Game Show Lab 7

Brigham Ray

Elliot Graves

10-3-2016

Lab Section 3

Game Show

1. Understand the Problem:

REQ: Game must have a theme smarter than a 5th

grader, game needs 10 questions and some questions have more correct answers

than others. No question can be given 2 times and they should be given in

random order. Main menu says 1) Play game 2) view credits 3) quit. Every time

the user finishes the game, they have the option to quit. The user has the

option to say quit if they want to go back to

the main menu. When user enters an invalid response, then it asks them to input

a valid response. After each question the score must be displayed.

ASSUMPTIONS: once the user stats the game they will play all the way through, quit

will only be available at the end of the game(main menu). The game is going

to be a list of dictionaries with key values as both the question and answers.

The main menu will allow the user to view credits and when the user quits the game restarts.

2. Plan the Solution:

# from random import shuffle

from random import shuffle

# Create the list of dictionaries

questions = [{"question": "Who Sailed Across the Ocean Blue?",

"answers": ["Christopher Columbus", "James", "John",

"Batman"],

"correct": "1"},

{"question": "What year was the first independence day?",

"answers": ["1776", "year 1776", "when the country was formed",

"on a good day"],

"correct": "1"},

{"question": "What was the slave trade called?",

"answers": ["triangular trade"],

"correct": "1"},

{"question": "Who Shot Abraham Lincoln?",

"answers": ["John Wilkes Booth", "Idiot filty"],

"correct": "1"},

{"question": "What color is the sky?",

"answers": ["blue", "black", "green"],

"correct": "2"},

{"question": "When was google established?",

"answers": ["1998", "1997", "2020", "1775"],

"correct": "1"},

{"question": "What famous mountains are in Nevada?",

"answers": ["Sierras", "Appalacians", "Rockies",

"Humphrey's peak"],

"correct": "1"},

{"question": "How many fingers does a human have?",

"answers": ["1", "2", "10"],

"correct": "3"},

{"question": "What university is in Flagstaff Arizona?",

"answers": ["ASU", "NAU", "UA"],

"correct": "2"},

{"question": "What is my major?",

"answers": ["CS", "Pottery", "Gender Studies", "Anthropology"],

"correct": "3"}]

# The length of questions dictionary will be assigned to a variable to be used

# in a for loop

len\_questions = len(questions)

# the number correct will be updated with each answer correct and reset each

# time the game restarts

correct = 0

# this is the main menu which will start the game and welcome the user. The

# Game is in another function which will be called from this one

def main\_menu():

print("Welcome to the game(Main Menu)!!!!!")

print(" ")

# User chooses what they want to do

user\_input3 = input("What would you like to do?: play, quit, credits ")

print(" ")

if user\_input3 == "quit":

# Main Menu will appear again if they quit

main\_menu()

elif user\_input3 == "credits":

print("Brig and Elliot made this fun little game")

main\_menu()

else:

# if they don't quit or view credits the game will start

main\_menu2()

correct = 0

# start of the game itself

def main\_menu2():

# calling correct from global allows you to call the variable from the global

# scope

global correct

print("Welcome to Are you Smarter Than A First Grader!!")

print("LETS FIND OUT!!")

# shuffle questions for user

shuffle(questions)

# This loop will iterate until the end of the length of the questions.

for e in range(0, len\_questions - 1):

# While statement allows you to repeat an iteration of a loop if the user

# chooses an incorrect response while the range is in the range of e

while True:

print(questions[e]["question"])

for i, choice in enumerate(questions[e]["answers"]):

print(str(i + 1) + ". " + choice)

answer = input("Choose an answer: ")

if answer == questions[e]["correct"]:

print("Nice Job First Grader")

correct += 1

print("Your current score is " + str(correct) + " /10")

# This will allow you to go to the next iteration if the answer is correctf

e += 1

if e > len\_questions - 1:

return results()

else:

continue

# this will simply continue on the same iteration of the loop until you answer

# the question correctly

else:

print("Sorry, Try again figgy.")

correct -= 1

continue

# this function prints the results and allows you to restart the game, view

# credits, and quit the game

def results():

# this will call correct from the global variable updated in the previous

# function to show the user what they got.

global correct

print("You got " + str(correct) + " questions coorect!")

print("Thanks for playing sweet lover!~! ")

user\_input = input("What would you like to do next? quit, restart," +

" credits")

if user\_input == "quit":

print("okay sweetie, have a nice day!!")

correct = 0

main\_menu()

elif user\_input == "restart":

correct = 0

main\_menu2()

else:

correct = 0

print("Brig and Elliot made this fun little game")

main\_menu()

# this piece of code initializes the main menu so the user can start playing

# the

# game

main\_menu()

3. Implementation:

The code runs according to plan! At first there was no results function which limited the code to only run a certain number of times. This was problematic since the code needed to rerun each time the user played the game. Making the function changed the code so it could do just that. No issues were encountered apon revision of the program.

4. Reflect and re-factor:

Reflect: Although the code ran according to plan, having multiple functions was very ambiguous and hard to write. It is also not a very good idea to use global variables. The reason being is that they cannot often be called within functions without using the global command within the function which would not have been necessary had the variable correct been in a function. Even though the code was slightly ambiguous and had some minor syntax glitches, it ran exactly as it should have. Figuring out how to repeat the problem after it was answered incorrectly was also met with issues. A while loop is the only way to repeat an iteration of a loop . As soon as a while loop was implemented, the program repeated a question if it was answered incorrectly. The question in questions had to be changed to a loop with a specific numbered range so that if the question was answered correctly the loop would go to the next iteration.