Computer Project #3

Assignment Overview

This assignment focuses on the design, implementation and testing of a Python program that plays the game Rock-Paper-Scissors.

It is worth 30 points (3% of course grade) and must be completed no later than 11:59 PM on Monday, September 28.

Assignment Deliverable

The deliverable for this assignment is the following file:

> `proj03.py` – the source code for your Python program

Be sure to use the specified file name and to submit it for grading via the **handin system** before the project deadline.

Assignment Background

The game of Rock-Paper-Scissors is a hand game usually played between two people in which each player simultaneously forms one of three shapes with an outstretched hand (see https://en.wikipedia.org/wiki/Rock-paper-scissors). The rules are:

> • Rock smashes (or blunts) Scissors, so Rock wins.
> • Scissors cut Paper, so Scissors win.
> • Paper covers Rock, so Paper wins.
> • If players choose the same object, it is a tie.

Assignment Specifications

You will develop a Python program which allows the user to play the Rock-Paper-Scissors game against the computer.

1. The program will first announce the game rules.

2. The program will repeatedly prompt the user to choose a shape (a lower case letter 'r', 'p', or 's' representing Rock, Paper, and Scissors, respectively) or to quit playing the game (the lower case letter 'q').   Any other input will cause the program to display a message and prompt again until valid input is entered.

3. The program will make a random choice for its own shape, then determine and announce the winner of the current round.

4. After the user chooses to quit playing the game, the program will display the following:

   a. The number of rounds won by the user (actual count and percentage).
   b. The number of rounds won by the computer (actual count and percentage).
   c. The number of rounds which were ties (actual count and percentage).
   d. The number of times the user chose Rock (actual count and percentage).
   e. The number of times the user chose Paper (actual count and percentage).
   f. The number of times the user chose Scissors (actual count and percentage).

   All output will be appropriately labeled; percentages will be displayed with one decimal place of accuracy.

Assignment Notes

1. To clarify the project specifications, sample output is provided at the end of this document.

2. The coding standard for CSE 231 is posted on the course website:

   http://www.cse.msu.edu/~cse231/General/coding.standard.html

3. Items 1-7 of the Coding Standard will be enforced for this project.

4. Your program must use the following function call to choose the computer's shape:

   ```
   .... = random.randint(1,3)
   ```

   That call to function `randint` returns an integer value between 1 and 3 (inclusive). Your program must use the following correspondence between integer values and shapes:

   1: Rock
   2: Paper
   3: Scissors

5. As in the previous project, the following two lines must appear in your program:

   ```
   import random
   random.seed( 0 )
   ```

   The first statement allows your program to access the `random` module, including function `randint`. The second statement initializes the random number generator so that the same sequence of values is generated every time you execute your program, which is useful when you develop and test your program (and when your TA grades it).

6. If you want an additional challenge (not a requirement), use *one* Boolean expression to choose when the user wins a round.

Suggested Procedure

- *Solve the problem using pencil and paper first.* You cannot write a program until you have figured out how to solve the problem. This first step can be done collaboratively with another student. However, once the discussion turns to Python specifics and the subsequent writing of Python statements, you must work on your own.

- Cycle through the following steps to incrementally develop your program:

  o Edit your program to add new capabilities (e.g. the first iteration might just be a loop that only accepts valid input, then create a game that loops until 'q' is input, and so on).

  o Use a divide-and-conquer approach to solving the problem. For example, begin by simply having choices be made without determining a winner—simply print the choices. Run the program and fix any errors. Next, add choosing a winner. Once that is tested and working, begin collecting statistics one at a time.

  o Use the **handin** system to submit the current version of your program.

- Be sure to use the **handin** system to submit the final version of your program.

- Be sure to log out when you leave the room, if you're working in a public lab.

The last version of your solution is the program which will be graded by your TA.

*You should use the **handin** system to back up your partial solutions, especially if you are working close to the project deadline. That is the easiest way to ensure that you won't lose significant portions of your work if your machine fails or there are other last-minute problems.*

*You would also be wise to save a copy of your completed program in your CSE file space (the H: drive on the lab machines) **before** the project deadline. If you write your program at home and turn it in from home, you should copy it to your CSE file space **before** the deadline.*

*In case of problems with electronic submission, an archived copy in the **handin** system or the CSE file space is the only acceptable evidence of completion.*

Sample Output

```
Welcome to the Rock-Paper-Scissors game.
Enter a single character: r, s, p, or q to quit.
Rock beats Scissors which beats Paper which beats Rock.

Enter a command (rpsq): s
User chose s and computer chose p
User wins this round
-----------------------

Enter a command (rpsq): r
User chose r and computer chose p
Computer wins this round
-----------------------

Enter a command (rpsq): s
User chose s and computer chose r
Computer wins this round
-----------------------

Enter a command (rpsq): p
User chose p and computer chose p
This round is a tie
-----------------------

Enter a command (rpsq): s
User chose s and computer chose s
This round is a tie
-----------------------

Enter a command (rpsq): r
User chose r and computer chose p
Computer wins this round
-----------------------

Enter a command (rpsq): x
Error. Try again.
Enter a command (rpsq): Q
Error. Try again.
Enter a command (rpsq): q

Summary statistics
  User wins: 1 (16.7%)
  Computer wins: 3 (50.0%)
  Ties: 2 (33.3%)

User statistics
  Rock: 2 (33.3%)
  Paper: 1 (16.7%)
  Scissors: 3 (50.0%)
```