PYTHON PROJECT

INSTITUTE NAME	BESANT TECHNOLOGIES
PROJECT TITLE	PYTHON-POWERED ROCK , PAPER, SCISSOR ARENA.
SUBMITTED TO	GOWTHAMI
CO-ORDINATOR	AKILAN
SUBMITTED BY	MARIA MONISHA J
COLLEGE	RVS COLLEGE OF ARTS AND SCIENCE, COIMBATORE.
DEGREE/COURSE	B.SC.MATHEMATICS
LOCATION	VILLUPURAM
MAIL ID	hiremaria17@gmail.com
MOBILE	+91-7010807428

ABSTRACT

The aim of this project is to develop a Python-based implementation of the classic childhood game, Rock-Paper-Scissors. By leveraging Python's special features, readable syntax, and predefined modules, this project will demonstrate the simplicity and power of Python for creating interactive applications. The objective is to provide an engaging user experience, where the players are the user and the computer, while showcasing the efficiency and elegance of Python programming.

1. PYTHON CONCEPTS USED IN THIS PROJECT :

1. ”Import”-A Keyword:

“import” keyword is used to import modules or functions from the library .

2. “Random()”-A Module:

A module is a file that contains python definitions, classes, functions and variables. It acts as a container for related codes and can be imported into other python programs using the import statement. In this project , “random()” – a module has been imported to provide choices for the computer.

3. For Loop:

For loop is a loop used to iterate the values, where the starting and ending range is known. In this project, for loop is used to iterate the counts of chances which will help us to find the score of the player and the computer.

4. Conditional Statements:

(1) If Condition:

If condition forms a **true block** enclosed with indentation symbol. If block will work when the conditions were satisfied and print the statement inside the print function.

(2) Elif Condition:

Elif condition is an **alternative block** for if block enclosed with indentation symbol. In simple words, when the conditions given in the block failed to satisfy, the elf block will work and print the statements inside the print function if the conditions were satisfied.

(3) Else Condition:

Else condition is meant to be a **false block**. This block is also enclosed with an indentation symbol. If the conditions in the if block and elf block are failed to satisfy ,then the else block will work and display the output.

Here, the conditional statements are used to check some conditions of the game .

5. Logical Operators:

Logical operators are used to **combine, compare the conditions** and check whether the given condition is true or false and returns conditional statement. The logical operators applied in this project are **“and” and “or”.**

(1) And:

The “and” operator works only if all the conditions are true. If any one of the conditions is failed to satisfy , then “and” operator will not work.

(2) Or:

The “or” operator will work if any one of the given conditions is true. It will stop working only if all the conditions are failed to satisfy .

6. Relational Operators:

Relational operators will **compare or relate two variables,** conditions and expose the relation between two variables or values of conditions. some of the relational operators used in this project are:

1) “==”, is equal to.

2) ”>”, greater than.

3) ”<”, lesser than.

7. Print()-A Function:

Print(), a function used to print the outputs.

SOURCE CODE:

```python
user_point = 0
computer_point = 0

for i in range(1, 4):
    user = input("r for rock, p for paper, s for scissor: ").lower()
    choose = ["r", "p", "s"]
    computer = random.choice(choose)

    print(f"Computer chose: {computer}")
    print(f"You chose: {user}")


    if user == computer:
        print("It's a tie.")
    elif (user == 'r' and computer == 's') or (user == 's' and computer == 'p') or (user == 'p' and
computer == 'r'):
        user_point += 1
        print("You win this round.")
    else:
        computer_point += 1
        print("Computer wins this round.")

if user_point > computer_point:
    print("Congratulations! You won the game.")
elif user_point < computer_point:
    print("Well tried. Better luck next time.")
else:
    print("It is a tie.")
```

EXPLANATION OF THE SOURCE CODE:

1.Importing the random module :

Importing the random() using an import() statement to assign choices for computer to choose.

2.Initialising scores:

Initialize scores of the user and computer as 0.

3.Loop creation:

For loop is created with range of 1 to 4 which means that it will iterate the values three times.

4.User input:

Getting input from the user using input().the inputs are restricted as r for rock , p for paper and s for scissor. The lower() defines that the input will be taken as lower cases by the interpreter.

5.Array setting:

An array of choices for computer has been created . the array contains choices as r for rock , p for paper and s for scissor(choose=[“r”,”p”,”s”]) .The computer will choose any one of these choices using random module .i.e., random.choice(choose)

6.Printing the choices:

Printing the choices made by the user and the computer using print().

7.Rules for the game:

Rule1: If both the user and computer chose same choice, the game will end up as a tie.

Example: User=s

Computer=s

Output: it is a tie.

Rule2: If the user chooses rock and the computer chose scissor , then the winner is user.

Example: User=r

Computer=s

Output: user is the winner.

Rule3: The user will be considered as the winner , if he/she chooses scissor and the computer chooses paper

Example: User=s

Computer=p

Output: the user won.

Rule4: The user will be again considered as winner , if he/she chooses the paper and the computer chooses the rock.

Example: User=p

Computer=r

Output: user is the winner.

Rule5: If any of the above rules is not followed , then the computer will be considered as the winner .

8.Score Updating:

The score of the user will get increased by 1 after satisfying the conditions in the elif block where the conditions are made for the winning of the user. If it is not , then the computer's score will get increased by 1. If it is a tie, then there will be no increase in the score of both the players. This process will repeat 3 times as the range is mentioned as 1 to 4 which is actually excluding the last number.(0,1,2,3)

9.Finding The Winner:

If the score of the user is greater than the score of the computer, then the user is supposed to win the game. The computer will win after getting high score than the user. If the scores of both the players are same , then the game is ended up as tie.

OUTPUT FOR THE SOURCE CODE :

Round 1:

```
r for rock, p for paper, s for scissor:  r
Computer chose: p
You chose: r
Computer wins this round.
```

Round 2:

```
r for rock, p for paper, s for scissor:  s
Computer chose: p
You chose: s
You win this round.
```

Round 3:

```
r for rock, p for paper, s for scissor:  p
Computer chose: s
You chose: p
Computer wins this round.
```

Final result : Well, tried. Better luck next time.

CONCLUSION

Through this project, I have enhanced my understanding of Python programming and have se en firsthand how a few lines of code can come together to create an interactive application. Th e game is user-friendly and demonstrates the fundamental principles of game development, in cluding user input handling, computer decision-making, and result determination based on pre defined rules.