

Python Project 1

HANGMAN GAME

Part 1 Play with strings

List of words

The first exercises are about manipulating strings and array with Python functions. We have chosen to do a function for each question.

Text Files

With those exercises we learned to open file, and to use files in Python. At first we thought that we needed to use `my_file = open("test.txt", "r")` but with this method it was possible that the program end before closing the file. To avoid closing problem we used the `with` statement as for example `with open("test.txt", "r") as my_file:`

Features

- As I'm working with Linux I made a makefile to automatically remove the `__pycache__` folder.
- In every function we implemented docstrings. Docstrings allow the user to type `help(function_name())` into the python shell to display more information about the function.

Part 2 Terminal Hangman

Project presentation

We had for project to code the Hangman game in Python. The hangman game is basically a string comparison between the secret word and the word inputted by the user.

To test our project faster, and also because we found it funnier with this feature, we implemented the two players mode. This mode is simply a mode in which another user input a word to make it guess to the other. We used this mode to debug as at the beginning our word list wasn't full.

We have decided to add some difficulties by trying to have a "failure proof" program. This means that we wanted our program not to crash with any input from the user.

Project organization

We wanted our project to be very well organized. To do so we split our functions into different files and we imported it in a main file: `hangman.py`. The `draw.py` file was used for every function used to display the hangman drawing. The `game.py` file was used for the game functions. As the one to input word or choose a word randomly in a file. And the `security.py` file is used for every security related functions.

We also open a private GitHub folder to share our code that we will put public after the rendering date. <https://github.com/Mr-Monster-0248/Hangman>. Please take a look at our project page where we even created a README for the future user of our game.

The algorithm

At first, we made our main loop. The one which allows the user to restart the game at the end. Then we have the title screen in which the user is invited to choose his game mode (single player or multiplayer). If the player input isn't the one waited by the program, the program will ask again, without crashing. Then the player needs to choose how to be portrayed (a man or a woman).

After this initialization, if the player had chosen 1 player mode he directly enters the main loop, else a 2nd player will have to input a word (without number or spaces) before entering the game loop.

The main game loop begins by displaying all the used letters. By displaying the list in which they are stored after each guess. Then we display the Hangman according to the number of mistakes. And then we ask the user to input his letter. Finally, we just need to repeat this loop until the word is found or until the user makes more than 6 errors.

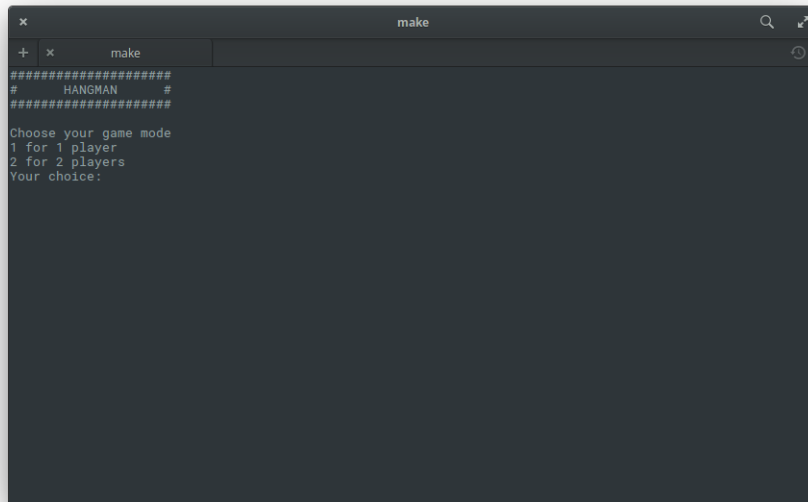
Security and features

We used the try except statements to catch errors and avoid crashing the program.

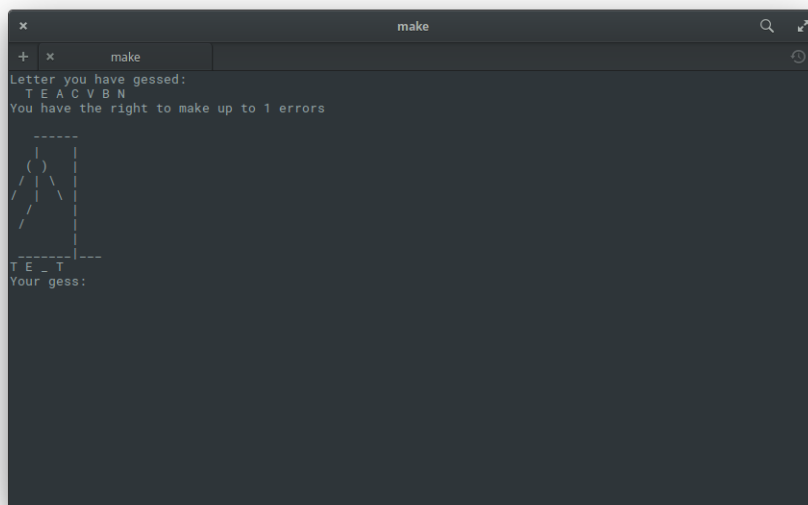
Every input is checked before returning also to improve security.

As one of us is using Linux and the other Windows. We recognize the OS to use the suitable command to clear the screen.

Screenshot of the game



Title screen and Main menu



Example of gameplay

Part 3 Hangman with GUI

Start with tkinter

We started to learn the tkinter library but we had not enough time to make something presentable for the project delivery. But in our GitHub page you will be able to see our first try and our tests with tkinter

Conclusion

This project was a very interesting way to learn how to use strings and file in python. We knew already some C and the comparisons between the languages were interesting. The strings handling is much more complex in C than in python, due to the fact that the memory in python is handled directly by the computer.

Now even if the project is finished we may continue the GUI part so feel free to come back to our GitHub project page.