

SOFTWARE UNIT TESTING REPORT

Student Name – Milan Sukhadiya

Student Id – S355516

GIT LINK - https://github.com/milansukhadiya/PRT582_ASSIGNMENT1

Table of Contents

Introduction.....	3
Objective	3
Requirements	3
Process	4
Conclusion	7
Reference	9

Introduction

In the modern world of development, different types of games have been developed using the python language for simplification and simple user interface. Among all games, **“Rock, Paper, and Scissor”** is a world-famous game for every human being all over the world. This is a type of hand game that is mainly played between two people. Three different types of shapes can be used in this interesting game and all of the shapes are: **“shape of rock (fist), paper (palm facing upward), or scissors (extended two fingers)”** (Bullock *et al.* 2022). There are mainly three rules that can be applicable at the time of playing this game and the rules are:

Rock vs Paper -> Paper wins

Rock vs Scissor -> Rock wins

Paper vs Scissor -> Scissor wins

For each win, the user gets 1 point, for each loss, the user loses 1 point, and for a tie, the user gets 0 points.

In this project work, the **“Rock, Paper, and Scissor”** game has been developed using python code with several frameworks and libraries.

Objective

- To implement a “Rock-Paper-Scissor” game in python
- To study the basic skills of game development

Requirements

1. The game has to be developed by taking the options from the users to choose the options.
2. The decisions have to develop with the options of the gameplay from the users.
3. The system of pointing is one point for the winner.
4. The first to reach a score of 5 will be the winners and it has to be displayed in the output sections.
5. There is the option for quitting and running the gameplay.
6. There is a code for the user to get quit anytime during their gameplay.

Process

```
In [*]: from random import randint
choice = ['rock', 'paper', 'scissor']
ch = 'y'
rounds = 0

def player_choice():
    plr_ch = input("\nPlease input complete word.\n\nEnter your choice Rock / Paper / Scissor: ")

    if plr_ch.lower() and plr_ch.lower() in ('rock', 'paper', 'scissor'):
        return plr_ch.lower()
    else:
        print("\nWrong Input... Please Retry...")
        player_choice()
```

Figure 1: Importing library function

The randint () has been used for the game as it holds the random integers between the two types of the given numbers that can be passed. This is the reason the randint function has been used.

```
def get_result(plr_ch):
    comp_choice = choice[randint(0,2)]
    print("\nComputer chose:", comp_choice, "\n")

    if plr_ch == comp_choice:
        result = "tie"
        print('{} is same as {}! No score change!'.format(plr_ch.upper(), comp_choice.upper()))
    elif comp_choice == 'scissor' and plr_ch == 'rock':
        result = 'win'
        print('ROCK Wins! Score +1')
    elif comp_choice == 'paper' and plr_ch == 'scissor':
        result = 'win'
        print('SCISSOR Wins! Score +1')
    elif comp_choice == 'rock' and plr_ch == 'paper':
        result = 'win'
        print('PAPER Wins! Score +1')
    else:
        result = 'lose'
        print('You lose! Score -1')
    return result
```

Figure 2: Defining the choices and generating the results

The above picture visualizes generating the solution with the conditions that have been used by printing the results that have been obtained after the end of the games.

```
def update_score(result):
    global wins, loss, tie
    if result == 'win':
        wins += 1
    elif result == 'lose':
        loss += 1
    else:
        tie += 1
```

Figure 3: Updating the scores

Update_score has been used for storing the results after each round in the *“Rock, Paper, and Scissor”* game.

```
def game(rounds):
    tot_score = 0
    global round_result
    for i in range(0,rounds):
        print("\nReady for Round", i+1)
        pc = player_choice()
        res = get_result(pc)
        round_result.append(res)
        update_score(res)
        tot_score = wins - loss
        print("\nAfter round",(i+1),"your score is: ",tot_score)

    return tot_score
```

Figure 4: Code to define the rounds

The game has been holding functions for generating the total score and printing the scores after the rounds.

```
def game_rounds(r = 0):
    r = input("\nEnter the number of rounds you want to play: ")

    try:
        global rounds
        rounds = int(r)
    except:
        print("\nWrong Input! Enter a number!")
        game_rounds()
```

Figure 5: Code to enter the number of rounds to be played

The above figure is defining of the number of times for playing the games.

```
def main():
    global ch, round_result
    print("\nWelcome to Rock, Paper, Scissor Game.")
    print('\nRules for Winning:\n
    Rock vs Paper -> Paper wins
    Rock vs Scissor -> Rock wins
    Paper vs Scissor -> Scissor wins\n
    For each win the user gets 1 point
    For each loss the user loses 1 point
    And for tie, the user gets 0 point\n')

```

Figure 6: Code to print the game rules

The above figure is defining the rules for the gameplay by printing the above things.

```
game_rounds()
ts = game(rounds)

print("\nAfter",rounds,"rounds, your final score is: ",ts)
print("\nYou have {} wins, {} ties and {} losses!".format(wins,tie,loss))
print("\nRound wise result is",round_result)
ch = input("\nDo you want to continue? Enter y for yes any other char to exit: ")
```

Figure 7: Code to enter the input if the player wants to continue or not

User input has been generated for the above figure and the gameplay.

```
while(ch == 'y' or ch == 'Y'):
    wins = 0
    loss = 0
    tie = 0

    round_result = []
    rounds
    main()
print("\nBye Bye!!")
```

Figure 8: Printing the result

The conditions have been used and the printing bye-bye after the gameplay has been visualized in the above image.

Welcome to Rock, Paper, Scissor Game.

Rules for Winning:

Rock vs Paper -> Paper wins
Rock vs Scissor -> Rock wins
Paper vs Scissor -> Scissor wins

For each win the user gets 1 point
For each loss the user loses 1 point
And for tie, the user gets 0 point

Enter the number of rounds you want to play:

Figure 9: Output of the rules related to the stone, paper, and scissor game

After the execution of the codes, the above figure has been generated. The user has to enter how many rounds he/she wants to play in the above box.

```
Enter the number of rounds you want to play: 1
```

```
Ready for Round 1
```

```
Please input complete word.
```

```
Enter your choice Rock / Paper / Scissor: 
```

Figure 10: Result of entering the round number and entering the choice

The above figure is defining the gameplay with that of the user choice and generates the winner. The user has to pick a single output in complete words in the respected area on the above image.

```
Please input complete word.
```

```
Enter your choice Rock / Paper / Scissor: Rock
```

```
Computer chose: paper
```

```
You lose! Score -1
```

```
After round 1 your score is: -1
```

```
After 1 rounds, your final score is: -1
```

```
You have 0 wins, 0 ties and 1 losses!
```

```
Round wise result is ['lose']
```

```
Do you want to continue? Enter y for yes any other char to exit: 
```

Figure 11: Output after the first game

The output results in the continuation of the gameplay for the users. The user has to enter y if he/she wants to continue or else he/she can enter anything except y to exit the game.

```
Do you want to continue? Enter y for yes any other char to exit: n
```

```
Bye Bye!!
```

Figure 12: Output after exiting the game

The above figure is generated with that of the end of the gameplay taking the information from the users.

Conclusion

The entire report is based on the gameplay of the rock, paper and scissors giving the required output and all of the screen shorts have been given in the above part of the report. Python has

been used for automation tasks, data visualization and analysis. The languages such as python have been adopted to develop the ***“Rock, Paper, and Scissor”*** game.

Reference

Bullock, E.A., Witherow, M.A. and Iftekharuddin, K.M., 2022. Hand Gesture Classification for Human-Robot Interaction in Rock-Paper-Scissors Game.