In this second part, we are just going to update our code a little bit in order to have a bit more functionality. First of all, we want to be able to update the user on his progress and the words remaining. To do this, we are going to tell the user what the word is but with dashes where they have not yet guessed for example “W O\_ D”.

To do this, we are going to create a list that will contain the letters in the <code> used\_letters </code> if the letter is in the <code>word</code>

#tell the user what the word is but with dashes where they haven't guessed eg W O \_ D

        word\_list = [letter if letter in used\_letters else "\_" for letter in word]

        print("Current word: ", " ".join(word\_list))

word\_list = []

for letter in word:

if letter in used\_letters:

word\_list.append(letter)

else:

word\_list.append(“\_”)

<code>” “.join(word\_list)</code> here is taking all the items in an iterable (in this case our <code>word\_list</code>) and then joining them into a string. <br />

These lines of code are placed within the while loop so every time that the loop runs, the player would be able to know how many spaces are left and it would help them in their guessing. <br />

The final thing we need to do is to implement a <code>lives</code> variable so that the game does not go on forever. I am choosing to have a game live of 6 meaning the player has six (6) lives or chances to guess. If the player guesses correctly, the player still has six lives but else, the player loses a life. <br />

We only remove a life if the player is wrong so where we are telling the player that their letter is not in the word, we are going to remove the life at that point. Also if, our player’s life is zero (0), that means they cannot play again so we are also going to update our while statement to include <code> while len(word\_letters) > 0 and lives > 0: </code>. We also want to tell the player how many lives they have left so we are going to add a print statement that shows them that on each occurrence of the while loop. If the <code>lives</code> variable is equal to zero i.e. <code>lives == 0</code> that means the player didn’t guess the word correctly and is out of lives, we send them a print statement telling them they failed and we also let the m know what the word was. After implementing these two things, this is what the final code looks like.

import random

from words import words

import string

def get\_valid\_word(words):

    word = random.choice(words) #randomly choose a word from the list

    while "-" in word or " " in word:

        word = random.choice(words)

    return word.upper()

def hangman():

    word = get\_valid\_word(words)

    word\_letters = set(word) #letters in the word

    alphabet = set(string.ascii\_uppercase)

    used\_letters = set() #what the user guessed.

    lives = 6

    #getting user input

    while len(word\_letters) > 0 and lives > 0:

        #tell the user the letters already used

        print(f"You have {lives} lives left and you have used these letters: ", " ".join(used\_letters))

        #tell the user what the word is but with dashes where they haven't guessed eg W O \_ D

        word\_list = [letter if letter in used\_letters else "\_" for letter in word]

        print("Current word: ", " ".join(word\_list))

        user\_letter = input("Guess a letter: ").upper()

        if user\_letter in alphabet - used\_letters: #alphabet - used\_letters would give you a set of the letters in the alphabet not yet used.

            used\_letters.add(user\_letter)

            if user\_letter in word\_letters:

                word\_letters.remove(user\_letter)

            else:

                lives = lives - 1 #takes away a life if wrong

                print(f"Your letter {user\_letter} is not in the word.")

        elif user\_letter in used\_letters:

            print("You have already used this letter before. Please try again.")

        else:

            print("Invalid character. Please try again.")

    if lives == 0:

        print(f"Sorry, you died. The word was {word}")

    else:

        print(f"You guessed the word {word} !!!")

hangman()

Below are two (2) examples of when the code is run. The first one is a scenario where we successfully guess the word and in the second one we fail. <br />

<img src=”/images/hangman\_lives.JPG”><br />

<img src=”/images/hangman\_dies.JPG”><br />

<br />

Cheers 🥂