MANSUKHBHAI KOTHARI NATIONAL SCHOOL

H&M ROYAL, SR.19, KONDHWA BK., OPP. TALAB FACTORY, PUNE-411048

PLOTTING GRAPHS OF SCORES OF TWO GAMES IN PYTHON

For the CLASS XII 2020-21

As a part of the Informatics Practices Course (065)

Submitted By-
Umair Shaikh
XII-A

Under the Guidance of
Ms. NILOFER SAYYED
Informatics Practices Teacher

CERTIFICATE

This is to certify that Umair Shaikh, Ubaid Karbhari, and Ruwaid Sambulkhani, the student of class – XII-A have completed their IP Project under the guidance of Mrs. Nilofer Sayyed.

ACKNOWLEDGEMENT:

In the accomplishment of this project successfully, many people have bestowed upon me their blessings and the heart pledged support, this I am utilizing to thank all the people who have been concerned with this project.

I would like to thank our principal Mrs. Varsha Gokhale and our IP teacher Mrs. Nilofer Sayyed, whose valuable guidance has been the ones that helped us patch this project and make it a full-proof success. Her suggestions and instructions have served as the major contributor to the completion of the project.

Then I would like to thank my parents and friends who have helped me with their valuable suggestions and guidance has been very helpful in various phases of the completion of the project. Last but not least I would like to thank our classmates who have helped me a lot.

Umair Shaikh

XII-A SCIENCE

INDEX:

AIM

The project aims to make a two games (guessing game and rock paper scissors) and it allows the user to compare their data to previous users who have played the game in the form of a bar graph/chart.

OBJECTIVE AND SCOPE OF THE PROJECT

The project aims to make a two games (guessing game and rock paper scissors) and it allows the user to compare their data to previous users who have played the game in the form of a bar graph/chart.

We have done this using various python modules namely, random.py, pandas.py, and matplolib.py.

SYSTEM REQUIREMENTS

Requirement	Minimum	Recommended
RAM	4 GB of free RAM	8 GB of total system RAM
Disk space	2.5 GB and another 1 GB for caches	SSD drive with at least 5 GB of free space
Operating system	Officially released 64-bit versions of the following: • Microsoft Windows 8 or later • macOS 10.13 or later • Any Linux distribution that supports Gnome, KDE, or Unity DE.	Latest 64-bit version of Windows, or Linux (for ex, Debian, Ubuntu, or RHEL)

SALIENT FEATURES

Two interactive python games hangman and guessing game. The user has to guess a number between 1 and 10 in few tries and in rock paper scissors the user has to beat the computer by choosing either rock, paper, or scissors!

It allows multiple users to play the game and at the end of each game, it prompts the current user to see a bar graph to compare their scores to previous users that have played the game.

Modules and functions used in the project:

1. The random.py module has been used to generate a random number.
2. The pandas.py module has been used to store user data in a data frame and write to and read that data from a CSV file.
3. The matplolib.py module has been used for data visualization by plotting a bar graph that compares user data(score) of all the users that have played the game

A WALKTHROUGH OF THE PROJECT(CODE)

Code of the guessing game:

→

```python
import random
import pandas as pd
import matplotlib.pyplot as plt


def play_gg():
    ranNum = random.randint(1, 10)
    tries = 0
    score = 50
    username = input("Enter your name: ")
    print(ranNum)

    def append_csv_and_read_from_it():

        players = {username: score}
        playerData = pd.DataFrame(players, index=players.keys())

        playerData.to_csv("scorelist.csv", header=None, mode="a")
        cols = ["Username", "Score"]

        playerDatatoPlot = pd.read_csv(
            "scorelist.csv", names=cols, header=None)

        playerDatatoPlot.set_index("Username")

        playerDatatoPlot.plot(kind="bar", x="Username", y="Score")


    print(
        f"Hello {username}, Welcome to the guessing game. "
        f"A random number has been chosen between 1 and 10 and you have"
        f" 5 tries to guess it."
    )
    while tries < 6:
        userInput = int(input("Enter a guess: "))
        if userInput > ranNum:
            print(
                f"Wrong! The right answer is less than {userInput}. Try again.")
            tries += 1
            score -= 10  # score decreases with the number of tries
        elif userInput < ranNum:
            print(
                f"Wrong! The right answer is more than {userInput} . Try again.")
            tries += 1
            score -= 10  # score decreases with the number of tries
        elif userInput == ranNum:
            print(f"You got it right. It took you {tries} tries to get it.")
            print(f'Your score is {score}')
            showPlot = input("Do you want to see the bar chart? (y/n) ")
            if showPlot.lower() == "y":
                append_csv_and_read_from_it()
                plt.show()
                break
            elif showPlot.lower() == "n":
                append_csv_and_read_from_it()
                print("Thanks for playing. Goodbye!")
                break
            else:
                print("Invalid Input!")
                break
        if tries == 5:
            append_csv_and_read_from_it()
            print(f"Your score is {score}")
            print(
                "You have run out of tries and you've lost the game. Thanks for playing! Goodbye..")
            break
```

Code of the rock paper scissors game:

→

```python
import random
import pandas as pd
import matplotlib.pyplot as plt


def play_rps():

    rps_tries = 0
    rps_score = 0
    rps_playagain = 0
    rps_username = input("Enter your name: ")

    def append_csv_and_read_from_it_rps():

        playersRPS = {rps_username: rps_score}

        playerDataRPS = pd.DataFrame(playersRPS, index=playersRPS.keys())

        playerDataRPS.to_csv("rps_scorelist.csv", header=None, mode="a")

        colsRPS = ["Username", "Score"]

        playerDatatoPlotRPS = pd.read_csv(
            "rps_scorelist.csv", names=colsRPS, header=None)

        playerDatatoPlotRPS.set_index("Username")

        playerDatatoPlotRPS.plot(kind="bar", x="Username", y="Score")

    while rps_tries < 5:

        print('Make your choice:')
        choice = str(input())
        choice = choice.lower()

        print("My choice is", choice)

        choices = ['rock', 'paper', 'scissors']

        computer_choice = random.choice(choices)
        print(computer_choice)

        print("Computer choice is", computer_choice)
        if choice in choices:
            if choice == computer_choice:
                print("it is a tie")
                rps_tries += 1
                print(rps_score)
            if choice == 'rock':
                if computer_choice == 'paper':
                    print("you lose, sorry :(")
                    rps_score -= 10
                    rps_tries += 1
                    print(rps_score)
                elif computer_choice == 'scissors':
                    print("You win!!!!! congrats :)")
                    rps_score += 10
                    rps_tries += 1
                    print(rps_score)
            if choice == 'paper':
                if computer_choice == 'scissors':
                    print("you lose, sorry :(")
                    rps_score -= 10
                    rps_tries += 1
                    print(rps_score)
                elif computer_choice == 'rock':
                    print("You win!!!!! congrats :)")
                    rps_score += 10
                    rps_tries += 1
                    print(rps_score)
            if choice == 'scissors':
                if computer_choice == 'rock':
                    print("you lose, sorry :(")
                    rps_score -= 10
                    rps_tries += 1
                    print(rps_score)
                elif computer_choice == 'paper':
                    print("You win!!!!! congrats :)")
                    rps_score += 10
                    rps_tries += 1
                    print(rps_score)

        else:
            print('invalid choice, try again')

        if rps_tries == 5:
            print("You have run out of tries")

            showPlotRPS = input("Do you want to see the bar chart? (y/n) ")
            if showPlotRPS.lower() == "y":
                append_csv_and_read_from_it_rps()
                plt.show()
                break
            elif showPlotRPS.lower() == "n":
                append_csv_and_read_from_it_rps()
                print("Thanks for playing. Goodbye!")
                break

            else:
                print("Invalid Input!")
                break
```
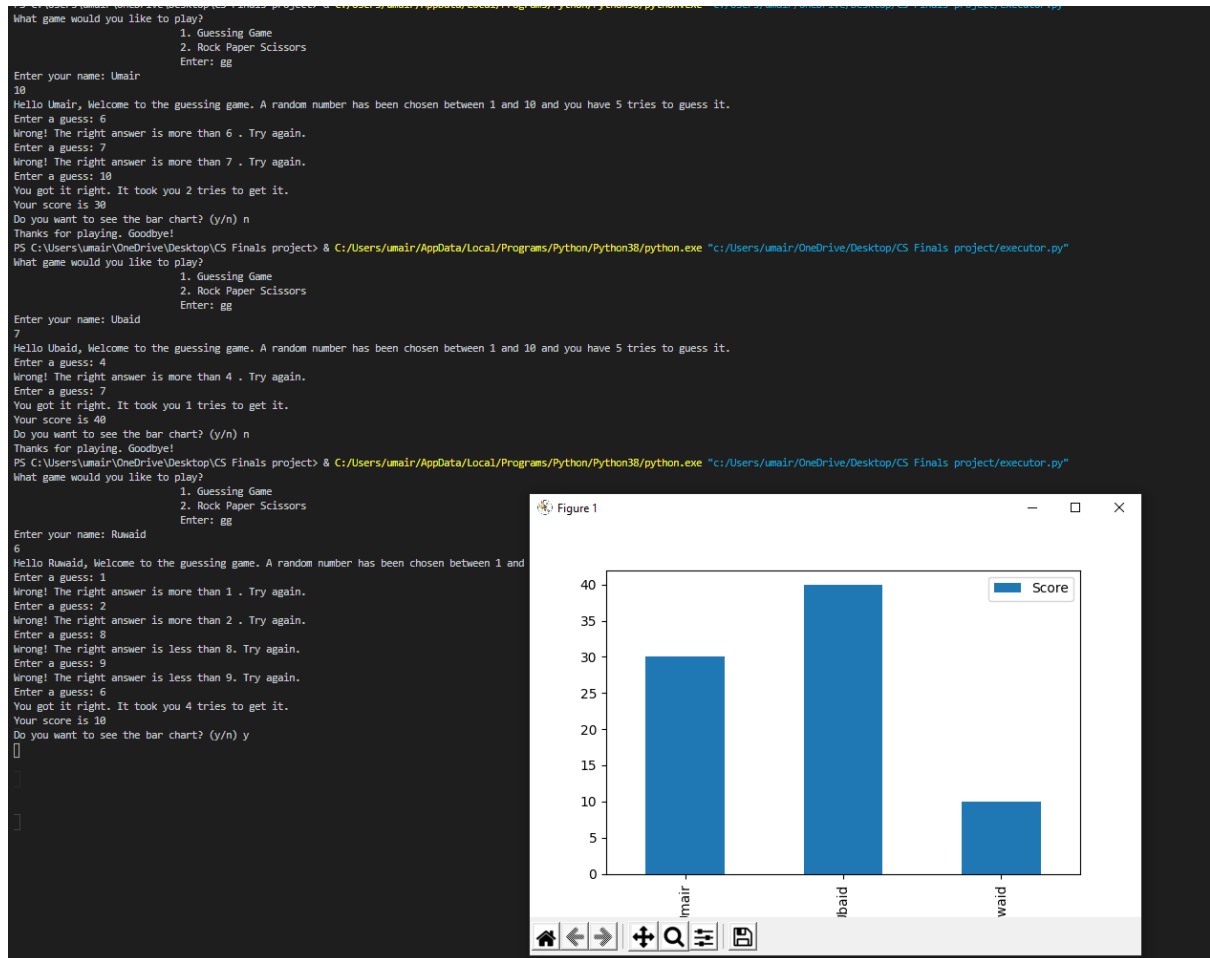
Code of the executor file:

```python
executor.py > ...

1    import guessing_game
2    import rps
3
4    chooseGame = input('''What game would you like to play?
5                                1. Guessing Game
6                                2. Rock Paper Scissors
7                                Enter: ''')
8
9    if chooseGame == "gg":
10       guessing_game.play_gg()
11   elif chooseGame == "rps":
12       rps.play_rps()
13
```
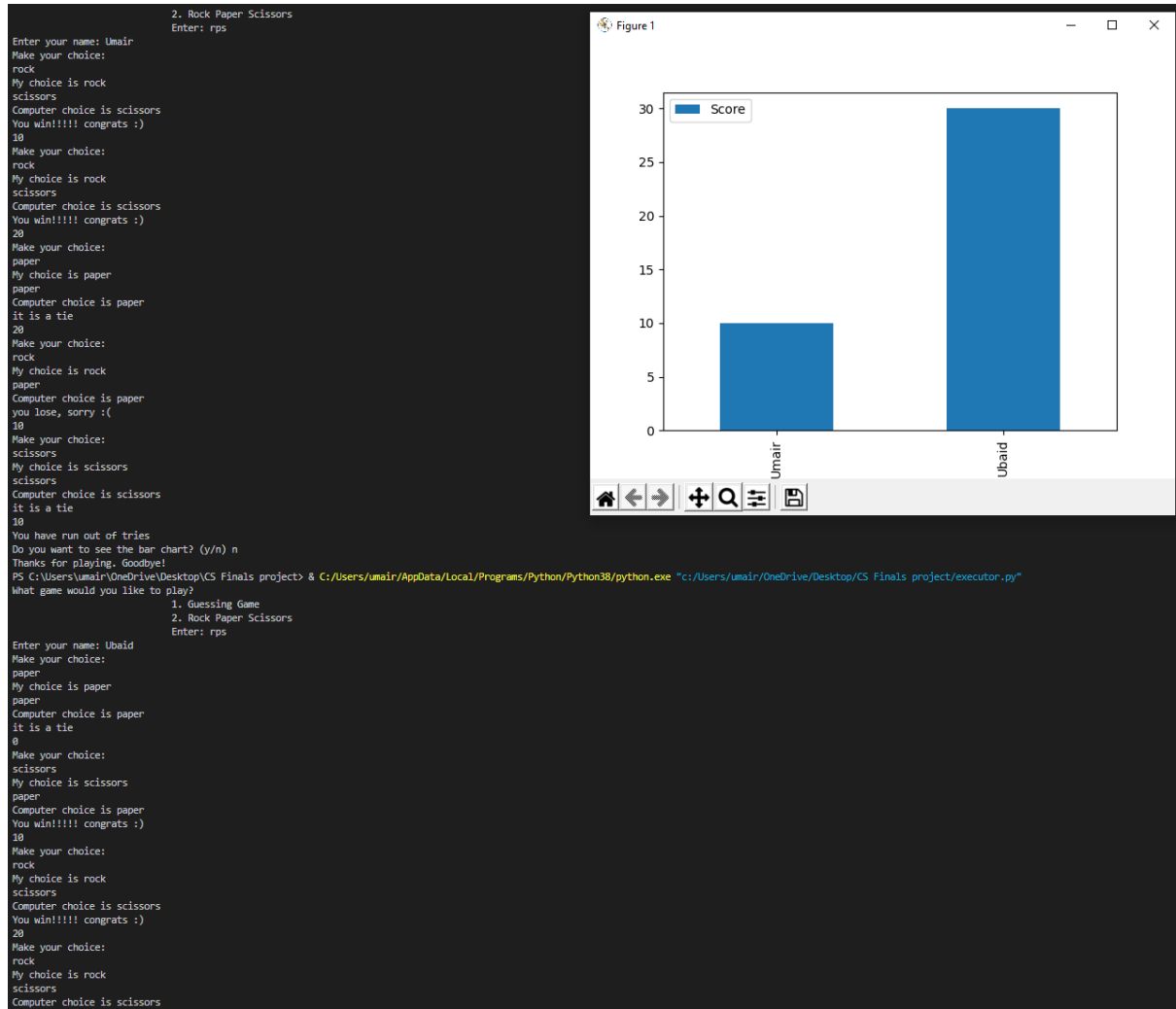
A WALKTHROUGH OF THE PROJECT (OUTPUT)

Output of guessing game:

Output of the rock paper scissors game:

BIBLIOGRAPHY

https://www.geeksforgeeks.org/

https://www.w3schools.com/

https://matplotlib.org/

https://pandas.pydata.org/docs/

https://stackoverflow.com/