Cours

NSI

Interface graphique en python

1) Introduction et vocabulaire

Python est fourni avec le module graphique tkinter (on l'importe au début des programmes)
La première commande consiste toujours à créer un **OBJET** tkinter par:
nom_quelconque=Tk()
Cet objet va ensuite contenir des éléments graphiques appelés **WIDGETS**.
Les widgets principaux sont:
Frame, qui est lui-même un conteneur, il contient d'autres widgets et sert à la mise en page,
le widget Frame est un widget dit "parent"
Label, texte non modifiable par l'utilisateur, mais modifiable par le programme,
Entry, zone de texte où l'utilisateur peut saisir des informations,
Text, super widget zone de texte (nombreuses possibilités d'interaction!!!),
Button, pour déclencher une fonction,
Canvas, pour faire des dessins,….
Chaque widget est configurable au niveau des couleurs, des bordures, de la police, ……
Comme toujours, n'hésitez pas à consulter la référence en ligne (voir site ISN)

2) Positionnement des widgets

Il y a 2 outils principaux pour placer les widgets, le **packer** et le **grider**.
Le packer place les widgets au fur et à mesure de leur création, de gauche à droite et de haut en bas. Il possède néanmoins des instructions de placement (TOP, LEFT, ……)
Le grider place les widgets en ligne et en colonne à l'endroit que vous avez choisi:
toto.grid(row=0,column=2) place le widget toto à la ligne 0 et la colonne 2 (origine en haut à gauche).
Dans un même conteneur (Frame), il ne faut pas mélanger le packer et le grider!
Par contre on peut le faire dans des conteneurs différents (voir exemple tkinter_multi.py)
Au début utilisez le grider, c'est le plus facile à maîtriser. Il dispose de nombreuses options et surtout :
columnspan (et rowspan) pour placer un widget sur plusieurs colonnes (lignes),
sticky pour que le widget remplisse l'espace disponible,….
On rencontrera ces options peu à peu! (Consulter la référence Tkinter!!!)

3) Un début

Voici un exemple de création graphique, avec un seul conteneur, un texte court, un bouton.
(Téléchargeable sous le nom "tkinter_boum.py")

```
1   from tkinter import *
2   def boum():
3       titre.config(text="BOUM", fg="red")
4   cadre=Tk()
5   cadre.title("TEST")
6   titre=Label(cadre,text="ATTENTION DANGER",fg="blue",bg="white")
7   titre.grid(row=0,column=0)
8   bouton=Button(cadre,text=" Surtout ne pas appuyer sur le bouton!",command=boum)
9   bouton.grid(row=1,column=0)
10  cadre.mainloop()
```

Ligne 4: création de l'objet tkinter. On le nomme cadre. C'est la fenêtre parente de tous les widgets
Ligne 5: le titre de la fenêtre du programme
Ligne 6: création d'un widget de type **Label**, que l'on nomme "titre" pour agir dessus dans le programme.
(**fg** est un raccourci de foreground et **bg** de background, commandes utilisées pour les couleurs)
Ligne 7: on place ce widget avec le grider en (0,0)
Ligne 8: création d'un widget de type **Button**. La fonction boom est lancée lors de l'appui.
Ligne 9: placement du bouton
Remarque: on aurait pu réunir les lignes 8 et 9 en une seule, car le programme ne doit pas agir sur le bouton. Il n'a donc pas besoin d'avoir un nom. On aurait alors la ligne:

```
Button(cadre,text=" Surtout ne pas appuyer sur le bouton!",command=boom).grid(row=1,column=0)
```

Ligne 10: c'est ce que l'on nomme "démarrage de la boucle d'événements" sur le cadre. L'ordinateur détecte régulièrement (des centaines de fois par seconde!) le passage et les clics de souris sur le cadre, le clavier,…..
Lignes 2 et 3: définition de la fonction lancée par le bouton.
La méthode "config" modifie le Label "titre".
C'est encore un exemple de langage "OBJET", la fonction (appelée "méthode" dans le langage objet), s'applique à la fin de l'objet.
Elle est précédée d'un point. On a déjà rencontré ceci dans les chaînes avec la méthode "count", ou les listes avec "append", ou encore avec le grider…Et ce n'est pas fini!!)
Cette méthode "config", commune à plusieurs widgets, permet de modifier les couleurs (bg et fg), la police (type, taille, gras, italique, ….) ,le texte, …..

4) Exercice

Compléter le programme ci-dessus en ajoutant un bouton "Désamorcer la bombe" qui replace le widget "titre" dans sa situation initiale.
Vous allez donc ajouter "bouton1", le placer avec le grider où vous voulez.
Puis vous aller créer la fonction "desamorcer()" qui modifie le widget "titre".

5) Le widget Entry

C'est le widget de base pour les échanges avec le programme.
On peut principalement récupérer son contenu avec la méthode get(),
modifier son contenu avec les méthodes "insert()" et "delete()"
désactiver et activer son utilisation avec l'option "state" (NORMAL ou DISABLED).
Il y a 23 options et 15 méthodes !
Voici un exemple où l'utilisateur indique son année de naissance et son nom.
Le programme le salue et lui donne son âge dans un Label ("info").
(On reprend l'un de nos premiers programmes python!)
Il est téléchargeable sous le nom "tkinter_salut.py"

```
from tkinter import *
def affiche():
    age=2012-int(entre_annee.get())
    nom=entre_nom.get()
    texte="Bonjour "+nom+". 2012 est l'année de tes "+str(age)+" ans.
    info.config(text=texte)
cadre=Tk()
cadre.title("SALUT")
titre_nom=Label(cadre,text="Ecrivez votre nom ici: ",font="arial 20 bold")
titre_nom.grid(row=0,column=0)
entre_nom=Entry(cadre, width=20,font="arial 20 bold")
entre_nom.grid(row=0,column=1)
titre_annee=Label(cadre,text="Votre année de naissance: ",font="arial 20 bold")
titre_annee.grid(row=1,column=0)
entre_annee=Entry(cadre, width=4,font="arial 20 bold")
entre_annee.grid(row=1,column=1,sticky=W)
bouton=Button(cadre,text=" Dialoguer avec la machine",font="arial 20 bold",bg='yellow',fg='red',command=affiche)
bouton.grid(row=2,column=0)
info=Label(cadre,text="Moi, ordinateur, je vais vous causer",font="arial 20
bold",bg='white',fg='blue',relief=RAISED,border=4)
info.grid(row=3,column=0,columnspan=2,sticky=W+E)
cadre.mainloop()
```

Quelques remarques:
font="arial 20 bold" permet d'avoir de grands widgets!
Dans le grider, sticky=W, permet de placer le widget à gauche (West), sticky=W+E étire le widget à gauche et à droite. Il occupe alors toute la place disponible dans sa case (en largeur).
relief=RAISED,border=4 permet de mettre une bordure en relief d'épaisseur 4

6) Exercices

a) Compléter le programme ci-dessus en ajoutant un bouton "reset" qui efface le contenu des deux widgets Entry.
b) Améliorer ce programme pour qu'il demande la date de naissance complète, et qu'il réponde selon le cas: "Cette année tu as eu 25 ans", ou "Tu auras bientôt 25 ans", ou "Bon anniversaire, tu as 25 ans aujourd'hui"
c) Télécharger sur le site ISN les programmes "tkinter_multi.py" et "test_couleur.py".
Changer alors les couleurs des textes, des fonds…,
En effet tkinter possède un certain nombre de couleurs définies ('red', 'orange', ….), mais on peut aussi choisir une couleur par son code rvb (rouge, vert, bleu) ou hexadécimal
(de noir #000000 à blanc #FFFFFF)
Exemple: label.config(bg = "#74BBEB") qui donne un fond bleu clair.
Vous y verrez le widget Text (30 options, 45 méthodes!!!!) et une présentation en 2 Frames.

7) Les boîtes de message

Il existe plusieurs boîtes de message: les alertes (**messagebox**), celles qui demandent une info (**simpledialog**), celles qui permettent de choisir un fichier (**filedialog**), ….

Méthodes de messagebox: showinfo, showwarning, showerror, askquestion, askokcancel, askyesno, askretrycancel

Méthodes de simpledialog: askstring, askinteger and askfloat

Méthodes de filedialog: askopenfilename et asksaveasfilename

Ouvrez le fichier "tkinter_box.py" pour voir des exemples simples d'application.

```
from tkinter import *
def bonjour():
    messagebox.showinfo("Politesse","Bonjour tout le monde")
    info.config(text="T'as vu je suis poli!")
def motdepasse():
    passe=simpledialog.askstring("Sécurité", "Quel est le mot de passe?")
    info.config(text="Je connais ton mot de passe: "+passe)
def fichier():
    nom=filedialog.askopenfilename()
    info.config(text="Alors tu veux ouvrir le fichier: "+nom+"?")
cadre=Tk()
cadre.title("Boîtes de message")
info=Label(cadre,text="Ici des informations en temps réel!",font="arial 20 bold")
info.grid(row=0,column=0,sticky=W+E)
bouton1=Button(cadre,text=" Sois poli si t'es pas joli !",font="arial 20 bold",bg='yellow',fg='red',command=bonjour)
bouton1.grid(row=1,column=0,sticky=W+E)
bouton2=Button(cadre,text=" Mot de passe",font="arial 20 bold",bg='yellow',fg='red',command=motdepasse)
bouton2.grid(row=2,column=0,sticky=W+E)
bouton3=Button(cadre,text=" Ouvrir Fichier",font="arial 20 bold",bg='yellow',fg='red',command=fichier)
bouton3.grid(row=3,column=0,sticky=W+E)
cadre.mainloop()
```

8) Exercices

a) Parcourir la "référence tkinter" pour découvrir les boîtes de dialogue et leurs options.
b) Ecrire un programme (code de César) avec interface graphique qui demande dès le début un mot de passe (avec askstring) et qui s'arrête si le mot de passe est incorrect (instruction cadre.destroy()).
Il y aura 2 Entry, un pour le message, l'autre pour le décalage.
Un Label pour le message codé.
Et des belles couleurs, une belle présentation, un titre. Le plus beau programme sera publié sur le Net!

9) Les boutons radio

Le widget **Radiobutton** permet de ne proposer à l'utilisateur qu'un nombre restreint de choix, **un seul choix** étant possible dans le même groupe de boutons.
Les boutons d'un même groupe sont associés à la **même variable** tkinter qui doit être **déclarée**!
C'est une particularité de tkinter, ce module gère ses propres variables, utilisables par ailleurs dans le programme python (heureusement!!!)
Dans l'exemple ci-dessous, la variable couleur tkinter est déclarée à la ligne 8:

```
couleur=StringVar()
```

C'est une variable chaîne, il y a aussi IntVar(), …..
Etudions l'exemple: (Téléchargeable sous le nom "tkinter_radio.py")

```
1   from tkinter import*
2   def changer_couleur():
3       affichage.config(fg=couleur.get())
4   cadre=Tk()
5   cadre.title("Exemple de boutons radio")
6   affichage=Label(cadre,text='TEST DES BOUTONS RADIO',fg='white',bg='black',font="arial 20 bold")
7   affichage.grid(row=0,column=0,columnspan=3,sticky=E+W+N+S)
8   couleur=StringVar()
9   radio=Radiobutton(cadre,text='ROUGE',variable=couleur,value='red',bg='gray',command=changer_couleur)
10  radio.grid(row=1,column=0,sticky=E+W+N+S)
11  radio=Radiobutton(cadre,text='JAUNE',variable=couleur,value='yellow',bg='gray',command=changer_couleur)
12  radio.grid(row=1,column=1,sticky=E+W+N+S)
13  radio=Radiobutton(cadre,text='VERT',variable=couleur,value='green',bg='gray',command=changer_couleur)
14  radio.grid(row=1,column=2,sticky=E+W+N+S)
15  radio.select()
16  radio.invoke()
17  cadre.mainloop()
```

Lignes 2 et 3: définition de la fonction qui change la couleur du Label "affichage".

La méthode get() renvoie la valeur de la variable "couleur".

Ligne 6: création d'un Label dont on va changer la couleur de police. On le fait assez grand avec la commande "font="arial 20 bold" (police arial, taille 20, gras)
Ligne 7: placement du Label sur 3 colonnes (columnspan=3) et expansion dans toutes les directions: East, West, North, South

Lignes 9 à 14: création et placement des 3 boutons radio.

Ils portent tous le même nom (en fait on garde la trace seulement du dernier!), ce qui n'est pas important ici, car ils appellent la même fonction. On aurait pu les différentier si on voulait pouvoir modifier leur titre par exemple.
Ils sont tous affectés à la même variable tkinter "couleur" (ils forment donc un groupe), mais chacun correspond à une valeur différente de la variable. Cette valeur est affectée par "value=".
Ils lancent tous la fonction "changer_couleur".
Quand on clique dessus, la variable "couleur" prend la valeur "value" et la fonction est lancée.
Le widget "affichage" est alors configuré avec la nouvelle couleur.

Ligne 15 et 16: on sélectionne le dernier bouton et on déclenche sa fonction (facultatif)

10) Mini projets

Voici quelques suggestions:
Codage d'un nombre dans une base vers une autre.
Codage et décodage d'un fichier texte, avec sortie dans un fichier texte, avec choix du mode de codage…
Etude dynamique d'une suite avec graphique sur lequel on peut déplacer le 1° terme (widget Canvas!)