REPORT programming project

MOUROT Adrien-HOUYOUX Sarah-PAVLOV André

2023-2024

## A screenshot of a computer Description automatically generatedINTRODUCTION

At the end of the semester, we had to do a programming project, basically we had to code a game, the ‘Guess Who?’ and the interface to sign up and log in to the game, we should then be able to pick the character we want to, chose a difficulty and the appropriate language, and finally play a game with someone. Our objective is to make the game as pleasant as possible, and to code a good game.

## LIBRARIES

In this project, we needed to use a lot of libraries. We first needed to use *pygame* because it was in the aim of the project, learning how to use *pygame* to create a game platform to provide several games to users.

A screen shot of a computer program

Description automatically generatedThen 9 other libraries were necessary to achieve the goal we had in this project. Some libraries were used only to make the code easier to handle by using simpler functions thanks to those libraries.

The others were used because they were necessary for the smooth running of the code and the achievements we needed to have to complete the game interface and the games correctly.

This is the list of the different libraries used in this code and their utilization in it:

* The *pygame\_menu* library is a set of Python modules that provides a simple and easy-to-use interface for creating menus in *pygame* games, in this code the library is used to create the main menu and the ‘how to play’ game state.
* The *pygame\_textinput* library allows to display texts on pygame’s windows. These texts are not buttons or selectors, which are allowed by *pygame,* but only texts to display information.
* The *typing* module used to import the “*Tuple*” and “*Any*”; *Tuple* is used to specify the expected type of a function parameter “*selected\_value”*, and *Any* is used to specify that another function parameter “*difficulty\_value*” can be of any type.

A black screen with white text

Description automatically generated

* The *tkinter* library is a standard GUI toolkit for Python, which provides a powerful object-oriented interface to the Tk GUI toolkit. It uses *pygame* and *pygame\_textinput* libraries for creating a graphical user interface.

A screen shot of a computer program

Description automatically generated

* A computer code on a black background

  Description automatically generatedThe *sys* module is used in the program to handle command line arguments, exit the program gracefully, or interact with the Python runtime environment in other ways. In this example, the *sys* module is used to exit the program when the user closes the window. When the “*pygame.QUIT*” event is detected, “*pygame.quit()*” is called to uninitialized all *Pygame* modules, and then “*sys.exit()*” is called to exit the program.
* The *subprocess* library here is used to run a new Python script, ‘*guesswho.py’*, from the current script. The “*Popen* “function allows to open other windows when we want the last to be open at the same time.



* The *os* library allows to find paths and list the documents and files in the path.

A screen shot of a computer program

Description automatically generated

* We use the *csv* library to write and read into csv files.

A black background with white text

Description automatically generated

* The *random* library allows you to take random images for the guess who game.

## WHAT WE LEARN

First, it was the first time for us coding a game, so it was fun at first. We learned a lot about libraries, mainly *pygame*, and about the numerous uses of this library. We also learned how to upload images, how to interact with them, how to create a menu with different buttons and functionalities, such as changing the language, difficulty, ...

It was very interesting to see what we can do with python, and playing its own game after coding it is also really fun.

The line *“#type: ignore*” allows to delete the display of errors if the errors is not a problem for running.

## CHALLENGES

We faced a few problems/challenges during the

* One of the challenges was the time that we took to translate the whole game, it was quite long to find all the words to translate, and the fact that we had to isolate the functions into the written sentences because it caused an error with the translation. For example, concerning the rules of the game we say “Hello (name of the player)”, then we had to isolate the function that displayed the name of the player for the translation to work well.
* The delete user function was also quite complicated, because we had to find and delete the corresponding line in the *file\_username*, which was not the easiest to do.
* Also, when coding for the choice of the character by importing the image, we had problems finding how to move the hitbox on which we must click to import the image.

## THE FUTUR UPDATES

This project provided us with a great opportunity to use the *pygame* library. Although the code is not as optimized as we intended, we should thoroughly review and revise it for better efficiency, redefine certain definitions, and eliminate the unnecessary use of certain libraries. There are various aspects that could have been optimized.

Beyond just improving the code's efficiency, this project serves as a solid foundation for a game platform. Building upon what we have created, we can add more games, functionalities, and options.

Both the game itself and the platform can be updated to achieve a specific visual style. We can add or change colors, shapes, and music to create an entirely new platform. In essence, this project serves as a starting point for creating the game platform we envision.

Finally, as we all know, games are best enjoyed with other people. Adding a component with an HTML script that enables interaction and turns it into an online gaming platform for multiplayer use would be a great addition. This update would be a significant achievement, providing a tangible product to play and share.