Chapitre Découverte

Initiation à Tkinter

Table des matières

La bibliothèque Tkinter (**T**ool **K**it **Inter**face) permet la gestion d'interface graphique en python.

1 La fenêtre principale

La création de l'objet principale de la bibliothèque Tkinter est la fenêtre à l'aide de la fonction `Tk`. Dans cette fenêtre nous pouvons placer des widget (windows gadget).
Les principaux widgets sont :

- les **Label** pour insérer un texte.
- les **Button** pour commander une action.
- les **Entry**, zones de saisie pour permettre l'entrée d'une donnée sous forme d'une chaine de caractères.
- les **Canvas** pour insérer des dessins ou des images.

Exemple :

```
from tkinter import *

ma_fenetre = Tk()

ma_premiere_phrase = Label(ma_fenetre,text="bonjour␣le␣monde")

ma_premiere_phrase.pack()

ma_fenetre.mainloop()
```

La fenêtre principale est la racine des objets qu'elle contient, c'est donc le premier argument dans la création de l'objet.

Exercice 1 :
Recopier le script :

```
from tkinter import *
ma_fenetre = Tk()
ma_fenetre.title('la␣fenetre')
ma_fenetre.geometry('400x300+400+400')
ma_fenetre['bg']="yellow"

ma_premiere_phrase = Label(ma_fenetre,text="bonjour␣le␣monde",fg = "blue")
ma_premiere_phrase.pack()

le_bouton = Button(ma_fenetre,text="Quitter", command = ma_fenetre.destroy)
le_bouton.pack(side="bottom")

ma_fenetre.mainloop()
```

Analyser les différentes commandes, et ajouter les commentaires.

2 Le canvas

Petit exemple :

```
from tkinter import *

fenetre = Tk()

# creation du canvas
canvas = Canvas(fenetre, width=500, height=500, bg="ivory")

# forme geometrique de base
rectangle = canvas.create_rectangle(50,60,150,160,fill="black")

cerle = canvas.create_oval(280,80,325,125,fill="pink")

polygone = canvas.create_polygon(80,250,200,200,120,350,fill = "blue")

for i in range(20) :
    canvas.create_line(5*i,0,0,100-5*i,fill="red")

#insertion de widgets
phrase = Label(text="bonjour")

bouton = Button(text = "Quitter" , command = fenetre.destroy )
petite_fenetre = canvas.create_window(50,150,window = phrase )
petite_fenetre_2 = canvas.create_window(50,200,window = bouton )


canvas.pack()
fenetre.mainloop()
```

3 Triangle de Sierpinski

3.1 Création d'un canvas

Créer une fenêtre de taille 1000×1000, en y incluant un canevas de même taille.

3.2 Dessiner un point dans le canvas

En utilisant la méthode `create_line` sur un canevas écrire la fonction `dessin_point` qui prend en arguments un couple de coordonnées (x, y) et qui dessine dans le canevas un point de 9 pixels centré en (x, y).

3.3 Milieu de deux points

Écrire la fonction `milieu` qui prend en argument deux couples de coordonnées et renvoie le couple des coordonnées du milieu des deux points.

3.4 Définition du triangle équilatéral

Déterminer les coordonnées des trois sommets A, B, C d'un triangle équilatéral, et placer les sur le canevas.
Ces trois points seront définis en variable globale.

3.5 Triangle de Sierpinski

On dispose d'un dé cubique équilibré dont les faces sont numérotées de 1 à 6.
On choisit un point M au hasard sur le canevas.
Lancer le dé.

- Si le numéro obtenu est 1 ou 2, placer le milieu de $[AM]$.
- Si le numéro obtenu est 3 ou 4, placer le milieu de $[BM]$.
- Si le numéro obtenu est 5 ou 6, placer le milieu de $[CM]$.

Le milieu précédemment construit est renommé M.
On réitère l'opération autant de fois que possible à l'aide du nouveau point M.

3.5.1 Lancer du dé

Écrire la fonction `lancer_de` qui prend en argument un couple de coordonnées (représentant le point courant M), et renvoie, après avoir lancer le dé, les coordonnées du nouveau point M.
On utilisera la fonction `randint` de la bibliothèque `random`.

3.5.2 Dessin du triangle

Écrire la fonction `triangle` qui prend en argument un entier n et qui dessine n points courants M.