EFREI L1 Int.

Techniques & Tools of Programming
Project 1 (V_0): Hangman

Contents

1 Objective **1**

2 Game Description **2**

I Play with Strings **2**

3 List of Words **3**

4 Text Files **3**

II Hangman: Without GUI **3**

5 Initial configuration **4**

6 During the Game **4**

7 Failure Case **5**

8 Victory Case **6**

9 Hangwoman **7**

III Hangman: With GUI (optional) **7**

10 Project planning **9**

10.1 Dates 9

10.2 For the project defense 9

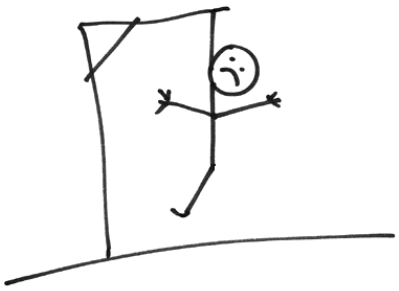
10.3 For the report 9

1 Objective

The goal of this project is to design and implement the **Hangman** game. It is divided into three parts: In the first, you will learn how to manipulate `strings` through some simple exercises. The second part is dedicated to the **Hangman** game itself, and the last one (optional) is to create Graphical User Interface (GUI) for the game.

2 Game Description

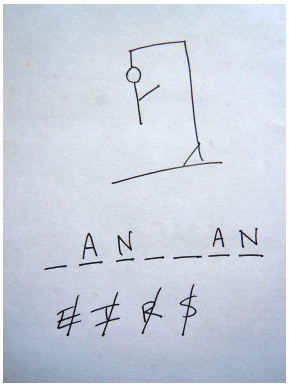
Hangman is a paper and pencil guessing game for two or more players. One player thinks of a word, phrase or sentence and the other tries to guess it by suggesting letters or numbers, within a certain number of guesses. In this project we will focus on words only.



The word to guess is represented by a row of dashes, representing each letter of the word. In most variants, proper nouns, such as names, places, and brands, are not allowed. Slang words, sometimes referred to as informal or shortened words, are also not allowed. If the guessing player suggests a letter which occurs in the word, the other player writes it in all its correct positions. If the suggested letter or number does not occur in the word, the other player draws one element of a hanged man stick figure as a tally mark.

The player guessing the word may, at any time, attempt to guess the whole word. If the word is correct, the game is over and the guesser wins. Otherwise, the other player may choose to penalize the guesser by adding an element to the diagram. On the other hand, if the other player makes enough incorrect guesses to allow his opponent to complete the diagram, the game is also over, this time with the guesser losing. However, the guesser can also win by guessing all the letters or numbers that appears in the word, thereby completing the word, before the diagram is completed.

The following Figure illustrates an example of Hangman game in progress. The underlined letters appear in the word in their correct places, while the crossed-out letters do not appear, and each crossed-out letter corresponds to one part of the drawing. In this case, the secret word is "hangman".



Part I
Play with Strings

3 List of Words

1. Given a list of words, write a program to:
   (a) display the words of the list where the first and the last characters have been removed,
   (b) find the longest word of the list
   (c) find the bigger word of the list following the alphabetical order
   (d) create a list of integers representing the length of the corresponding word in the list,
   (e) display the number of items in the list followed by the elements of the list,
   (f) removes all the occurrences of a given word,
   (g) replaces a word `$w_1$` by a word `$w_2$`,
   (h) reads a word from the user and display the number of occurrences of this word and their positions

4 Text Files

1. Write a program that opens a text file, containing words, and
   (a) displays the number of words and the number of characters in the file,
   (b) displays all the words in the alphabetical order, each followed by its number of occurrences in the file.
   (c) convert the letters of the file to all uppercase.
   (d) get a word w_1 from the user and displays all the words w_2 in the file having w_1 as a sub-word.
   (e) randomly choses a word from the file.
   (f) from a given word w_1, get a letter l from the user and search it in the word. If the letter exists, the program then displays a word w_2 of the same length as w_1 where $w_2[i] = \_$ if $w_2[i] \neq l$ and $w_2[i] = l$ if $w_1[i] = l$.

Part II
Hangman: Without GUI

In this section we display the current status of the game as follows: A figure representing the hangman is progressively displayed depending on the number of correct guessed letters. At each iteration, the user is informed with the number of wrong attempts left and the previously guessed letters. If the user gives an already guessed letter, he/she is informed and the attempt is not counted.

5 Initial configuration

- Initially:

```
Letters you have guessed:

you have the right to make up to 6 errors

    ______
    |    |
         |
         |
         |
         |
         |
         |
      ___|___


__ __ __ __ __ __ __
Guess a letter:
```

6 During the Game

- During the game:

```
Letters you have guessed:
t z h r g b a c
you have the right to make up to 1 errors

    ______
    |    |
    ( )  |
   / | \  |
  /  |  \ |
    /    |
   /     |
      ___|___


h a __ g __ a __
Guess a letter:
```

- In case the user give an already guessed letter:

```
Letters you have guessed:
t z h r g b a c
you have the right to make up to 1 errors

    ______
    |    |
   ( )   |
  / | \  |
 /  |  \ |
   /     |
  /      |
      ___|___


h a __ g __ a __
Guess a letter: a
You've already guessed that letter. Try again.

Letters you have guessed:
t z h r g b a c
you have the right to make up to 1 errors

    ______
    |    |
   ( )   |
  / | \  |
 /  |  \ |
   /     |
  /      |
      ___|___


h a __ g __ a __
Guess a letter:
```

7 Failure Case

- in case of failure: the random word is displayed.

```
Guess a letter: l

Letters you have guessed:
t z h r g b l a c
you have the right to make up to 0 errors

    ______
    |    |
   ( )   |
  / | \  |
 /  |  \ |
   / \   |
  /   \  |
      ___|___


h a __ g __ a __
You lose! Booo
Your word was hangman
```

8 Victory Case

- in case of victory

```
Guess a letter: n

Letters you have guessed:
a g h m n
you have the right to make up to 6 errors

    ______
    |    |
         |
         |
         |
         |
         |
      ___|___


h a n g m a n
Yay! You win!
>>>
```

9 Hangwoman

1. For the sake of fairness, modify your program so that the user choose between **hangman** or **hangwoman**. In the latter case, the displayed figure is the following:

```
Letters you have guessed:
4 r z t y e
you have the right to make up to 0 errors


    ______
    |    |
   { }   |
  / | \  |
 /  |  \ |
   / \   |
  /___\  |
   | |   |
      ___|___


__ __ __ __ __ __ __
You lose! Booo
Your word was hangman
>>>
```

2. Finally, change your program so that, at the end of the game, the user is asked whether he/she wants to play again. If yes, the game is re-launched.

Part III
Hangman: With GUI (optional)

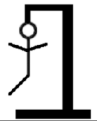
In the final part of the project, you are expected to write a Python program having the same requirements as the previous part, but with a GUI. For this purpose you can use the **tkinter** module of Python. In this part, you have to be as much autonomous as possible since the GUI is optional. A possible GUI could be as follows, at the beginning of the game, during the game, in case of failure and in case of victory, respectively. You can find help about the `tkinter` module on https://www.tutorialspoint.com/python3/python_gui_programming.htm. If you want to use the following images, they are available on `https://lipn.univ-paris13.fr/~klai/HangmanFig/`.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Recommencer Quitter

_ _ _ _ _ _

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Recommencer Quitter

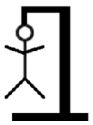
- O O



A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Recommencer Quitter

RECALL

GAME
OVER



A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Recommencer Quitter

HANGMAN

YOU
WIN!

10 Project planning

10.1 Dates

1. Publication of project subject: October 23, 2017
2. Project follow-up session: October 26, 2017
3. Project defense: November 16, 2017
4. Submission of code sources and report (on Campus EFREI): November 16, 2017 at midnight.

10.2 For the project defense

You will have to present the game program. The teacher will ask for a demonstration of the program, in which all the points of the specifications will be verified. You **must** be able to explain your source code in detail. **Any suspicion of cheating or plagiarism will result, at least, in a mark of zero**.

10.3 For the report

The report must contain: an introduction, a simple presentation of the project, the different algorithms, an explanation of the main difficulties encountered, an analysis of the project and a conclusion.

Have fun