DAY2 TP

**Subject of the TP**: Make the game hangman algorithm. A player must guessing, letter after letter, a secret word. At the beginning, the unknown word is displayed hidden under stars: "\*\*\*\*\*\*\*\*\*\*". As soon as a letter belonging to the word is found by the player, one or more stars are replaced by this letter: "\* e \*\*\*\*\* e \*\*". And so on until you discover the word in full. Level of difficulty: it is necessary to give a limited number of tests at the beginning. The attempts number decreases with each letter given. The game is won if the word is found in less or as much imposed attempts. If you have time, you can then play the hangman game in python console.

def get\_user\_name():

# The function is in charge of recovering the user name.

# The user name must contain minimum 3 characters, numbers and letters only. #

# If this name is invalid, we call recursively the function to obtain new name. #

    user\_name = input("type your name")

    # We make the first letter in Uppercase and others in lowercase. #

    user\_name = user\_name\_capitalize()

    if not user\_name.isanum() or len(user\_name)<4:

        print("This name is invalid.")

        # We call the function again to have another name.

        return get\_user\_name()

    else:

        return user\_name

def get\_letter():

    # This function get the typed letter by user.

    # If the get chain is not a letter, then we call recursively the function until we get a letter.

    letter = input("Type one letter")

    letter = letter.lower()

    if len(letter)>1 or not letter.isalpha():

        print("The letter you typed in is invalid.")

        return get\_letter()

    else:

        return letter

def get\_secret\_word(complete\_word, found\_letters):

    # This function sends back the secret word, partially found or not

    # We send back the original word with \* replacing the letters we stil haven't been found.

    secret\_word = ""

    for letter in complete\_word:

        if letter in found\_letters:

            secret\_word += letter

        else:

            secret\_word += "\*"

    return secret\_word

    def run():

    # Number of attempts per game

    nb\_try = 8

    # Word needs to be guessed

    word\_to\_guess = "alphabet"

    # Variable to stock the found letters

    found\_letters = []

    # We get a user name

    user = get\_user\_name

    print(" Welcome {0} Let's play".format(user))

    #initialization game

    word\_found = get\_secret\_word(word\_to\_guess,found\_letters)

    nb\_chances = nb\_try

    #Ongoing game

    while word\_to\_guess!=word\_found and nb\_chances>0:

        print(" Word to guess {0} (still {1} chances)".format(word\_found,nb\_chances))

        # The letter has already been choosen

        if letter in found\_letters:

            print("You have already used this letter.")

            # This letter is already existed in the word to guess

            elif letter in word\_to\_guess:

                print("Well guess!")

            # The letter does exist in the word to find

            else:

                nb\_chances -= 1

                print("... no,this letter doesn't exsit in this word.")

            found\_letters.append(letter)

            word\_found = get\_secret\_word(word\_to\_guess,found\_letters)

            # Did we find the word or our chances are they exhausted ?

            if word\_to\_guess == word\_found:

                print("Congratulations! You found the word {0}.".format(word\_to\_guess))

            else:

                print("Game over ! Sorry, you lose.")

run()