Lab 03 Math Quiz

Elliot Graves, Brigham Ray

9/24/2016

Section 4

1. Problem Statement

We are given preset program that we need to improve upon to add three different difficulty levels for the user to answer math problems and then give them feedback on how well they did.

Requirements include

* Improving upon the premade quiz
* Add three levels to the program, beginner, Intermediate, and Advanced
* Having the user be able to input which difficulty to choose from and how many problems they want to solve.
* Giving feedback after all problems are solved on how many correct and how many wrong.

Constraints include

* Using basic math problems that include +, -, \*, /, //, %, \*\*,
* Only altering upon the premade math quiz given

Assumptions include

* The user has the ability to solve basic math problems
* The user has knowledge of the basic operands python uses for solving math problems.

1. Planning

We began by inspecting and typing out the given basic math quiz from the lab to check how it worked and began looking at how we could alter it following the steps in the lab manual. Following on step one, we began adding in new code to allow the user to be able to select how many questions they want. After this was implemented we ran a test and then began on step two, which was to add different difficulty levels. We found this to be a little challenging as we needed the user to be able to select which difficulty they wanted as well as changing the operands to higher numbers and add in multiplication and division to the higher levels.

To start out we will import randint to give the test a wee bit of variation. This will allow each of the problems in our test to maintain difficulty level.

#from random import randint

Then we will give the user a choice of both the number of questions and the difficulty level of the test they are going to take. They can choose in between one to five questions and from three difficulty levels: easy, medium, hard.

#the user gets to choose from a level

levels = ["easy", "medium", "hard"]

user\_input = input("choose a level: ")

#the user will choose from 5 lovely questions

user\_numofquestions = input("How many questions from 1 to 5? ")

num\_of\_questions = ["1", "2", "3", "4", "5"]

Each question is represented by an if statements. The code will execute the number of if statements that actually run based on how many questions the user chooses. Then they will be assigned a score out of 5 and told if they need to improve or not based on how they scored on the test.

#start of the test of first level "easy"

#each if statement represents a new question

if user\_input == levels[0]:

correct = 0

for i in range(1, int(user\_numofquestions) + 1):

if i == 1:

n1 = randint(1, 10)

n2 = randint(1, 10)

prod = n1 + n2

ans = input("What is %d plus %d " % (n1, n2))

if int(ans) == prod :

print("That's right, well done.\n")

correct = correct + 1

else:

print("No, I'm afraid the answer is %d.\n" % prod)

if i == 2:

n1 = randint(1, 10)

n2 = randint(1, 10)

prod = n1 - n2

ans = input("What is %d minus %d " % (n1, n2))

if int(ans) == prod :

print("That's right, well done.\n")

correct = correct + 1

else:

print("No, I'm afraid the answer is %d.\n" % prod)

if i == 3:

n1 = randint(1, 10)

n2 = randint(1, 10)

prod = n1 + n2

ans = input("What is %d plus %d " % (n1, n2))

if int(ans) == prod :

print("That's right, well done.\n")

correct = correct + 1

else:

print("No, I'm afraid the answer is %d.\n" % prod)

if i == 4:

n1 = randint(1, 10)

n2 = randint(1, 10)

prod = n1 - n2

ans = input("What is %d minus %d " % (n1, n2))

if int(ans) == prod :

print("That's right, well done.\n")

correct = correct + 1

else:

print("No, I'm afraid the answer is %d.\n" % prod)

if i == 5:

n1 = randint(1, 10)

n2 = randint(1, 10)

prod = n1 + n2

ans = input("What is %d plus %d " % (n1, n2))

if int(ans) == prod :

print("That's right, well done.\n")

correct = correct + 1

else:

print("No, I'm afraid the answer is %d.\n" % prod)

#the results of the testing to show the user and give advice

print("I asked you " + str(int(user\_numofquestions)) + " questions you got " + str(int(correct)) + " right")

if correct / int(user\_numofquestions) <= 1 and correct / int(user\_numofquestions) > .666:

print("You done good young lad.")

elif correct / int(user\_numofquestions) <= .666 and correct / int(user\_numofquestions) >= .333:

print("Study some more!!")

elif correct / int(user\_numofquestions) < .333 and correct / int(user\_numofquestions) >= 0:

print("Ask the teacher for help.")

The same statement will be made for 3 tests and the code will execute accordingly!

1. Implementation and testing

After all was planned we began testing the code and fixing the various syntax errors and smoothing out the program.

1. Reflection

After much trial and error, the code finally runs! We learned a lot of things from this. First off, adding a forward slash at the end of a line allowed us to continue as if we were just writing on the same line. This was useful in getting rid of the extra characters which kept the code from being pep8 compliant. We realized a plus character would not do this exact function if there was no string following because of python syntax rules. We were able to use the data from a few arrays to our advantage making the code extremely easy to create. It ended up looking very concise because the code ended up visually looking like the code for just 3 tests and

not a whole lot of ambiguity. Knowing beforehand the rules of python syntax and having a higher quality editor would have allowed us to create more formidable python code from the start. From the outset, we had no idea of how to do this lab but creating a few arrays and a for loop allowed us to make 3 user created tests to run.