

Partie « Graphic Interface User GUI.py »

Cette partie est réalisée dans le but de faire l'acquisition des données brutes par l'utilisateur.
This part is carried out for the purpose of acquiring the raw data by the user.

Afin d'utiliser ce morceau du programme il faut avoir les fichiers GUI.py, modif_tesxte.py, renommer_puits.py, donnees_puit.txt et donnees.txt dans le même dossier.
In order to use this part of the program, you must have the GUI.py, modif_tesxte.py, rename_puits.py, data_wind.txt and data_txt files in the same folder.

GUI.py :

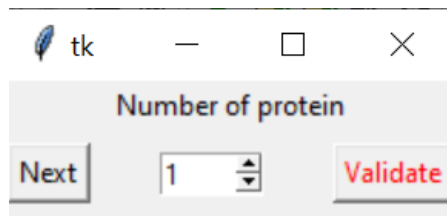
C'est le programme principal que devra lancer l'utilisateur.
It is the main program that the user will have to launch.

Ce programme contient 5 classes. Chacune correspondante à une fenêtre graphique. Les fenêtres graphiques ont été réalisées à partir de la bibliothèque Tkinter.
This program contains 5 classes. Each corresponding to a graphic window. The graphics windows were created from the Tkinter library.

class StartPage : Sert à acquérir le nombre de protéines qui vont être utilisées. L'information sera stockée dans le fichier *donnees.txt* quand l'utilisateur appuie sur le bouton «Validate » via la méthode **def cliquer**.

Une fois l'information écrite le bouton « Next » permet de détruire la fenêtre et de passer à la fenêtre suivante.

Used to acquire the number of proteins that will be used. The information will be stored in the data.txt file when the user presses the «Validate » button via the def cliquer method. Once the information has been written, the « Next » button allows you to destroy the window and go to the next window.



class PageOne : Sert à acquérir le nombre de plaque 96 trous qui sont à traiter. L'information sera stockée dans le fichier *donnees.txt* quand l'utilisateur appuie sur le bouton «Validate » via la méthode **def cliquer2**. (⚠ Pour la création des worklistes ce paramètre n'est pas encore pris en compte)

Une fois l'information écrite le bouton « Next » permet de détruire la fenêtre et de passer à la fenêtre suivante.

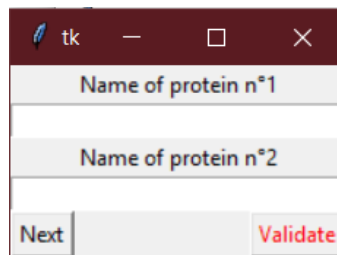
Used to acquire the number of 96-hole plates that are to be processed. The information will be stored in the data.txt file when the user presses the "Validate" button via the `def cliquer2` method (⚠ For the creation of worklists this parameter is not yet taken into account). Once the information has been written, the « Next » button allows you to destroy the window and go to the next window.



`class PageTwo` : Sert à acquérir le nom de chaque protéines qui vont être utilisées. L'information sera stockée dans le fichier donnees.txt quand l'utilisateur appuie sur le bouton «Validate » via la méthode `def cliquer`.

Une fois l'information écrite, le bouton « Next » permet de détruire la fenêtre et de passer à la fenêtre suivante.

Used to acquire the name of each protein that will be used. The information will be stored in the data.txt file when the user presses the «Validate » button via the `def cliquer` method. Once the information is written, the « Next » button allows to destroy the window and go to the next window.

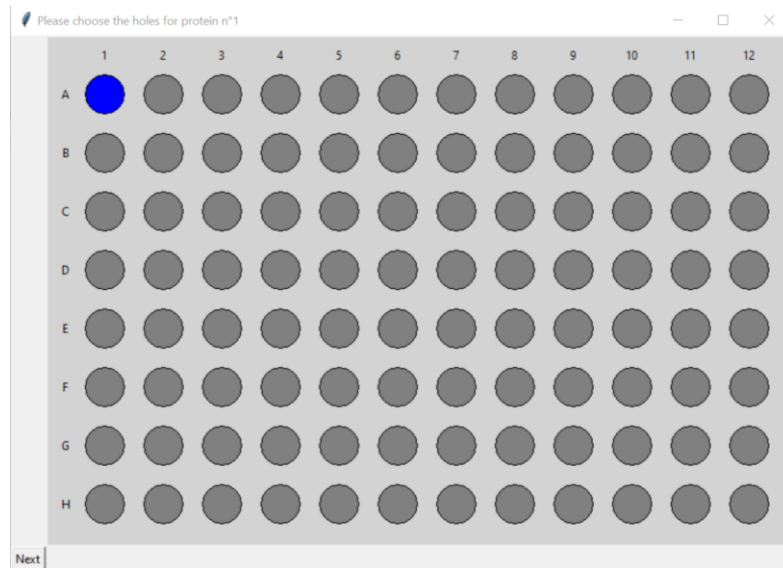


`class AppliCanevas` : Used to plot an interface that looks like a 96-well-plate so that the user can specify the place where he/she wants to put the proteins in. When this class is called in the main, it plots the window and runs the method `def creer_widgets` that is going to execute the following classes:

- `def dessine_cercles` : draws wells on the interface with a grey background
- Detection of a click by the user (`.bind`): When the user clicks on a well, the method selection is called. What selection does will be described later.
- `Indice_wells`: adds the letters (A,B,...,H) and the numbers from 1 to 12 to the well-plate representation.

- **def selection:** a file is created: "donnees_puits.txt". If the well has never been selected before, the well turns into blue and the name of the well is written in the file.

If the well selected has already been selected, the well is deselected: it is removed from the file "donnees_puits.txt" and the colour turns into grey again.



class PageFour : Sert à acquérir les paramètres spécifiques à chaque protéine pour les étapes de "Dispense" et "Rincing". Les information sera stockée dans le fichier *proteine_n_%.txt* quand l'utilisateur appuie sur le bouton «Validate » via la méthode **def cliquer**.

*Used to acquire the parameters specific to each protein for the "Dispense" and "Rincing" steps. The information will be stored in the *proteine_n_%.txt* file when the user presses the "Validate" button via the **def cliquer** method.*

Incubation parameters for protein n°1

Incubation Step: DiTi capacity for : Aspire / Dispense (μL):
 10 10
 Position of Diti: 5-1 1 1 5-1

Rinsing Step: DiTi capacity for Aspire / Dispense (μL):
 10 10
 Position of Diti: 5-1 1 1 5-1

Incubation Step: Volume to dispense in each well (μL):

Incubation Step: Incubation time (min):

Rinsing Step: Volume to aspirate for rinsing (μL):

Rinsing Step: Incubation time (min):

Nb of rinsing steps: 1

Multipipetting Y/N: ☐

Next Validate

Le programme d'exécution principal est réalisé dans la partie main qui va appeler les classes une par une dans l'ordre voulue. Une boucle est réalisé pour l'affichage de **AppliCanevas** et **PageFour** afin d'acquérir les paramètres pour chaque protéine (si plus de 1).

*The main execution program is carried out in the main part of the code which will call the classes one by one in the desired order. A loop is created for displaying **AppliCanevas** and **PageFour** in order to acquire the parameters for each protein (if more than 1).*

```

415 if __name__ == "__main__":
416     fenetre1 = Tk()
417     interface = StartPage(fenetre1)
418     interface.mainloop()
419     interface.destroy()
420
421     fenetre2 = Tk()
422     interface = PageOne(fenetre2)
423     interface.mainloop()
424     interface.destroy()
425
426     fenetre3 = Tk()
427     interface = PageTwo(fenetre3)
428     interface.mainloop()
429     interface.destroy()
430
431     (nb_protein,name_protein)=lire()
432     txtprot(nb_protein,name_protein)
433
434     for i in range (nb_protein):
435         j=i+1
436         titre=('Please choose the holes for protein n°%d' % j)
437         app = AppliCanevas()
438         app.title(titre)
439         app.mainloop()
440         app.destroy()
441         writehole(j)
442
443         titre=('Incubation parameters for protein n°%d' % j)
444         fenetre4 = Tk()
445         fenetre4.title(titre)
446         fenetre4.geometry("500x500")
447         app = PageFour(fenetre4)
448         app.mainloop()
449         app.destroy()
450

```

modif_texte.py :

Ce programme contient des méthodes qui sont ensuite appelé dans le programme principal GUI.py.

This program contains methods which are then called in the main GUI.py program.

def lire() : appelé dans la fonction principale GUI.py dans la partie main. Elle permet de récupérer le nombre de protéines ainsi que leur noms dans le fichier *donnees.txt*
called in the main function GUI.py in the part main_init. It allows to retrieve the number of proteins as well as their names in the data.txt file

def txtprot(nb_protein,name_protein) : appelé dans la fonction principale GUI.py dans la partie main. Elle permet de créer les fichiers texte *Protein_n_%.txt* et d'écrire le nom de la protéine correspondante dans chacun des fichiers.
called in the main function GUI.py in the part main_init. It allows you to create the Protein_n_%.txt text files and to write the name of the corresponding protein in each of the files.

def writehole(j) : appelé dans la fonction principale GUI.py dans la partie main. Elle permet de récupérer les puits sélectionnés à partir de *donnees_puit.txt* et de les écrire dans le fichier *Proteine_n_%.txt* pour chaque protéine.

called in the main function GUI.py in the part main_init. It makes it possible to recover the wells selected from data_well.txt and to write them in the Protein_n_%.txt file for each protein.

Renommer_puits.py:

This file contains several little functions used in GUI

- Increase_letter: increments the letters
- Lien_lettre_nombre: associates 2 numbers to a letter. The first one is 1 if the letters from A to D and 2 for the letters from E to H. The 2nd number is 1 for A, 2 for B etc.
- Traduire: if the user wants to select the well A1 for example, the computer will not save A1. So this function allows to translate what the computer saves into A1.
- Modifier_fichier: used to remove a well from "donnees_puits.txt" if the user clicks twice on the same well
- Puits_deja_ecrit: used to know whether the well has already been selected or not

Well_selection_2.py:

This function takes a file name as parameter. It returns 3 lists: bras, bloc and colonnes.

- bloc(i)=0108?0 if the selected well is part of the top of the plate (that is if the letter of the well is between A to D)

bloc(i)=0108 1 if the selected well is part of the bottom of the plate (that is if the letter of the well is between E to H)
- colonnes(i)= number of the column of the ith selected well but translated for the robot
- bras(i)=number of the arm that the robot must use to fill the ith well.