_bet
            print(f"remaining funds ${total_funds}")
```

PLAY() FUNCTION CONTINUED

- After entering in your first bets, a prompt will come up, asking if you want to start game or continue betting.
- Used a while loop to catch any other input other that ‘yes’ or ‘no’.
- If ‘yes’ the game starts and the results are displayed and funds are calculated.
- Another prompt is ask after the game, to see if the user wants to exit or continue playing. If they want to continue, the ‘what_you_bet_on’ variable is reset to empty.

```
finished_betting = input("Enter 'yes' to start game or 'no' to "
                         "continue placing bets: ")
while finished_betting != "yes" and finished_betting != "no":
    finished_betting = input("Invalid input. Enter 'yes' to start "
                             "game or 'no' to continue placing bets

if finished_betting == "yes":
    random_number, color, even_odd = display_result(data_set, file_
    total_funds = win_lose(what_you_bet_on, random_number, \
                           next_bet, total_funds, color, even_odd)
    play_roulette = input("Do you wish to exit game? (yes/no): ")

    while play_roulette != "yes" and play_roulette != "no":
        play_roulette = input("Invalid input. Please enter 'yes' or
    if play_roulette == "no":
        what_you_bet_on = []
    elif play_roulette == "yes":
        break
```

DISPLAY_RESULT() FUNCTION

- Used the random module to generate random number.
- Used a for loop to iterate through the ‘data_set’ variable, and matched it to the respective color and even/odd.
- The ‘data_set’ variable contains a list of dictionaries with all the available numbers and data to bet on.

```
def display_result(data_set, file_name):
    # This function generates the random result and adds it
    # to the file_name variable.
    random_number = random.randint(0, 36)
    for data in data_set:
        if data["number"] == random_number:
            color = data["color"]
            even_odd = data["even_odd"]
            if color == "red":
                print(f"The number landed on is {random_number} "
                      f"{fg('red')}{color}{attr('reset')} {even_odd}")
            else:
                print(f"The number landed on is {random_number} "
                      f"{bg('white')}{fg('black')}{color}{attr('reset')} {even_o
            add_history(file_name, color, even_odd, random_number)

            return random_number, color, even_odd
```

WIN_LOSE() FUNCTION

- Uses a for loop to iterate through the 'what_you_bet_on' variable (list).
- It matches it with the generated random number from the 'display_result()' function.
- Then calculates amount won and displays the remaining funds.

```
def win_(parameter) what_you_bet_on: Any
        what_you_bet_on, random_number, next_bet, total_funds, color, even_odd):
    # This function displays if the user has won and how much
    # they have won.
    winnings = 0
    for element in what_you_bet_on:
        if element == random_number:
            winnings = (next_bet * 18) + next_bet
            amount_won = winnings - next_bet
            print(f"You won ${amount_won} as the number was {element}")
            total_funds += winnings
        elif element == even_odd or element == color:
            winnings = (next_bet * 2)
            total_funds += winnings
            amount_won = winnings - next_bet
            print(f"You won ${amount_won} as the number was {element}")
    print(f"Your remaining funds are ${total_funds}")

    return total_funds
```

CHALLENGES FROM BUILD PROCESS

- Difficult to keep track of all the variables and where they change in the functions after calling the functions.
- Trying not to write repeated code.

Favourite part:-

- Completing the ‘play()’ function, as that was what tied everything together, and was the most difficult to make and keep track of.

THANK YOU FOR LISTENING