CSE 231
Spring 2009

Programming Project 4

This assignment is worth 30 points (3% of the course grade) and must be completed and turned in before midnight on Monday, February 9th.

Assignment Overview

In this assignment, we are going to implement a simple number guessing game that is often know as the game of Mastermind. This assignment will give you more experience on the use of:

1. Strings
2. User input
3. If statements
4. While statements

Task

We create a secret number (typically by prompting a "game administrator"). We then prompt the "game player" to guess the number. Our program gives the player some feedback on their guess. Based on that feedback, the player makes another guess. Guessing continues until the "secret" number is guessed or until the maximum number of tries is reached.

Our responsibility in this program is to "host" the game, that our program will provide interaction with the game administrator and the game player. Ideally, these would be different people; but for purposes of testing, you will be both the administrator and the player.

Program Specifications:

1) Prompt the administrator for a secret number. The number should be 5 digits long and no digit should be repeated in the number. 12345 is a good number, 55432 is not, nor is 444. If the administrator enters an ill-formed "number," print an error diagnostic and re-prompt for a correctly formed secret number. When a correctly formed number is entered, print some blank lines so the game player won't be able to see it.
2) Ask the player to provide a guess 5-digit number. If the player's input is not a 5-digit number (not all digits, not of length of 5), ask the player to provide another guess.
3) Given a 5 digit guess by the player, report two results.
   a) First, report how many of the digits in the "secret number" also appear in the guess, independent of whether the guess digits are in the correct positions.
   b) Second, we report how many of the guess digits are in the correct position.

   For example, imagine that the secret number is 12345 and that the player guesses 12567. Note that three digits in the secret number (the 1,3,5) are also in the guess. Moreover, two of the digits (1 and 2) are in their correct positions. Thus, your program should report the number of correct digits as 3 and the number in the correct positions as 2.
4) After each guess, report the player's progress (number of guesses, their guess, number of correct digits, number of digits in the correct positions). See the example.

5) At the end, if the player successfully guesses the number, print out how many guesses the player has used.
6) Set a maximum number of guesses (your choice, but report it when the game starts) for the game. If the player exceeds this maximum, report that the player lost.
7) During the process, allow the player to quit the game. If they quit, report that they lost and what the number of guesses was.

Deliverables

You must use Handin to turn in the file proj04.py – this is your source code solution; be sure to include your section, the date, the project number and comments describing your code. Please be sure to use the specified file name, and save a copy of your proj04.py file to your H: drive as a backup.

Tips/Hints:

1) Manipulation of the guess is best done as a string (don't convert it to a number)
2) Some useful string functions:
    a. `len(yourString):` how many characters in the string, returned as an integer
    b. `yourString[i]:` the character (string of length 1) at index I of yourString
    c. `yourString.isdigit():` are all the characters numbers. Returns a Boolean
    d. `testString in yourString:` is testString a member of yourString. Returns a Boolean. For example, `'123' in '0123456789'` returns `True`
    e. `yourString[s:f]:` the slice (subsequence) of length `f-s` starting at index `s`. It includes the characters from index `s` up to, but not including, the character at index `f`.
    f.
    ```
    for dig in guess:
         print dig
    ```
    Iterate through each digit in the guess. Do something with that digit (here print it)

Extra Credit (5 points)

Instead of prompting for the secret number, generate a random secret number. That number should consist of 5 digits, with no digit repeated. At the beginning of the game, give an option to have the number provided by an administrator or to have the computer generate it. To do this, look at the `random` package. In particular:

- `random.randint(start,end):` Generate a random number between start and end. For example, `random.randin(10,100)` generates an integer between 10 and 100 inclusively

Sample Program provided

A sample program is provided in the project directory. Copy the two files proj04.py and solution.pyc to your local directory, then open proj04.py and run.

Sample screen shots of the game in action are found in the Proj04ScreenShots directory.