Quan Yuan
Oct 14, 2022
Project 5 Report

SUMMARY
This project is to realize the spaceman game using python.It can be divided into several small tasks.

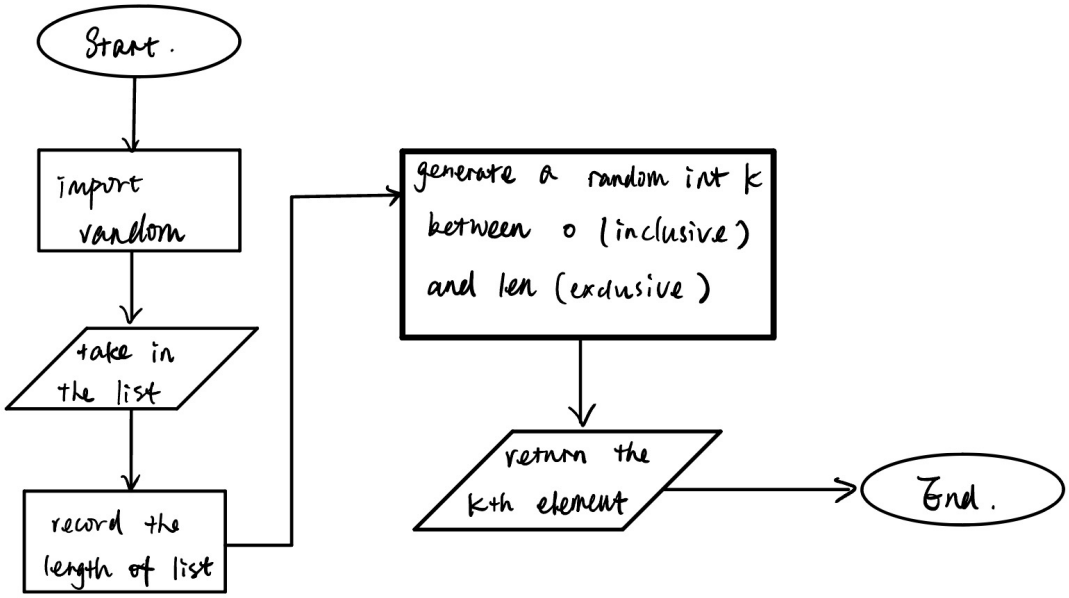
1. choosing a random word from a given list
2. storing/displaying a hidden word
3. finding a character and setting that space in the hidden word
4. telling the word has been found
5. terminating when user made 6 wrong guesses
6. main() function

The following parts are organized by above tasks, each containing a plan about how to realize it.

TASK 1:
The creation of the word list is on the top of the program, which will be a global variable and easy to notice and modify. The name of the list is DICT for dictionary.

Then the function of random pick a word form DICT is to take DICT as an input, generate a random number k form 0 to length-1, both inclusive. Then return the k th element in DICT.

The flowchart is as follows:



TASK2:
When we get the random word, we want to display it as hidden word.

So we create a list called temp, and make it have the same elements as word's characters. Then we make all elements "_". So when we print temp, we only know how many "_" are there, but don't know what letter each "_" represents.

Another important point is that if we directly print a list, it will be like "["_", "_", …]". This format is not what we want. We want to print only "_ _ _…". So I developed a toSting function that takes a list as input. Then combine all elements into 1 string, then return. In this way, we can get the desired format.

TASK3:
Each time user guess a letter correct, we need to reveal all the corresponding positions.

So after we checked the letter is in the random word, we could for loop the word(string) to see which positions are the same as the input letter. Then we change the corresponding positions in temp(list) to the input letter. And print toString(temp).

TASK4:
When the user get all characters, we have to promptly detect this status, and tell the user win, instead of continuously ask the user to guess.

As designed before, temp will contain "_" if the character is not guessed, and reveal the letter if it is guessed. For example, the random word is "apple", then temp initially will be ["_", "_", "_", "_", "_"]. After the user guessed "p", temp will be ["_", "p", "p", "_", "_"].

So we could notice that when the user get all letters correct, every element in temp will be the corresponding character in the random word. And "_" will never exists.

Hence, we could check if "_" is still in temp to determine whether the user get all characters.

TASK5:
We don't want the user have unlimited guesses as they will eventually get the word. In this program, we want to stop and tell the user lose if they have 6 wrong guesses.

So we initial a variable called wrong_attempt as 0. Each time the user input a letter, we check if the letter is in the random word. If not, we increment the wrong_attempt by 1. Then check if the wrong_attempt has reached 6. If yes, we stop and tell the user they lose.

There are 2 things have to pay attention to. One is that we don't want a magic number appears in the program, or it will be difficult to understand and maintain. Hence, we could initial a global variable called WRONG_ATTEMPT at the top. And set it to be 6. Then we replace all 6 in the program by WRONG_ATTEMPT. So when in later stages, we want to change it to 5, or 7, we could simply change the WRONG_ATTEMPT at the top.

The other thing is that what if the user guessed a letter they have already guessed? My way to check is to see if the letter is in list temp. Because when the first time the user get it right, we will reveal all corresponding positions and change them in temp. So if the letter is in temp, we know that the user has guess it before. In this case, we also increment wrong_attempt by 1, and check if it reached WRING_ATTEMPT.

TASK6:
I create a function for task 1, called random_word(), and combined task 2-5 in 1 function called spacemen(). Random_word() will take DICT as input and return a random word from the list.

Then spaceman() will take the random word as input, then ask the user to guess.

In spaceman(), we create a list temp first like in TASK 2. Then initialize wrong_attempt to be 0. Then we enter a while loop to keep asking the client for guessing letter. The breaking condition of the while loop is either wrong_attempt reach WRONG_ATTEMPT, OR "_" is not in temp anymore. So the while loop will be:

while (wrong_attemp<WRONG_ATTEMPT) & ("_" in temp)

Hence, either condition not fulfilled will break the while loop.

Then in the while loop, we ask for the guessing letter first. Then nest a if…elif…else… statement to reflect the situations in task 3, 4 and 5. And keep updating temp if guessed correct.

After the while loop, we need to check why the while loop breaks, because we have 2 conditions. If it is because wrong_attempt reach WRONG_ATTEMPT, we print "lose". If it is because "_" not in temp, we print "win".

This is how spaceman() is organized.

Then in main() function, we only call random_word() to get the word, and spaceman(word) to play the game.

It is easy and clean in main() for others to understand the flow.