Quick Math Game

INTRODUCTION

The Quick Math © game provides the user a stream of arithmetic expression in a set amount of time and scores the user based on the number of questions answered in that set time.

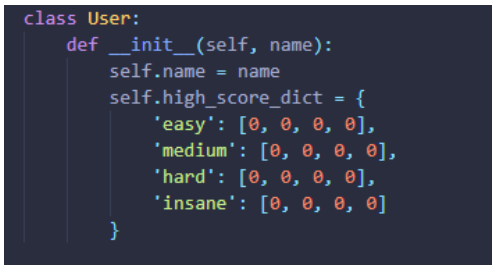
KEY FEATURES

- Multiple modes (Easy, Medium, Hard and Insane)
    - Easy (3 operands (random operators between 1 to 5) and 2 operators (+ or – 50/50 chance)) e.g., 3 + 2 - 1
    - Medium (3 operands (random operators between 1 to 5) and 2 operators (+, – or * 40/ 40/ 20 chance)) e.g., 3 * 2 + 1
    - Hard (4 operands random operators between 1 to 9) and 4 possible operators (+, -, * or / 20/20/30/30 chance)) e.g., 3 * 2 + 1 - 4
    - Insane (5 operands random operators between 1 to 9) and 5 possible operators (+, -, *, / or ** 10/10/30/30/20 chance)) e.g., 3 ** 2 + 5 – 4 * 3
- Different Durations (30 secs, 60 secs, 1 minute and 2 minutes)
- Multiple users
- High score table

DESIGN AND IMPLEMENTATION

For the implementation of this game, 2 classes were created

1. User: creates user object with name attribute and their high scores for each difficulty and duration



```
class User:
    def __init__(self, name):
        self.name = name
        self.high_score_dict = {
            'easy': [0, 0, 0, 0],
            'medium': [0, 0, 0, 0],
            'hard': [0, 0, 0, 0],
            'insane': [0, 0, 0, 0]
        }
```

2. QuickMaths: starts a game session and updates the user high score attributes dictionary if necessary

DIFFICULTY LEVEL

Each difficulty level is designed such that selection of operands and operators are randomly generated with varying level of probability for each case.

- **Easy**: In easy mode, 3 operands are randomly generated using Python's Random method randrange with values between 1 – 5 and the operators are selected such that there is 50/50 chance of getting a plus or minus. This is implemented using a list of containing '+' and '-' and passing that list to the Python's Random choice method
- **Medium**: In medium mode, 3 operands are randomly generated using Python's Random method randrange with values between 1 – 7 and the operators are selected such that there is 40/40/20 chance of getting a plus, minus or times. This is implemented using a list of containing 2 '+', 2 '-' and 1 '*' and passing that list to the Python's Random choice method
- **Hard**: In hard mode, 4 operands are randomly generated using Python's Random method randrange with values between 1 – 9 and the operators are selected such that there is 20/20/30/30 chance of getting a plus, minus, times or division. This is implemented using a list of containing 2 '+', 2 '-', 3 '*' and 3 '/' and passing that list to the Python's Random choice method. A check is made to ensure that the result is never not a round figure. If an expression is generated such that it is not a round figure, another is generated (since this occurs in milliseconds). It has little to no effect on integrity of the game. For an extra kick a bracket is randomly inserted using a function named 'insert brackets', it is called randomly (50/50) using the Random's choice method and a list with True and False as items

- **Insane**: In insane mode, 4 operands are randomly generated using Python's Random method randrange with values between 1 – 9 and the operators are selected such that there is 10/10/40/30/10 chance of getting a plus, minus, times, division or exponent. This is implemented using a list of containing 1 '+', 1 '-', 4 '*', 3 '/' and 1 '**' and passing that list to the Python's Random choice method. A check is made to ensure that the result is never not a round figure. If an expression is generated such that it is not a round figure, another

is generated (since this occurs in milliseconds). It has little to no effect on integrity of the game. For an extra kick a bracket is randomly inserted using a function named ‘insert brackets’, it is called randomly (50/50) using the Random’s choice method and a list with True and False as items

GAME DURATION

Each difficulty level can be played with a duration of 30 seconds, 45 seconds, 1 minute and 2 minutes. Durations are implemented using a while loop. Start time is initiated at the beginning of the game and end is initiated at the beginning of the game using Python Time time method. End is updated before every answer check and start of new question session. If the end – start is greater than the duration. The game is considered over and the score counter is stopped

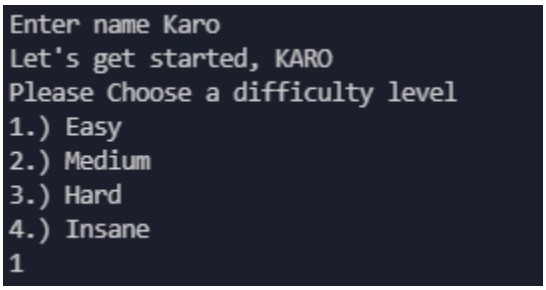
HIGHSCORES

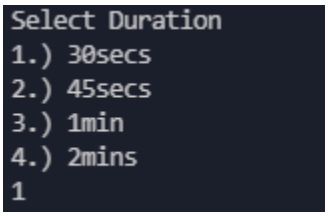
After each round the score is checked against the user high score, if the user score for the difficulty mode and duration is greater than the recorded high score for that mode and duration. The score is updated, and the user is notified that they broke their high score.

GAME WORKFLOW

- Call game function
- Create list of users (empty)
- Game asks for name
- Checks if name in list, if so, get user object else create new user object and append to user list
- Enter game loop
    - Create QuickMaths object, instantiating with user object (line A)
    - Call play game method in quick math object
        - Choose Difficulty Level
        - Select Duration
        - Question starts → If correct 1 is added to score
        - Question loop till duration ends

        - Check if score is greater than user high score and update if necessary for user
    - Enter Continuation loop
        - Try again as current user: continues the game loop with current user object
        - Try again as another user: creates new user object and appends to user list and continues the game loop
        - Show High score tables: Asks for difficulty and duration, sorts user list based on high score of selected difficulty and duration in descending order and loops through the list and print formats a high score table for the selected mode and duration
        - End Game (breaks game loop)





```
2 - 6 - 3
Solution: -1
Wrong answer Chief
3 - 4 - 3
Solution: -4
Correct
6 + 4 - 6
Solution: 4
Correct
4 + 5 + 3
Solution: 12
Correct
```

```
Your score is 9. KARO, you just broke your highscore for EASY mode (30 secs)
1.) Try again as current user
2.) Try again as new user
3.) Show Highscore Table
4.) End Game
```

```
Your score is 10. JOSH, you just broke your highscore for EASY mode (30 secs)
1.) Try again as current user
2.) Try again as new user
3.) Show Highscore Table
4.) End Game
3
Please Choose a difficulty level
1.) Easy
2.) Medium
3.) Hard
4.) Insane
1
```

```
Select Duration
1.) 30secs
2.) 45secs
3.) 1min
4.) 2mins
1
High Score Table for 30 secs
       Name     Score
 1.    JOSH       10
 2.    KARO       9
```