In this week’s lab we will be building the games of guess the word and the mastermind.

1)In the game of guess the word, a word is given by the program that the player has to guess, letter by letter.

Program gives the number of letters as a clue.

The player guesses one letter at a time until the entire word has been guessed or the player guesses 6 letters incorrectly.

If the player guesses a letter they already guessed, notify the player and let the player guess a letter again.

An example interaction can look like this:

>>> Guess the Word!

\_ \_ \_ \_ \_ \_ \_ \_ \_

>>> Guess your letter: S

Incorrect!

5 Incorrect Guesses remaining.

>>> Guess your letter: E

E \_ \_ \_ \_ \_ \_ \_ E

To write the logic of our guess the word game, we need a few components:

* We need to load a word list and select a random word from it.
  + You can create a list of strings(words) and select one word from the list randomly
  + Word list is provided. Make sure it is in the same folder as your code.
* Get letters from the player and keep track of what letters are provided previously.
* Keep track of which letters in the word have already been guessed correctly and display the obstructed word.
* The game logic.

I do recommend implementing the components separately and combine them into the game later on.

Note1: For selecting word from a list of words you can use random.choice function. It will allow you to select a random element from a list. An example of using it will be:

import random

foo = ['a', 'b', 'c', 'd', 'e']

print(random.choice(foo))

Note2: Lists are your friends, and don’t forget the existence of the append function.

Note3: You can iterate through characters of a string using the following loop

for character in <string>:

#do something with the character

Note4: while True loop will run indefinitely unless the break is executed.

Following code will allow you to read the Word file into a list called WordList

WordList = []

with open('WordList.txt') as file:

for line in file:

WordList.append(line.strip())

2) Mastermind is a game of code breaking. Computer, the code maker, creates a code. The player, the code breaker, makes a series of guesses. After each guess, the code maker tell the code breaker if their guessed pegs are the right color and in the right place, are the right color but the wrong place, or are the wrong color entirely. The code breaker makes another guess in the next row, building upon information from previous guesses, trying to match the code the code maker created at the beginning of the game.

(https://en.wikipedia.org/wiki/Mastermind\_(board\_game))

Your game should utilize a list to create your code. The mastermind game contains 6 different colors for its code. Because our consoles are unable to display color without using more fancy functions, you need to use 6 letters to replace the color codes. You can use ‘A’,’B’,’C’,’D’,’E’,’F’ characters.

To write the logic of Mastermind game, we need a few components:

* We need to randomly create a 4 digit letter code.
  + You may need to utilize random.choice function.
* Get letters from the player.
* You need to give feedback on which letters are in the correct locations, which letters are in the code but not in the code locations and which letters are wrong.
  + You can use ‘1’(Correct Location),’2’(Correct Letter Wrong Location),’3’(Not in The Code) letters to notify the user.
* The game logic.

An example interaction can look like this:

>>> Welcome to MasterMind!

>>> Guess the Code: ABCD

1233

>>> Guess the Code: AEBF

1212

>>> Guess the Code: AFBE

You guessed it right!

Number Of Guesses: 3