**CPSC 230: Computer Science I**

**Fall 2020**

**Assignment:**

**Due**

**Overview**

You are a treasure hunter on the search for gold! You are inside an abandoned mansion with six rooms. Only one room contains the gold while the rest have a monster lurking within that will kill you as soon as you walk in. Luckily, you have the mastered the Pythonic ways and are equipped with the power of file I/O. The game is simple, make a choice of rooms 1 through 6, and either you get the gold, or you perish. But you also have 3 lives which means that you can restart the program and play again. With the power of files, you will also remember which room killed you last time, so you will be able to make a better decision. If you run out of lives or have already found the treasure, you cannot play the game again.

Now, leveraging file I/O, you will insert a line in a file, save.txt, whenever you make a choice. Suppose that room 1 has a ghost, room 2 has a wraith, and room 3 has the treasure. If you start the game for the very first time and select room 1, then an entry should be added to save.txt as **1,ghost**. So that next time you play the game and try to enter room 1 again, you get a prompt that says “*Remember last time? There was a* ***ghost*** *in room* ***1***”, and you get to pick another room. Now you could go to room 2 and die, but now another entry will be added to save.txt, so it looks like this:

**1,ghost**

**2,wraith**

So, if you play the game one more time, and try to enter either room 1 or 2, you will be told what happened the last time you tried to enter this room. If you pick 3, the entry added to the file will be

**3,treasure**.

If you ever run the program and there are 3 entries already in save.txt, you’ve lost all lives and the game should just say “*You can’t play anymore, Game Over!*” and end without letting you enter a room choice. Similarly, if the entries include **treasure**, then you can’t play again as well.

For ease, you will need the following 3 functions. Feel free to add more:

1. **write\_file( )** This function takes 3 parameters, a file name, room number, and the name of the monster. The function will write the room number and monster name, comma separated, to the file with the name provided as parameter.
2. **read\_file( )** This function takes a file name as parameter and returns the content of the file with that name as a list of strings (think *readlines* method). Make sure that you have exception handling in place if the file does not exist.
3. **build\_dictionary( )** This function takes a list of strings as parameter (presumably the content returned from **read\_file** function) and builds a dictionary from it.

All 3 functions are related as the content written by **write\_file** function is being read in by the **read\_file** function, and then being used by the **build\_dictionary** function. You will build your dictionary by making the content on the left of the comma to be the key, and the content on the right to be the value. This should result in a dictionary that contains integer room numbers as keys and monster names as values.

In your program, you should first load in the save file, if it exists, and generate a dictionary from its contents. Next, you will check if the number of pairs in the dictionary are greater than 3. You will also check if “treasure” is one of the values in your dictionary. If either of these cases are true, then you don’t need to present the player with a choice. Instead, just end the program. Otherwise, you should set up a scene and present the player with a choice of a room to select. If the player selects a room with a monster in it, describe in detail their gruesome death, and add the room number and the monster’s name in the save file. Also, if the current room selection is already an entry in the dictionary, then the player needs to be cautioned and should be allowed to pick another room. If the player has selected the room with the treasure, tell them about their fortune and again, add the room number and “treasure” to the save file.

The layout of the program is left open ended so you can be as creative as you want in your game design. Furthermore, you can add more features to the game, as long as the 3 lives + treasure mechanism is still in place.

**Due Date**

This assignment is due by midnight on . Submit via Canvas; create a compressed (zip) folder with all your files in it. It should be named FirstintialLastname\_Assignment1 (i.e. RAli\_Assignment1). Please make sure to include all required files (README, source files).

**Grading**

* 1. Assignments will be graded on correctness, adherence to style, and the inclusion of meaningful comments.