GRAND PROJECT

The objective

This project aims to create is to create a game platform to provide several games to users by using the **pygame** library.

The overview of the Project

Session (1/5) – Understanding problem, Coding.
Session (2/5) – Coding
Session (3/5) – Coding
Session (4/5) – Coding
Session (5/5) – **demonstration and evaluation**.

Preparation

1) Download Visual Studio Code:
https://code.visualstudio.com/

2) In Visual Studio:
   - In the left side menu, click extensions and search for Python.
   - Add the Python extension.
   - In the left side menu, click ***Explorer*** and open folder to open your project folder.

3) Installation steps:
1- python --version
2- pip --version
3- pip install pygame
   alternative option - python3 -m pip install pygame
   alternative option2 - pip install pygame –pre

The Project submission

You will work in groups of 3 students during the project. The maximum mark is 20 points.

The Project presentation will take place in **the last session** (5/5). You don't need to prepare any PowerPoint slides. You will run your codes to show the implemented platform.

The Project submission will be at the end of the session (4/5).
- Submission file: In a zip file with the **group name via email.**

Teacher's email address for other questions: aybuke.ozturk@ipsa.fr

THE PROJECT STEPS

Step 1: Run ***1_Example-MainMenu.py*** file to understand how to create a main frame.

This practice lets you learn how to create a main frame, how to use button and other pygame menu functionalities and lets you practice reading from a csv file and writing to a csv file.

1) Main sign up and log in frames (0.5 points)

Create a main frame:

- Change the title **"Welcome to Game Platform!"**
- Add four entries and two buttons as presented in Figure 1.
- Update the frame's left up corner name as **"IPSA GAME PLATFORM".**
- Finally, change the color of the frame blue to another color. (To start implementation, please use the given ***1_Example-MainMenu.py*** file.)

Figure 1: Main frame

2) SIGN UP (1.5 points)

Write a ***`Sign-up`*** function that uses an email and a password.

- If the user exists, display on the frame **"Already signed up email!"** as presented in Figure 2.
- If the user doesn't exist, update the ***`File_username.csv`*** file with the new user email and password and then display on the frame **"New user is registered!"** as shown in Figure 3.
- Add standard to password by putting limits between 6 to 30 characters. If the user doesn't enter characters between limits, display on the frame **"The password is too short (min 6 characters)."** as presented in Figure 4.

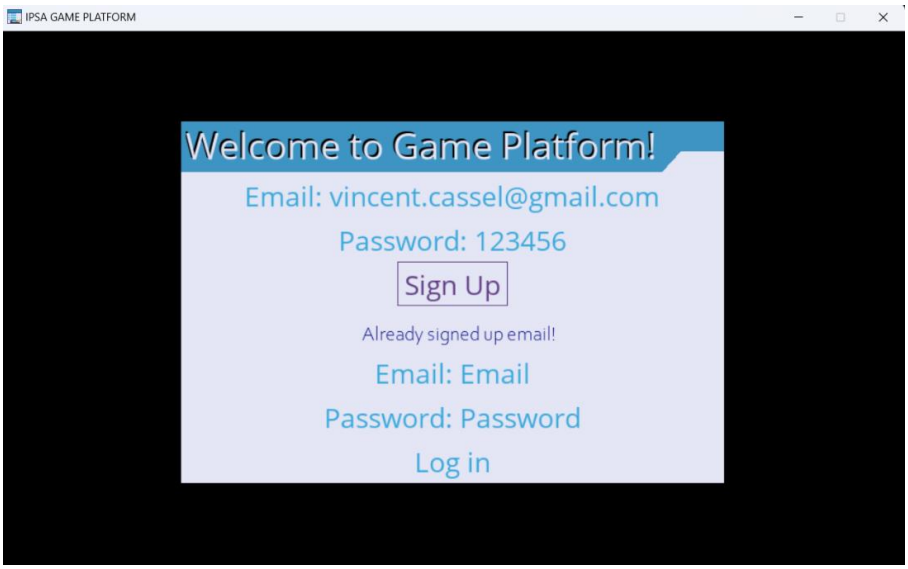

Figure 2: Sign up – Already signed up email.

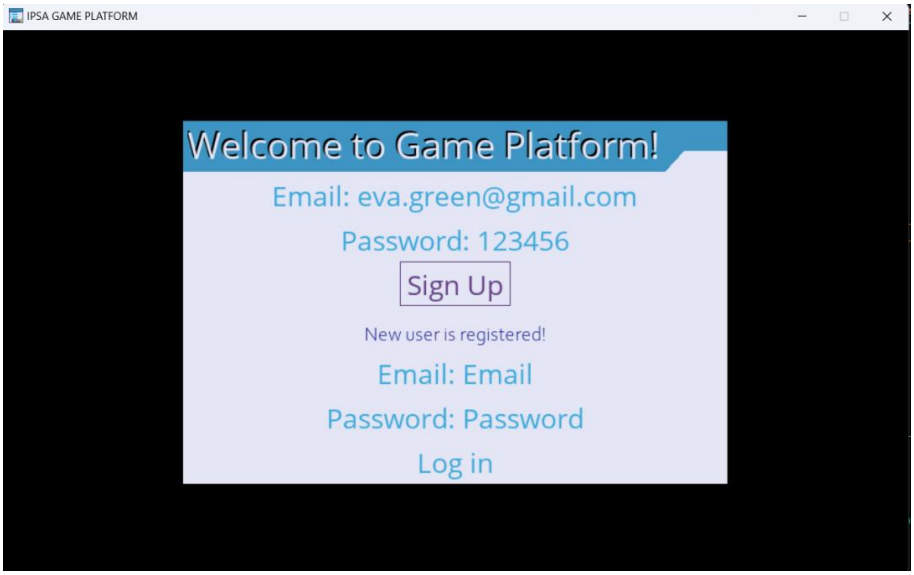

Figure 3: Sign up – The new user is registered.

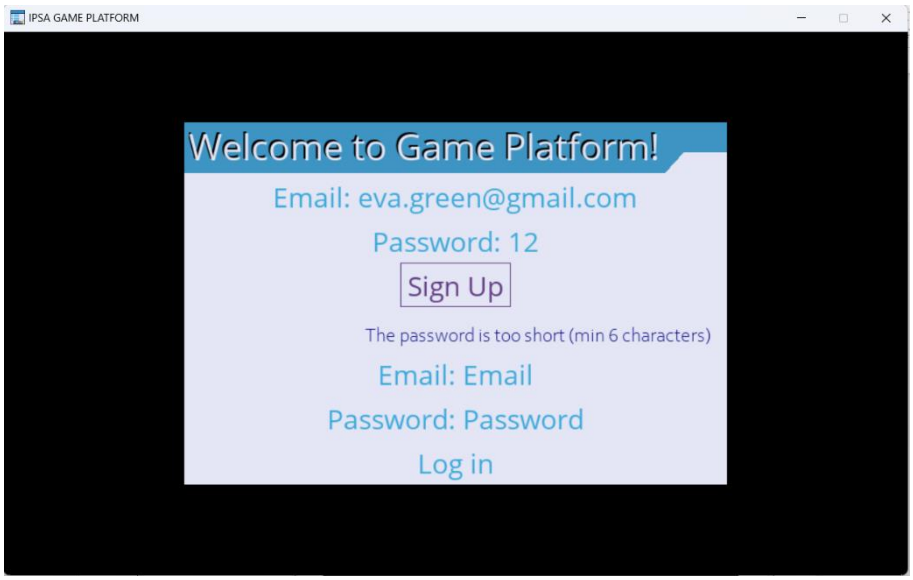

Figure 4: Sign up – The password is too short.

3) LOGIN (1.5 points)

Create a ***`Log-in`*** function that uses an email and a password.

- If a user doesn't exist or the username/password is wrong, display on the frame **"Username or password is not correct!"** as shown in Figure 5.
- If the username and password are correct, switch the frame to the main menu frame as presented in Figure 6.

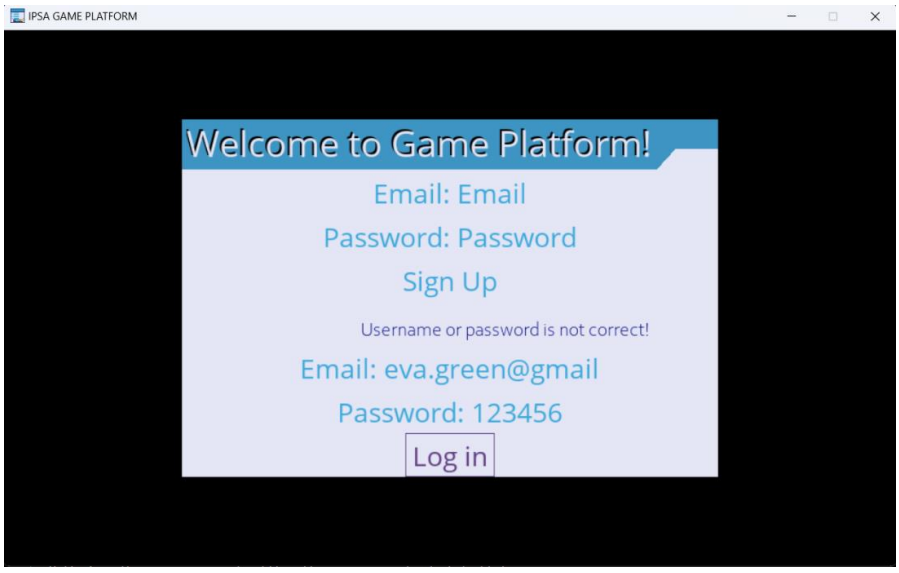

Figure 5: Log in – Username or password is not correct.

4) Main Menu (0.5 points)

In the main, add the title ***Menu*** and 6 buttons:
Button 1: **Profile,** Button 2: **How to play,** Button 3: **Game,** Button 4: **Score Board,** Button 5: **More Games,** Button 6: **Settings,** Button 7: **Quit.**
All the buttons allow us to return to the main menu.

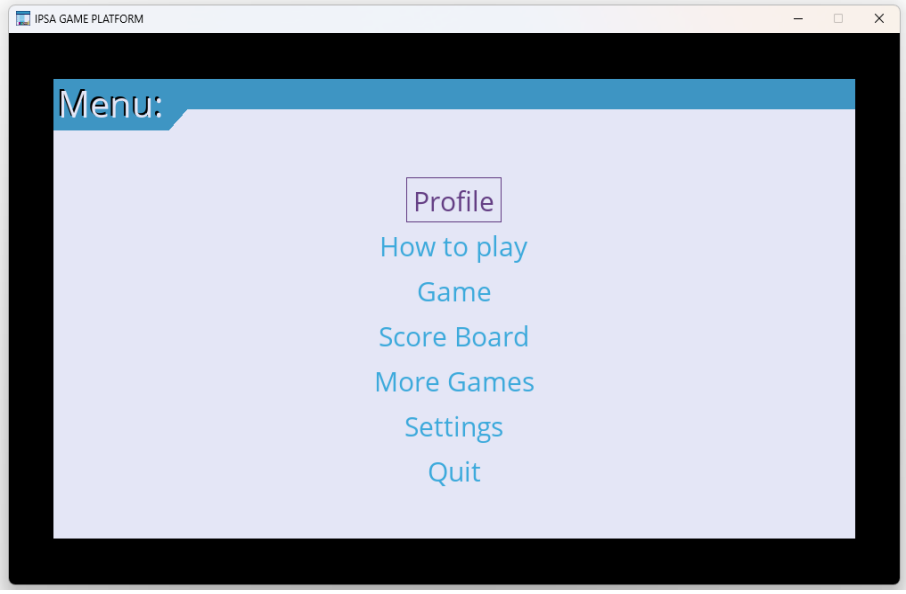

Figure 6: The main menu

Step 2: Run ***2_Example-DisplayImageandText.py*** file to understand how to display images and text on the frame using pygame library.

5) Profile (1.25 point)

Step 3: Run ***3-Example-Change_image_by_Clicking.py*** file as an example. **Button 1** (Profile) is to choose your character by clicking or by uploading your image as presented in Figure 7.

Figure 7: The main menu – Profile

Step 4: Run ***4_Example-UploadImage.py*** file to understand how to upload an image using pygame and Tkinter library. If the user prefers to upload their image, it is also possible to write a nickname as presented in Figure 8.

Figure 8: The main menu – Profile

6) How to play (0.5 points)

Button 2 (How to play) is to present the username and game information as presented in Figure 9. (All the buttons allow us to return to the menu.)

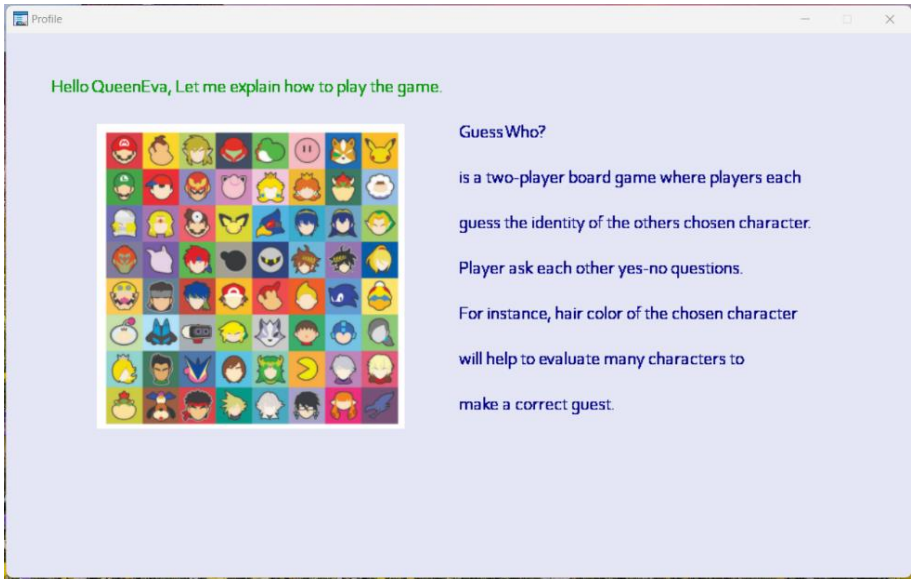

Figure 9: The main menu – How to play

7) Score Board (1.25 points)

Button 4 (Score Board) is to present the score of the gamers as presented in Figure 10. You can change the color of the text, add stars for the top 3 gamers, etc. The username and score information comes from the ***`File_username.csv`*** file.

Figure 10: The main menu – Score Board

8) More Games (1 point)

Button 5 (More Games) is to present 3 additional games: ***"Snake"***, **"Table Tennis"**, and "**Starfield"** as given in Figure 11. The game files are provided in the student file. They should be embedded in the main code. When the user clicks one of the games, the new frame should open to play the game and should return to the same menu when it finishes.

Step 5: ***`5-Example-Image_Open_New_Window.py`*** file is to understand how to switch between frames by clicking images to present your additional games.

Figure 11: More Games

9) Settings (0.5 point)

Button 6 (Settings) is to present difficulties, sound, language, etc. as shown in Figure 12. The entries propose switch options. For example, Difficulty: easy and hard, Sound: on and off.

1. You need to add one more feature that you think your gaming platform should have.
2. You need to use difficulty and one of your choice features in the game. (Minimum 2 features will be adapted in your game)

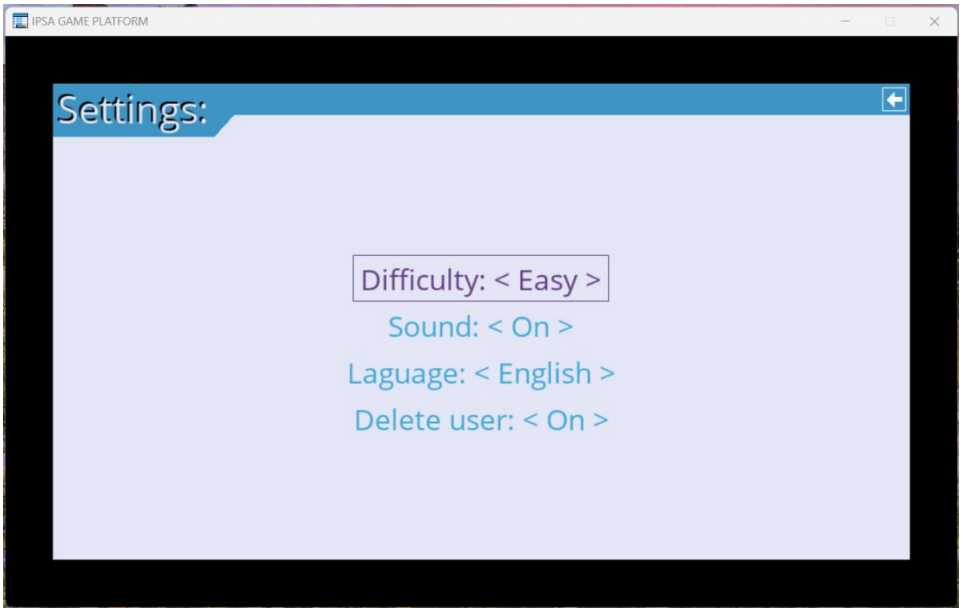

Figure 12: Settings

10) Game (4 points)

Button 3 (Game) is to present your main game.

1. If the difficulty is easy, show the game with 11 cards as presented in Figure 13.
2. If the difficulty is hard, show the game with 18 cards as presented in Figure 14.
3. There will be two additional frames between the two players' frames to switch the gamers. (Since gamers play the game with a single laptop.) as presented in Figures 15 and 16.
4. The game will propose random cards to guess your card and a random card as the opponent card in each game.
5. For the images, the given images in the student file should be used.
6. There should be a ***delete*** button to evaluate characters.
7. There should be a ***My Guess*** button to guess your card.
8. There should be a S***witch frame*** button to switch between the two players' frames.
9. If a gamer makes a guess, the game finishes. The frame shows who is the winner as presented in Figure 17.

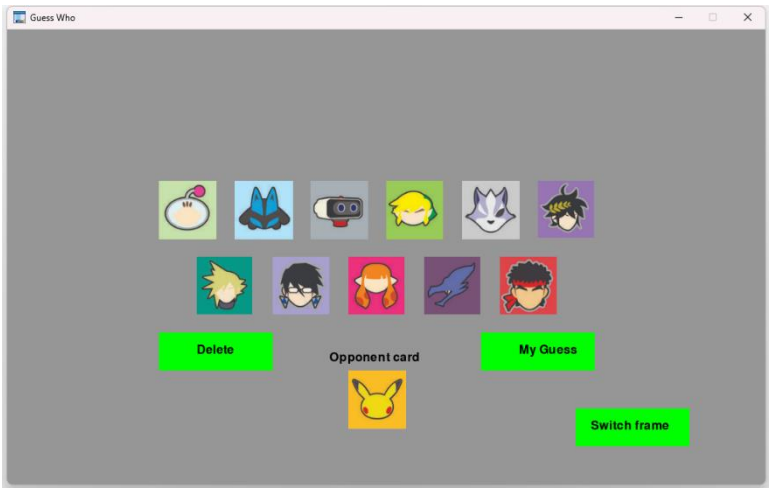

Figure 13: Game – difficulty easy

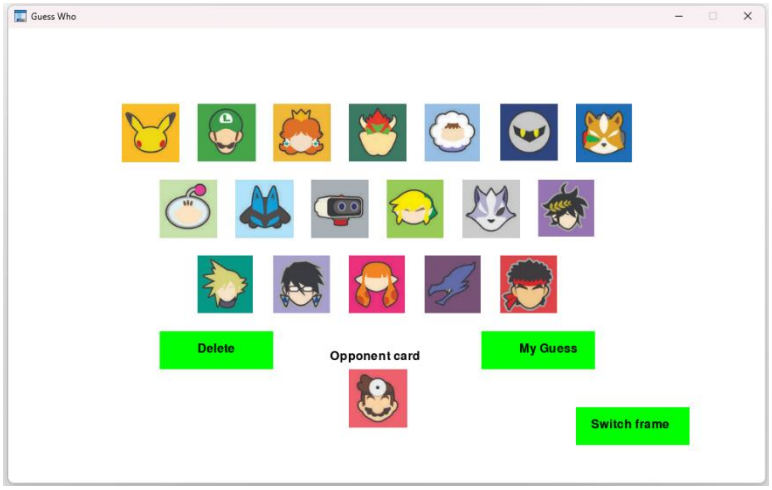

Figure 14: Game – difficulty hard

Figure 15: Game – switch to the second gamer frame

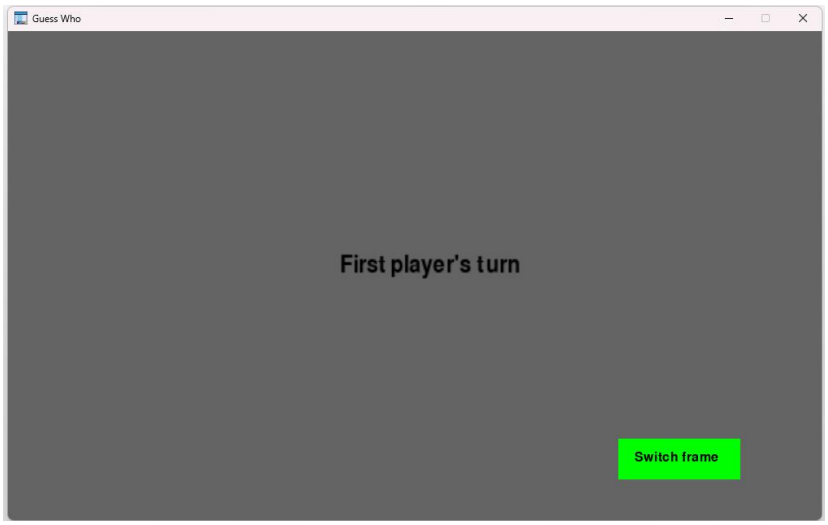

Figure 16: Game – switch to the first gamer frame

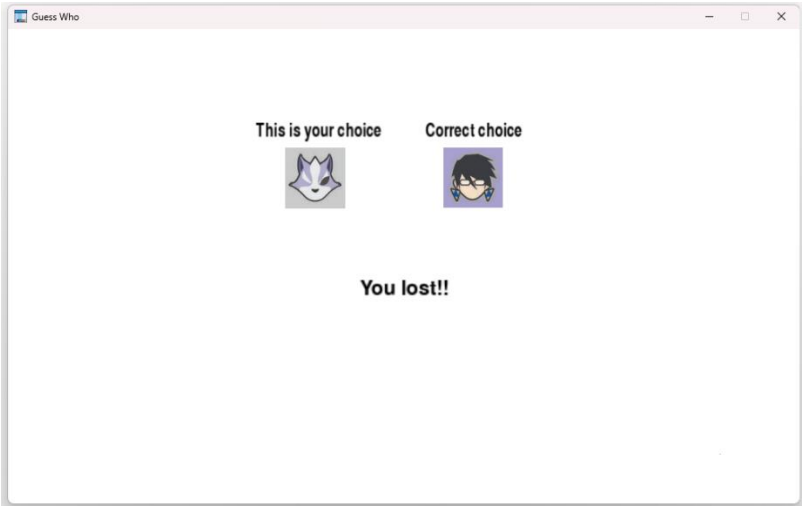

Figure 17: Game – result

11) Quit (0.5 points) — **Button 7** (Quit) is to close the main menu frame.

12) Report (3 points) — Answer the questions given in the student file.

13) Presentation (4 points) — Demo presentation and explanation of the code.