Scope of Variables

except g_target and g_fix_num belongs to global variables, other variables in my code is local variables.

Goal

The user has to guess a 4-digit number with no repeated digit in this game. He/she will be given 8 chances to guess(an invalid input will not be counted). If the user can guess the right number within 8 times, then win. Otherwise, game over.

Structure

In my assignment 1, I define 5 functions as below:

```
def is_valid_number(number):…
def target_number():…
def check_correct(p_target,p_number):
def game(target):…
def startup(show): …
```

-> judges number valid or not
-> generates a random target number
-> give feedback to the user
-> the guess part
-> command start or end

After that is a main part:

```
if __name__ == '__main__':
    show = False
    if len(sys.argv) == 2 and sys.argv[1] == 'show':
        show = True
    if len(sys.argv) == 3 and sys.argv[1] == 'fix':
        if is_valid_number(sys.argv[2]) == False:
            print('Number {} is not valid'.format(sys.argv[2]))
            g_fix_num = target_number()
        else:
            show = True
            g_fix_num = sys.argv[2]
    startup(show)
```

Program Flow

I divide the condition into three. The program flow and the solution are followed by every condition individually. Furthermore, many kinds of bugs are inserted into this part.

The first condition: The user fixes a valid number as the target number.

1. Click ‘Run Python File in the Terminal’.
2. First enter command ’n’ to exit from the beginning.

```
bash-3.2$ python3 /Users/yu/Documents/cuhk/大一下/CSC1002/assignment/asm1/asm1.8.py
Game? (y/n) >n
bash-3.2$ python3 /Users/yu/Documents/cuhk/大一下/CSC1002/assignment/asm1/asm1.8.py fix 0123
0123
Game? (y/n) >y
Your Guess 1/8 :
```

3. The user inputs the Python file path as sys.argv[0] and ‘fix’ as sys.argv[1]. Also the target number is followed by the above two as sys.argv[2].
4. Show the fixed number to the user in a new line, if it is valid. If not, Python generates a new valid number automatically and randomly without showing to the user all the time and fixing the number when a game restarts.
5. Assume the user inputs a valid number. Like the above picture, ‘0123’ is the fixed number. Then the user input ‘y’ to start the game.

```
bash-3.2$ python3 /Users/yu/Documents/cuhk/大一下/CSC1002/assignment/asm1/asm1.8.py
Game? (y/n) >n
bash-3.2$ python3 /Users/yu/Documents/cuhk/大一下/CSC1002/assignment/asm1/asm1.8.py fix 0123
0123
Game? (y/n) >y
Your Guess 1/8 : 1234
1234 : Correct : 0 Position : 3

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

Your Guess 3/8 : 0123
1234 : Correct : 0 Position : 3
5678 : Correct : 0 Position : 0
0123 : Correct : 4 Position : 0
Congratulations !! Number : 0123
0123
Game? (y/n) >
```

6. The user attempts to guess the number several times. Every time after the user guesses, Python gives the feedback that how many digits are just the same value with the target number(the number behind the ‘Position’ reflects this) and how many are also in their correct position(the number behind the ‘Correct’ reflects this).
7. Once the guessed number is equal to the target number, then break from the guess loop and congratulate the user. Or another condition, the user used up his/her eight chances and game is over.
8. Because the user fixes the target number at the beginning, when a new game starts again, the target number immutable.(See the above picture)
9. Input the command ‘y’ or ’n’ to start or end the new game. If ‘y’, then the game starts like the above process. If ’n’, exit.

The second condition: The user wants to see the target number generated by Python.

1. Click ‘Run Python File in the Terminal’.

```
bash-3.2$ python3 /Users/yu/Documents/cuhk/大一下/CSC1002/assignment/asm1/asm1.8.py
Game? (y/n) >n
bash-3.2$ python3 /Users/yu/Documents/cuhk/大一下/CSC1002/assignment/asm1/asm1.8.py show
1630
Game? (y/n) >
```

2. First enter command 'n' to exit from the beginning.
3. The user inputs the Python file path as sys.argv[0] and 'show' as sys.argv[1].
4. Show the generated random target number to the user in a new line.(As the above picture, it is 1630.) Attention: 'show' indicates the user wants to see the target number but has nothing to do with whether the number is fixed.
5. Assume the user inputs a valid number. Like the above picture, '0123' is the fixed number. Then the user input "y" to start the game.(If "n",exit.)

```
bash-3.2$ python3 /Users/yu/Documents/cuhk/大一下/CSC1002/assignment/asm1/asm1.8.py
Game? (y/n) >n
bash-3.2$ python3 /Users/yu/Documents/cuhk/大一下/CSC1002/assignment/asm1/asm1.8.py show
1630
Game? (y/n) >y
Your Guess 1/8 : 0123
0123 : Correct : 0 Position : 3

Your Guess 2/8 : 2222
Please enter a 4-digit number with no repeated digits

Your Guess 2/8 : 2abc
Please enter a 4-digit number with no repeated digits

Your Guess 2/8 : 123
Please enter a 4-digit number with no repeated digits

Your Guess 2/8 : 1603
0123 : Correct : 0 Position : 3
1603 : Correct : 2 Position : 2

Your Guess 3/8 : 1630
0123 : Correct : 0 Position : 3
1603 : Correct : 2 Position : 2
1630 : Correct : 4 Position : 0
Congratulations !! Number : 1630
7235
Game? (y/n) >
```

6. The user attempts to guess the number several times. Every time after the user guesses, Python gives the feedback that how many digits are just the same value with the target number(the number behind the 'Position' reflects this) and how many are also in their correct position(the number behind the 'Correct' reflects this).
7. Attention: in the attempt, if the guessed input is invalid, then Python will ask the user to enter a 4-digit number with no repeated digits.(Until the valid umber is input, the count of guess will not be plus one every time.)

```
Your Guess 8/8 : 8901
1245 : Correct : 2 Position : 0
2345 : Correct : 1 Position : 2
3456 : Correct : 0 Position : 2
4567 : Correct : 0 Position : 2
5678 : Correct : 0 Position : 2
6789 : Correct : 0 Position : 1
7890 : Correct : 1 Position : 0
8901 : Correct : 0 Position : 0
Game is over !! Number : 7235
8760
Game? (y/n) >
```

8. If the count of guess time is 8, then the game is over because the user has used up all the chances. (See the left picture.)
9. Since sys.argv[2] is 'show', the target number will change once a new game begins(Like the left picture, it turned into 8760.).

The third condition: without command ‘show’ or ‘fix’

1. Click ‘Run Python File in the Terminal’.
2. Enter command ’y’ to start.
3. The user attempts to guess the number several times. Every time after the user guesses, Python gives the feedback that how many digits are just the same value with the target number(the number behind the ‘Position’ reflects this) and how many are also in their correct position(the number behind the ‘Correct’ reflects this).
4.give the below example to show the condition that the user inputs the right number within 8 times.

```
bash-3.2$ python3 /Users/yu/Documents/cuhk/大一下/CSC1002/assignment/asm1/asm1.8.py
Game? (y/n) >y
Your Guess 1/8 : 1234
1234 : Correct : 3 Position : 0

Your Guess 2/8 : 5690
1234 : Correct : 3 Position : 0
5690 : Correct : 0 Position : 1

Your Guess 3/8 : 1230
1234 : Correct : 3 Position : 0
5690 : Correct : 0 Position : 1
1230 : Correct : 2 Position : 1

Your Guess 4/8 : 1204
1234 : Correct : 3 Position : 0
5690 : Correct : 0 Position : 1
1230 : Correct : 2 Position : 1
1204 : Correct : 2 Position : 1

Your Guess 5/8 : 0234
1234 : Correct : 3 Position : 0
5690 : Correct : 0 Position : 1
1230 : Correct : 2 Position : 1
1204 : Correct : 2 Position : 1
0234 : Correct : 2 Position : 1

Your Guess 6/8 : 1034
1234 : Correct : 3 Position : 0
5690 : Correct : 0 Position : 1
1230 : Correct : 2 Position : 1
1204 : Correct : 2 Position : 1
0234 : Correct : 2 Position : 1
1034 : Correct : 4 Position : 0
Congratulations !! Number : 1034
Game? (y/n) >
```

5.give the below example to show the condition that the user has used up his/her 8 times of guessing, not including the right answer(the target number).

```
bash-3.2$ python3 /Users/yu/Documents/cuhk/大一下/CSC1002/assignment/asm1/asm1.8.py
Game? (y/n) >y
Your Guess 1/8 : 1234
1234 : Correct : 0 Position : 1

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

Your Guess 3/8 : 9012
1234 : Correct : 0 Position : 1
5678 : Correct : 1 Position : 1
9012 : Correct : 0 Position : 1

Your Guess 4/8 : 2367
1234 : Correct : 0 Position : 1
5678 : Correct : 1 Position : 1
9012 : Correct : 0 Position : 1
2367 : Correct : 2 Position : 0

Your Guess 5/8 : 1237
1234 : Correct : 0 Position : 1
5678 : Correct : 1 Position : 1
9012 : Correct : 0 Position : 1
2367 : Correct : 2 Position : 0
1237 : Correct : 1 Position : 1

Your Guess 6/8 : 3894
1234 : Correct : 0 Position : 1
5678 : Correct : 1 Position : 1
9012 : Correct : 0 Position : 1
2367 : Correct : 2 Position : 0
1237 : Correct : 1 Position : 1
3894 : Correct : 1 Position : 1

Your Guess 7/8 : 1842
1234 : Correct : 0 Position : 1
5678 : Correct : 1 Position : 1
9012 : Correct : 0 Position : 1
2367 : Correct : 2 Position : 0
1237 : Correct : 1 Position : 1
3894 : Correct : 1 Position : 1
1842 : Correct : 0 Position : 0

Your Guess 8/8 : 1956
1234 : Correct : 0 Position : 1
5678 : Correct : 1 Position : 1
9012 : Correct : 0 Position : 1
2367 : Correct : 2 Position : 0
1237 : Correct : 1 Position : 1
3894 : Correct : 1 Position : 1
1842 : Correct : 0 Position : 0
1956 : Correct : 0 Position : 2
Game is over !! Number : 5397
Game? (y/n) >
```