1 Interface graphique

1.1 Introduction

Jusqu'ici, tous les programmes que l'on a écrit suivait une exécution ligne par ligne dictée par le programme. On commence par la première ligne et on finit par la dernière. Avec une interface graphique, on change de point de vue. En effet, l'utilisateur va interagir avec le programme en utilisant des fenêtres, la souris mais aussi le clavier tout cela alors que le programme est en train de fonctionner. En mode "console", c'est le programme qui a le contrôle et quand il a besoin de l'utilisateur, il lui donne la main avec des fonctions comme input . C'est seulement à ce moment là que l'utilisateur peut interagir avec le programme. Avec une interface graphique, c'est le programme qui va réagir en fonction de ce que souhaite l'utilisateur. Le programme se met alors en attente des ordres donnés par l'utilisateur. On dit qu'il réceptionne des événements.

Pour réaliser une interface graphique, il existe un module sous Python qui se nomme **Tkinter**. Nous allons présenter quelques notions de bases, dans les différentes séances qui vont suivre mais cela ne sera évidemment par exhaustif. Il vous faudra approfondir par vous même ce sujet.

1.2 Une première fenêtre

Commençons par une petite application qui possède qu'un bouton et qui se termine quand on clique sur un bouton. Les éléments graphiques se nomment des **widgets**. Dans le programme qui suit, nous allons utiliser deux widgets différents : un label et un bouton.

Un label est tout simplement une zone de texte pré-remplie. L'utilisateur ne peut la modifier. Avec un bouton, l'utilisateur va cliquer dessus ce qui va déclencher une action de la part du programme. Dans notre premier exemple, on va juste arrêter le programme.

```
from tkinter import *

# on cree la fenetre
fen1=Tk()

# On cree le texte (label) que l'on attache a la fenetre
tex1 = Label(fen1, text='Bonjour tout le monde !', fg='red')

# on insere le label dans la fenetre
tex1.grid(row=0,column=0)

# on cree le bouton que l'on attache a la fenetre
bou1 = Button(fen1, text='Quitter', command = fen1.destroy)

# on insere le bouton dans la fenetre
bou1.grid(row=1,column=0)

# affichage de la fenetre !
fen1.mainloop()
```

On peut remarquer que lors de la création du widget, on peut lui passer, ce que l'on nomme, des paramètres. Par exemple, pour le Label, on a spécifié que l'on souhaitait avoir un texte de couleur rouge.

On a ensuite ajouté ce widget dans la fenêtre à l'aide de la commande "grid". C'est une fonction de placement des fenêtres. Il faut imaginer la fenêtre comme une grille. Par exemple, à la ligne 13, on indique que le label sera placé sur la ligne 0 et la colonne 0.

On peut aussi noter que le programme se termine avec l'appel à la méthode mainloop qui va gérer les événements reçu par l'interface.

1.3 On récupère une entrée

Passons maintenant à plus compliqué en récupérant une chaîne de caractère. Par exemple, on souhaite demander son nom à l'utilisateur et récupérer cette information. Dans notre programme, on affichera le nom donné dans la console pour montrer que l'on a bien récupéré le nom donné.

```
1  from tkinter import *
2
3  def Affiche():
4          print(nom.get())
5          Wnom.delete(0,END)
6
7  fen1 = Tk()
8  fen1.title("Pour les textes !")
9
10 # La question
11 Wtexte= Label(fen1, text="Quel est votre nom ?", fg="black",width=30, height=3)
12 Wtexte.grid(row=0,column=0)
13
14 # La reponse
15 nom=StringVar()
16 Wnom=Entry(textvariable=nom, width=30)
17 Wnom.grid(row=1,column=0)
18
19 # un bouton pour valider
20 Wvalide = Button(fen1,text="Valider",fg="blue",command=Affiche)
21 Wvalide.grid(row=2,column=0)
22
23 # Et enfin un bouton pour quitter
24 Wquitter = Button(fen1,text="Quitter",command=fen1.quit)
25 Wquitter.grid(row=3,column=0)
26
27 fen1.mainloop()
```

Noter bien, à la ligne 20, comment on a fait pour traiter le nom de l'utilisateur puis comment dans la fonction Affiche, on fait pour trouver la valeur donnée par l'utilisateur.

EXERCICE 1:

Faire un petit programme qui demande un identifiant et un mot de passe et qui vérifie si le nom rentré ainsi que le mot de passe sont bien correct.

On utilisera deux listes, une qui va contenir les identifiants et l'autre les mots de passe.

2 Canvas

Le but de cette section est de découvrir le widget Canvas qui va être très pratique pour faire des dessins et y placer des images.

2.1 Droites, rectangles et Cie

Nous allons tout d'abord commencer par tracer des lignes, des cercles et placer des images dans cette surface rectangulaire qu'est le Canvas. Tapez le code suivant dans un nouveau programme Python.

```
from tkinter import *

fen1 = Tk()
fen1.title("Des courbes")

Wsurface=Canvas(fen1,height=300,width=300)
Wsurface.grid(row=0,column=0)

Wsurface.create_line(0,0,300,300,width=2,fill="#B33333")
Wsurface.create_line(0,300,300,0,width=2,fill="blue")

Wsurface.create_rectangle(100,100,200,200,outline="yellow",fill="brown")

Wsurface.create_oval(100,100,200,200,fill="white")
Wsurface.create_text(150,150,fill="black",anchor='w',text="Bonjour")

Wquitter = Button(fen1,text="Quitter",command=fen1.quit)
Wquitter.grid(row=1,column=0)
fen1.mainloop()
```

La compréhension du code est relativement simple. Reste les paramètres. Cette surface est associée à un repère orthogonal dont l'origine est située en haut à gauche. L'axe des abscisses est horizontale orienté vers la droite et celui des ordonnées verticale orienté vers le bas. Ainsi un point dans cette surface est repéré à l'aide de 2 nombres entiers positifs. Dans notre exemple, on a défini que la surface avait 300 pixels de largeur et autant en hauteur. Le point se situant au milieu de cette surface a ainsi pour coordonnées (150,150).

EXERCICE 2:

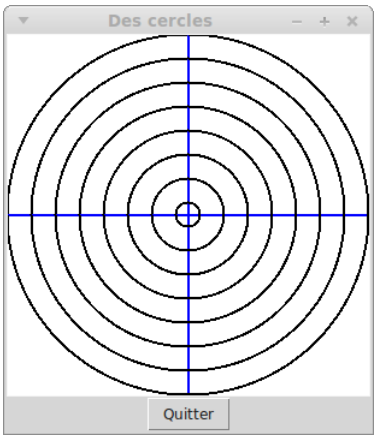
En reprenant l'exemple précédent,tracez une ligne verticale rouge d'épaisseur 4 au milieu de la fenêtre.

Le paramètre anchor peut prendre différentes valeurs : "n", "ne", "e", "se", "s", "sw", "w", "nw" et "center" qui indique comment positionner un texte dans la surface. Tester les différentes valeurs afin de comprendre l'utilité de ces différentes valeurs.

2.2 Exercices

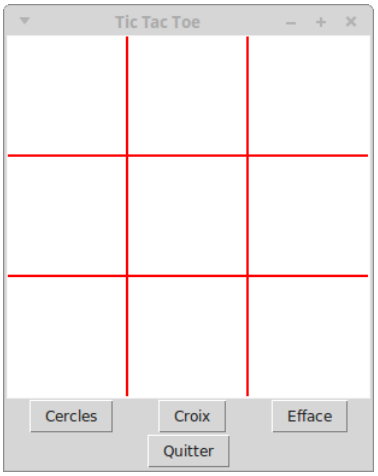
EXERCICE 3:

Créez une cible comme celle-ci-dessous :

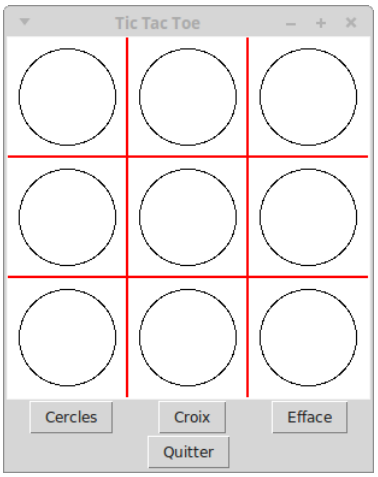


EXERCICE 4:

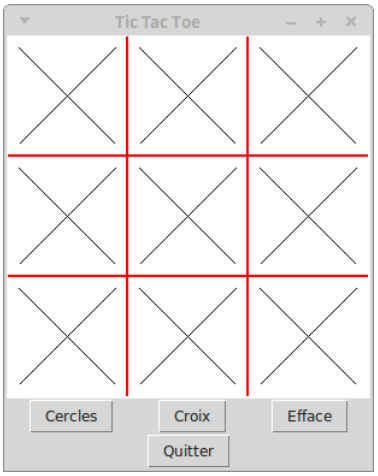
Créer l'interface suivante afin d'afficher soit des cercles soit des croix. Au début on aura une interface vide comme ci-dessous :



Quand on cliquera sur le bouton «Cercles» on obtiendra l'interface suivante :



De même pour les «Croix» :



Et le bouton «Efface» redonnera l'interface vide du début.

EXERCICE 5:

Reprendre l'exercice sur le jeu du pendu et proposé une version graphique.