Number-Guessing Game

OVERVIEW

In this assignment, you are going to design and develop an interactive number-guessing game. On each game, your program will first randomly choose a 4-digit number and then repeatedly prompt the user for a guess until a correct number is entered (ie the last entered number matches the pre-chosen number), on a condition that a maximum of 8 guesses is allowed per game. On each guess the game will provide the user with some feedback as to how many digits are correctly matched (digit in correct position of the chosen number) and how many digits are in the chosen number but not in correct position.

SCOPE

Note: refer to the sample output for references.

1. Ensure the randomly chosen number is a 4-digit number, with no repeated digits
2. When prompting the user for the guess, show current number of guesses and guess limit
3. Verify user input as follows:
    a. The input must be a 4-digit number, no repeated digits
    b. If input is not 4 character long, prompt user to enter a 4-digit number again
    c. If input is 4 character long, but not a numeric value, prompt user to enter a 4-digit number.
    d. If input is 4 character long and is a numeric value but with repeated digits, prompt user to enter a 4-digit number.
4. Feedback
    a. On each guess, show 2 numbers as follows:
        i. Count of correct digits (digits from user input correctly match those from chosen number)
        ii. Count of common digits (digits from user input are found in chosen number but not in correct position)
5. Feedback History
    a. Keep all the guesses of the current game, and display all on each guess in chronological order
6. Termination
    a. Either the current guess matches the chosen number, or
        i. Display a congratulation message and the chosen number
    b. the number of guess attempts reaches the limit
        i. Display a game-over message and the chosen number

7. Game Startup
    a. Prompt user to start a new game or terminate the application with a single letter “y” or “n”, case-insensitive.

STARTUP OPTIONS

Like functions, you can pass arguments to your program. When launching a python program via command line, we can pass values as arguments to the python program. For instances:

- python your_program.py value1, value2, value3 …

Then in the python program, use sys.argv to access the argument words. For example:

- sys.argv[0] = ‘your_prgram.py’
- sys.argv[1] = ‘value1’
- sys.argv[2] = ‘value2’
- sys.argv[3] = ‘value3’

Use command-line argument to control whether to show the chosen number and to accept a pre-defined number as follows:

1. show

    when ‘show’ is passed as first argument to your program, show your randomly chosen number right in the beginning of the game right before the very first prompt for the guess number. Do the same for every game that follows. The default is to hide the chosen number from the user.

    - Python your_program.py show

2. fix NNNN (NNNN is 4-digit string number)

    when ‘fix’ is passed as first argument to your program, use the second argument (NNNN) as the pre-chosen number for the game, plus

    - show this number (like ‘show’ above)
    - use this given number for every game that follows

Remember to import module ‘sys’ and use len() to check if any arguments are passed to your program.

See sample code that follows.

```
def action():
    while True:
        mode = input('Game? (y/n) >')
        if mode.lower() == 'n':
            break
        if mode.lower() == 'y':
            play(g_show_number)

def is_valid_number(x):
    return x.isdigit() and len(set(x)) == MAX_NUM_OF_DIGIT

if __name__ == '__main__':

    # show chosen number
    if len(sys.argv) == 2:
        g_show_number = sys.argv[1] == 'show'

    # use given number
    if len(sys.argv) == 3 and sys.argv[1] == 'fix':
        if is_valid_number(sys.argv[2]):
            g_show_number = True
            g_use_this_number = sys.argv[2]

    action()
```

SKILLS

In this assignment, you will be trained on the use of the followings:

- Use input() to prompt user for information
- Use standard objects (strings, numbers & lists)
- Control statements to interact with users
- Variable Scope
- String formatting (method style)
- Functions for program structure and decomposition

DELIVERABLES

1. Design documentation (A1_School_StudentID_Design.doc)
2. Program source code (A1_School_StudentID_Source.py)
3. Output (A1_School_StudentID_Output.txt)

Zip all files above in a single file (A1_School_StudentID.zip) and submit the zip file by due date to the corresponding assignment folder under “Assignment (submission)”

DUE DATE

March 12th, 2017

TIPS & HINTS

- Use randint() from module “random” to generate the 4-digit number
- Format the feedback for each guess using format() (number guessed, correct : xx, position: xx)
- Use a list to keep all previous guesses
- Beware of variable scope as you might keep a few variables as global such as guess results and guess count.
- Refer to python website for program styles and naming convention (PEP 8)

SAMPLE OUTPUT – SUCCESSFUL GUESS

```
Your Guess 1/8 : 1234
1234 : Correct: 0 Position : 0

Your Guess 2/8 : 5678
1234 : Correct: 0 Position : 0
5678 : Correct: 1 Position : 2

Your Guess 3/8 : 0567
1234 : Correct: 0 Position : 0
5678 : Correct: 1 Position : 2
0567 : Correct: 0 Position : 4

Your Guess 4/8 : 5076
1234 : Correct: 0 Position : 0
5678 : Correct: 1 Position : 2
0567 : Correct: 0 Position : 4
5076 : Correct: 0 Position : 4

Your Guess 5/8 : 7650
1234 : Correct: 0 Position : 0
5678 : Correct: 1 Position : 2
0567 : Correct: 0 Position : 4
5076 : Correct: 0 Position : 4
7650 : Correct: 2 Position : 2
```

```
Your Guess 6/8 : 6750
1234 : Correct: 0 Position : 0
5678 : Correct: 1 Position : 2
0567 : Correct: 0 Position : 4
5076 : Correct: 0 Position : 4
7650 : Correct: 2 Position : 2
6750 : Correct: 0 Position : 4

Your Guess 7/8 : 7605
1234 : Correct: 0 Position : 0
5678 : Correct: 1 Position : 2
0567 : Correct: 0 Position : 4
5076 : Correct: 0 Position : 4
7650 : Correct: 2 Position : 2
6750 : Correct: 0 Position : 4
7605 : Correct: 4 Position : 0
Congratulations !! Number : 7605
Game? (y/n) >y
```

SAMPLE OUTPUT – USER INPUTS

```
Your Guess 1/8 : 123
Please enter a 4-digit number

Your Guess 1/8 : 23kj
Please enter a 4-digit number

Your Guess 1/8 : 1112
No repeated digits, try enter again

Your Guess 1/8 : abcd
Please enter a 4-digit number
```

SAMPLE OUTPUT – EXCEEDING GUESS LIMIT

```
Your Guess 8/8 : 1078
1234 : Correct: 0 Position : 1
5678 : Correct: 0 Position : 2
0178 : Correct: 0 Position : 2
9156 : Correct: 0 Position : 3
9256 : Correct: 0 Position : 2
9167 : Correct: 0 Position : 2
1978 : Correct: 1 Position : 2
1078 : Correct: 0 Position : 2
Game is over !! Number : 8915
Game? (y/n) >n
```

MARKING CRITERIA

Scope + Deliverables = 80%:

Program style & structure = 20%