Nelson Rengifo

Abstract

The “Math Workshop” project aims to provide a practical and interactive platform for arithmetic practice. Inspired by the “Quick Math” mobile app, the project generates arithmetic problems based on user-selected difficulty levels and types, offering immediate feedback to enhance the learning experience. This report outlines the goals, software requirements, and the concept behind choosing and implementing this project.

Math workshop

Introduction to Computer Science | COP 1500 | Deepa Devasenapathy

Table of Contents

Table of Contents1

Goal of Project2

**Software Requirements3**

Introduction4

Flow Diagram5

**Control Flow6**

Appendix7-17

**Goal of the Project:**

The primary goal of the math workshop project is to facilitate arithmetic practice with an emphasis on immediate feedback. By generating random math problems based on user preferences, the project aims to create an engaging learning environment. Users can select from three difficulty levels (Easy, Medium, Hard) and four problem types (Addition, Subtraction, Multiplication, Division) to tailor their practice sessions.

**Software Requirements:**

To run the Math Workshop project, the following software and hardware requirements should be met:

* **Operating System**: Windows, MacOS, Linux
* **Python**: Version 3
* **Terminal or Command Prompt**: To execute the program via the command line
* **Input Devices**: Keyboard, Mouse
* **Output Device**: Monitor
* **Processor**: Intel Core i3 or equivalent
* **RAM**: 2GB

**Introduction**

**Why Did You Chose This Project?**

The decision to create this project emerged from a need for the challenges associated with traditional mobile applications of arithmetic practice. Conventional mobile applications often lack the engaging elements needed to captivate learners, resulting in burnout or disinterest in the application after a few tries. The aim was to use what I learned during the semester and create a practical tool that not only generates arithmetic problems but also provides immediate feedback in a game like manner to entice the user.

**Existing Technique and Your Improvisations**

The current market of math practice applications, notably the “Quick Math” mobile app, served as my source of inspiration. These applications demonstrated the program needed to generate problems of different types and difficulties while keeping it interactive and enjoyable. However, recognizing certain limitations, particularly the lack of immediate feedback on some problems, prompted my idea of adding it to the existing formula.

The key improvisation introduced in the Math Workshop project is the incorporation of immediate feedback after an incorrect answer is detected from the user. Now, the user not only encountered a wide range of randomly generated problems but also receives instant information about what they got wrong and the correct answer. The aim of this feedback mechanism is to enhance the “learning” nature of the practice problems, turning each problem into a lesson and not just a random problem.

Overall, the choice of the project was driven by the idea of improving an already great idea by adding the immediate feedback feature. This offered a more dynamic and interactive application for arithmetic practice.

**Flow Diagram**

A diagram of a flowchart

Description automatically generated

**Control Flow of Code**

1. Execution Start:
   1. The program starts executing from the main() function.
2. User Input for Difficulty and Problem Type:
   1. The get\_difficulty() function is called to get user input for the difficulty level and problem type.
   2. The user is prompted to enter the difficulty level (1 for Easy, 2 for Medium, 3 for Hard) and problem type (1 for addition, 2 for subtraction, 3 for multiplication, 4 for division).
   3. Input validation is performed to ensure the entered values are valid.
3. Problem Generation Loop:
   1. The get\_problem() function is called with the obtained difficulty level and problem type.
   2. Inside the get\_problem(), a loop runs 15 times to generate 15 problems based on the user’s choices.
   3. For each problem, the program selects random integers based on difficulty and problem type and asks the user to solve the problem.
   4. The user’s answer is compared with the correct answer, and feedback is provided.
   5. User responses and correct answers are recorded in lists like answer\_list and correct\_answer\_list.
4. Score Calculation:
   1. The get\_score() function is called to calculate the user’s score based on the entered answers and correct answers.
5. Score and Time Display:
   1. The final score is calculated as a percentage and displayed to the user.
   2. The total time taken for the 15 problems is also displayed in minutes and seconds.
6. Program Termination:
   1. The program execution ends.

**Appendix**

import random

import sys

import time

def main():

difficulty\_level, problem\_type = get\_difficulty()

answer\_list, correct\_answer\_list, total\_time = get\_problem(

difficulty\_level, problem\_type)

score = get\_score(answer\_list, correct\_answer\_list)

score = (score / 15) \* 100

score = round(score)

total\_minutes, total\_seconds = divmod(int(total\_time), 60)

print(

f"\nYour score was {score}% and time was {total\_minutes} minutes and {total\_seconds} seconds!")

def get\_difficulty():

difficulty\_level = ""

problem\_type = ""

print("""

Welcome to Math Workshop!

To begin, type the number of the difficulty level:

1. Easy

2. Medium

3. Hard

Next, type the number of the desired problem type:

1. Addition

2. Subtraction

3. Multiplication

4. Division

Each level has 15 questions and a timer to challenge yourself!

""")

while True:

try:

difficulty\_level = input("\nType Difficulty Here: \n")

if difficulty\_level not in ["1", "2", "3"]:

print(

"\nPlease, choose one of the difficulty levels from the above options\n")

else:

break

except KeyboardInterrupt:

answer = input(

"\nAre you sure you wish to end the program? (Yes) (No) \n").lower()

if answer == "yes":

sys.exit(0)

else:

continue

while True:

try:

problem\_type = input(

"\nType Problem Here: \n")

if problem\_type not in ["1", "2", "3", "4"]:

print(

"\nPlease, choose the desired problem and type the number associated with it.\n")

else:

break

except KeyboardInterrupt:

answer = input(

"\nAre you sure you wish to end the program? (Yes) (No) \n").lower()

if answer == "yes":

sys.exit(0)

else:

continue

return (difficulty\_level, problem\_type)

def get\_problem(difficulty\_level, problem\_type):

problem\_count = 0

answer\_list = []

correct\_answer\_list = []

start\_time = time.time()

while problem\_count != 15:

end\_time = time.time()

total\_time = end\_time - start\_time

try:

if difficulty\_level == "1" and problem\_type == "1":

integer\_1 = random.randint(1, 9)

integer\_2 = random.randint(1, 9)

print(f"{integer\_1} + {integer\_2}?")

answer = int(input("Type answer here: "))

answer\_list.append(answer)

correct\_answer = integer\_1 + integer\_2

correct\_answer\_list.append(correct\_answer)

if answer == correct\_answer:

print("\nCorrect!\n")

else:

print(f"\nIncorrect! Correct Answer: {correct\_answer}\n")

problem\_count += 1

elif difficulty\_level == "2" and problem\_type == "1":

integer\_1 = random.randint(1, 99)

integer\_2 = random.randint(10, 99)

print(f"{integer\_1} + {integer\_2}?")

answer = int(input("Type answer here: "))

answer\_list.append(answer)

correct\_answer = integer\_1 + integer\_2

correct\_answer\_list.append(correct\_answer)

if answer == correct\_answer:

print("\nCorrect!\n")

else:

print(f"\nIncorrect! Correct Answer: {correct\_answer}\n")

problem\_count += 1

elif difficulty\_level == "3" and problem\_type == "1":

integer\_1 = random.randint(1, 100)

integer\_2 = random.randint(100, 999)

print(f"{integer\_1} + {integer\_2}?")

answer = int(input("Type answer here: "))

answer\_list.append(answer)

correct\_answer = integer\_1 + integer\_2

correct\_answer\_list.append(correct\_answer)

if answer == correct\_answer:

print("\nCorrect!\n")

else:

print(f"\nIncorrect! Correct Answer: {correct\_answer}\n")

problem\_count += 1

elif difficulty\_level == "1" and problem\_type == "2":

integer\_1 = random.randint(1, 9)

integer\_2 = random.randint(1, 9)

print(f"{integer\_1} - {integer\_2}?")

answer = int(input("Type answer here: "))

answer\_list.append(answer)

correct\_answer = integer\_1 - integer\_2

correct\_answer\_list.append(correct\_answer)

if answer == correct\_answer:

print("\nCorrect!\n")

else:

print(f"\nIncorrect! Correct Answer: {correct\_answer}\n")

problem\_count += 1

elif difficulty\_level == "2" and problem\_type == "2":

integer\_1 = random.randint(1, 99)

integer\_2 = random.randint(10, 99)

print(f"{integer\_1} - {integer\_2}?")

answer = int(input("Type answer here: "))

answer\_list.append(answer)

correct\_answer = integer\_1 - integer\_2

correct\_answer\_list.append(correct\_answer)

if answer == correct\_answer:

print("\nCorrect!\n")

else:

print(f"\nIncorrect! Correct Answer: {correct\_answer}\n")

problem\_count += 1

elif difficulty\_level == "3" and problem\_type == "2":

integer\_1 = random.randint(1, 100)

integer\_2 = random.randint(100, 999)

print(f"{integer\_1} - {integer\_2}?")

answer = int(input("Type answer here: "))

answer\_list.append(answer)

correct\_answer = integer\_1 - integer\_2

correct\_answer\_list.append(correct\_answer)

if answer == correct\_answer:

print("\nCorrect!\n")

else:

print(f"\nIncorrect! Correct Answer: {correct\_answer}\n")

problem\_count += 1

elif difficulty\_level == "1" and problem\_type == "3":

integer\_1 = random.randint(1, 9)

integer\_2 = random.randint(1, 9)

print(f"{integer\_1} \* {integer\_2}?")

answer = int(input("Type answer here: "))

answer\_list.append(answer)

correct\_answer = integer\_1 \* integer\_2

correct\_answer\_list.append(correct\_answer)

if answer == correct\_answer:

print("\nCorrect!\n")

else:

print(f"\nIncorrect! Correct Answer: {correct\_answer}\n")

problem\_count += 1

elif difficulty\_level == "2" and problem\_type == "3":

integer\_1 = random.randint(1, 99)

integer\_2 = random.randint(10, 99)

print(f"{integer\_1} \* {integer\_2}?")

answer = int(input("Type answer here: "))

answer\_list.append(answer)

correct\_answer = integer\_1 \* integer\_2

correct\_answer\_list.append(correct\_answer)

if answer == correct\_answer:

print("\nCorrect!\n")

else:

print(f"\nIncorrect! Correct Answer: {correct\_answer}\n")

problem\_count += 1

elif difficulty\_level == "3" and problem\_type == "3":

integer\_1 = random.randint(1, 100)

integer\_2 = random.randint(100, 999)

print(f"{integer\_1} \* {integer\_2}?")

answer = int(input("Type answer here: "))

answer\_list.append(answer)

correct\_answer = integer\_1 \* integer\_2

correct\_answer\_list.append(correct\_answer)

if answer == correct\_answer:

print("\nCorrect!\n")

else:

print(f"\nIncorrect! Correct Answer: {correct\_answer}\n")

problem\_count += 1

elif difficulty\_level == "1" and problem\_type == "4":

integer\_1 = random.randint(1, 9)

integer\_2 = random.randint(1, 9)

print(f"{integer\_1} / {integer\_2}?")

answer = float(

input("Type answer here (round to 2 decimal places): "))

answer\_list.append(answer)

correct\_answer = round(integer\_1 / integer\_2, 2)

correct\_answer\_list.append(correct\_answer)

if answer == correct\_answer:

print("\nCorrect!\n")

else:

print(f"\nIncorrect! Correct Answer: {correct\_answer}\n")

problem\_count += 1

elif difficulty\_level == "2" and problem\_type == "4":

integer\_1 = random.randint(1, 99)

integer\_2 = random.randint(10, 99)

print(f"{integer\_1} / {integer\_2}?")

answer = float(

input("Type answer here (round to 2 decimal places): "))

answer\_list.append(answer)

correct\_answer = round(integer\_1 / integer\_2, 2)

correct\_answer\_list.append(correct\_answer)

if answer == correct\_answer:

print("\nCorrect!\n")

else:

print(f"\nIncorrect! Correct Answer: {correct\_answer}\n")

problem\_count += 1

elif difficulty\_level == "3" and problem\_type == "4":

integer\_1 = random.randint(1, 999)

integer\_2 = random.randint(100, 999)

print(f"{integer\_1} / {integer\_2}?")

answer = float(

input("Type answer here (round to 2 decimal places): "))

answer\_list.append(answer)

correct\_answer = round(integer\_1 / integer\_2, 2)

correct\_answer\_list.append(correct\_answer)

if answer == correct\_answer:

print("\nCorrect!\n")

else:

print(f"\nIncorrect! Correct Answer: {correct\_answer}\n")

problem\_count += 1

except ValueError:

print(

"\nMake sure to type an integer as your answer. If decimal, round to 2 decimal places.\n")

continue

except KeyboardInterrupt:

answer = input(

"\nAre you sure you wish to end the program? (Yes) (No) \n").lower()

if answer == "yes":

sys.exit(0)

else:

continue

return (answer\_list, correct\_answer\_list, total\_time)

def get\_score(answer\_list, correct\_answer\_list):

score = 0

for user\_input, answer in enumerate(answer\_list):

if answer == correct\_answer\_list[user\_input]:

score += 1

return score

if \_\_name\_\_ == "\_\_main\_\_":

main()