Ryan Demboski
Jordan Colebank
10/21/2019
Lab 7 - Game Show
Section 001

Lab 7 Report

1. Problem Statement:

The task we were given was creating a trivia game like you would see in a generic TV game show. The requirements of this lab were the following:

- Must have 10 questions that shuffle randomly and appear once each.
- Have a main menu where the user could choose to play the game, view the credits, or quit the program.
- An input for an answer that was not a valid answer (such as being given answer choices a, b, c, d, but the user typed w. The user needs to enter a valid answer in order to move on.)
- Score should be tracked, as well as the amount of questions asked at the time. For instance, getting 4 out of 10 correct.
- Play game and view credits should be functions.

2. Planning:

The first step we took in making this program was creating the data structure that held all of our questions and answers. We did this by making one list, with each question being inside the list held in a dictionary for each one. Next, we created our three functions that we needed for this assignment (play game, view credits, and quit game). Inside of the play game function is where our main code is held. This includes our for loops and if statements that allowed the program to function as intended. The view credits function was next, and lastly we finished with the quit game function, and tied it all together inside of the play game function so the user didn't have to relaunch the program to play again.

3. Implementation and Testing:

Pep 8 Compliance:

```
jec447@tatooine:~/csl26/labs/lab7$ pycodestyle-3 rad449_jec447_gameshow.py
jec447@tatooine:~/csl26/labs/lab7$
```

Code running:

```
Your current score is 0 out of 10
Jordan Colebank and Ryan Demboski created this game!
Enter PLAY to play the game, VIEW to view the credits,or QUIT to quit
```

```
Enter PLAY to play the game, VIEW to view the credits,or QUIT to quit
VIEW
Jordan Colebank and Ryan Demboski created this game!
Enter PLAY to play the game, VIEW to view the credits,or QUIT to quit
QUIT
Thanks for playing!
jec447@tatooine:~/csl26/labs/lab7$
```

Our results show that the program works exactly as intended per the requirements and is also pep 8 compliant. After the user plays the game, it loops back to the main menu, where the user can then choose whether to play again or quit, just like they could in the beginning. The score shows how many questions were correct out of the total amount given.

4. **Reflection and Refactoring:**

Our solution to this lab met all of the desired requirements as stated above. The major approach to this lab was to make sure that our play_game function was running. Once we got the function to run properly, we then manipulated it to where it fit the requirements (ie. having the questions randomized, non-repeating, etc.) After we went in and built the view and quit functions which didn't take too long to do. An alternative approach to this lab would possibly be to make sure our play_game function works before creating the entire list of questions to give us more time in class to write the play_game function with all of the requirements. Later we could then implement random questions in to our list.