

# Interface graphique avec le module Tkinter

---

[http://fsincere.free.fr/isn/python/cours\\_python\\_tkinter.php](http://fsincere.free.fr/isn/python/cours_python_tkinter.php)

## Contenu

Chapitre 7 - Interface graphique avec le module Tkinter .....	3
Logiciels utilisant Python et sa bibliothèque graphique Tkinter .....	3
Des scripts pour débiter .....	4
Exemple n°0 : widgets Button et Label .....	4
Exemple n°1 : widgets Button et Label .....	4
Exemple n°2 : widgets Frame, Label et Button .....	5
Exemple n°3 : widgets Entry, Label, Button et boîte de dialogue MessageBox .....	6
Exemple n°4 : widgets Spinbox et Label .....	7
Exemple n°5 : widgets Scale et Button .....	8
Exemple n°6 : widgets Canvas et Button .....	9
Exemple n°7 : widgets Canvas et Button ; gestion des images .....	10
Exemple n°8 : widgets Menu et Canvas ; gestion des images ; boîtes de dialogue FileDialog et MessageBox .....	13
Exemple n°9 : gestion du temps .....	16
Exemple n°10 : widgets Canvas et Button ; gestion du temps .....	17
Exemple n°11 : widget Canvas ; gestion du temps .....	18
Exemple n°12 : widgets Canvas et Button ; gestion de la souris .....	20
Exemple n°13 : widget Canvas ; gestion de la souris .....	22
Exemple n°14 : widgets Canvas et Button ; gestion du clavier .....	23
Exemple n°15 : widgets Checkbutton et Button ; musiques et sons avec pygame .....	25
Exercices .....	26
Quelques idées de projets .....	32
Programmes exécutables pour Windows .....	37
Webographie .....	38



# Chapitre 7 - Interface graphique avec le module Tkinter

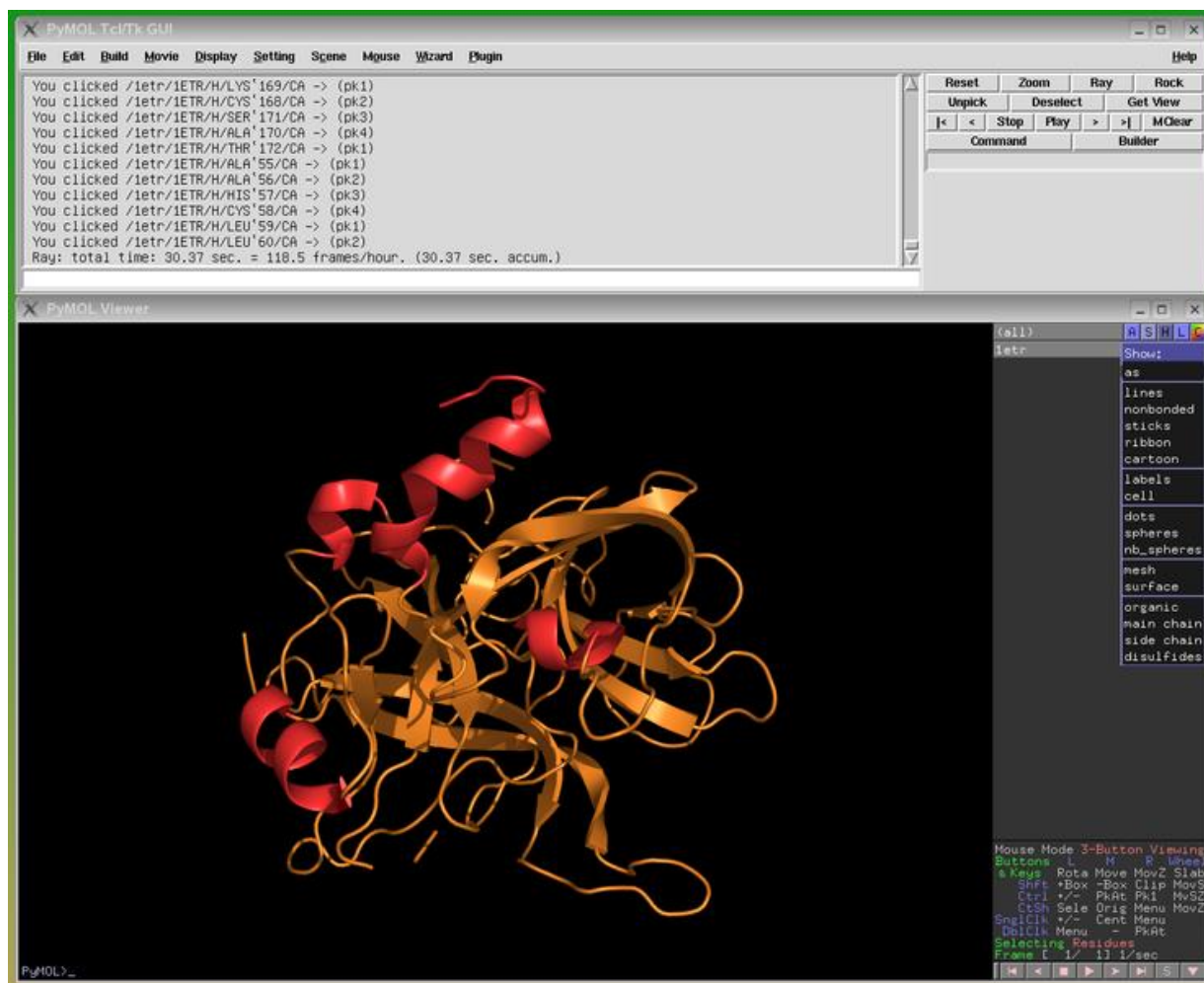
Le module `Tkinter` ("Tk interface") de Python permet de créer des interfaces graphiques (GUI : graphical user interface).

De nombreux composants graphiques (ou widgets) sont disponibles : fenêtre (classe `Tk`), bouton (classe `Button`), case à cocher (classe `Checkbutton`), étiquette (classe `Label`), zone de texte simple (classe `Entry`), menu (classe `Menu`), zone graphique (classe `Canvas`), cadre (classe `Frame`)...

On peut gérer de nombreux événements : clic sur la souris, déplacement de la souris, appui sur une touche du clavier, top d'horloge...

## Logiciels utilisant Python et sa bibliothèque graphique Tkinter

`Tkinter` est l'interface graphique des logiciels [IDLE](#) (environnement de développement intégré pour le langage Python) et [PyMOL](#) (logiciel libre de visualisation de structures chimiques en 3D) :

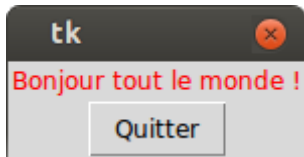


## Des scripts pour débiter

Les exemples qui suivent ont été écrits avec Python version 3.

### Exemple n°0 : widgets Button et Label

Commençons par le traditionnel Hello world !



```
# script bonjour.py
from tkinter import *

# Création de la fenêtre principale (main window)
Mafenetre = Tk()

# Création d'un widget Label (texte 'Bonjour tout le monde !')
Label1 = Label(Mafenetre, text = 'Bonjour tout le monde !', fg = 'red')
# Positionnement du widget avec la méthode pack()
Label1.pack()

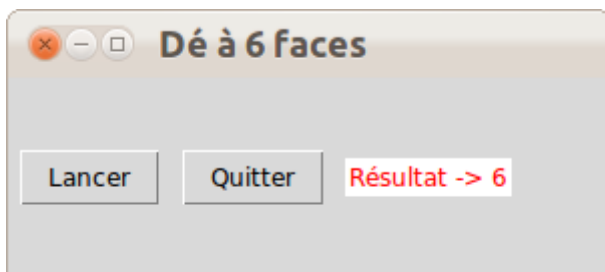
# Création d'un widget Button (bouton Quitter)
Bouton1 = Button(Mafenetre, text = 'Quitter', command = Mafenetre.destroy)
Bouton1.pack()

# Lancement du gestionnaire d'événements
Mafenetre.mainloop()
```

Ce code est détaillé [ici](#).

### Exemple n°1 : widgets Button et Label

Ce script simule un dé à 6 faces :



```
# script de.py
# (C) Fabrice Sincère

from tkinter import *
import random
```

```
def NouveauLance():
    nb = random.randint(1,6)
    Texte.set('Résultat -> ' + str(nb))

# Création de la fenêtre principale (main window)
Mafenetre = Tk()

Mafenetre.title('Dé à 6 faces')
Mafenetre.geometry('300x100+400+400')

# Création d'un widget Button (bouton Lancer)
BoutonLancer = Button(Mafenetre, text='Lancer', command = NouveauLance)
# Positionnement du widget avec la méthode pack()
BoutonLancer.pack(side = LEFT, padx = 5, pady = 5)

# Création d'un widget Button (bouton Quitter)
BoutonQuitter = Button(Mafenetre, text='Quitter', command =
Mafenetre.destroy)
BoutonQuitter.pack(side = LEFT, padx = 5, pady = 5)

Texte = StringVar()
NouveauLance()

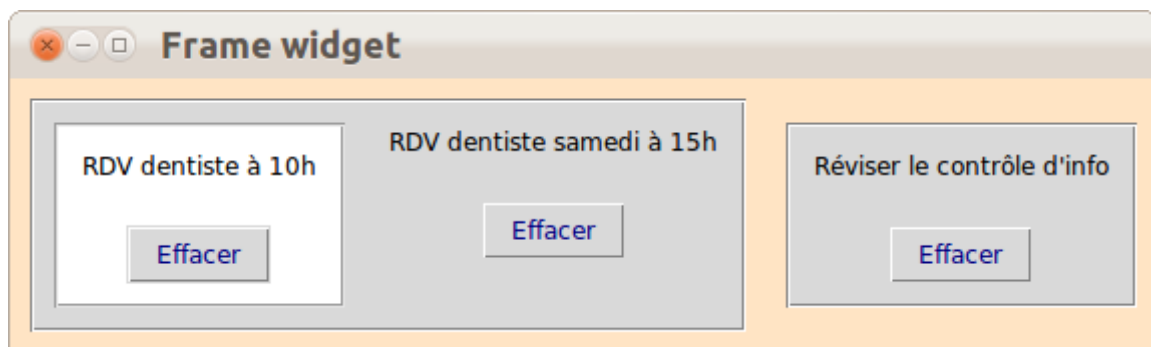
# Création d'un widget Label (texte 'Résultat -> x')
LabelResultat = Label(Mafenetre, textvariable = Texte, fg='red', bg
='white')
LabelResultat.pack(side = LEFT, padx = 5, pady = 5)

Mafenetre.mainloop()
```

Ce code est détaillé [ici](#).

## Exemple n°2 : widgets Frame, Label et Button

Un widget Frame est une zone rectangulaire qui peut contenir d'autres widgets.



```
# script frames.py
#(C) Fabrice Sincère
```

```
from tkinter import *
```

```
# Création de la fenêtre principale
Mafenetre = Tk()
Mafenetre.title('Frame widget')
```

```

Mafenetre['bg']='bisque' # couleur de fond

# création d'un widget Frame dans la fenêtre principale
Frame1 = Frame(Mafenetre,borderwidth=2,relief=GROOVE)
Frame1.pack(side=LEFT,padx=10,pady=10)

# création d'un second widget Frame dans la fenêtre principale
Frame2 = Frame(Mafenetre,borderwidth=2,relief=GROOVE)
Frame2.pack(side=LEFT,padx=10,pady=10)

# création d'un widget Frame... dans un widget Frame
# le widget Frame1 est le parent du widget Frame3
# le parent du widget Frame1 est le widget Mafenetre (fenêtre principale)
Frame3 = Frame(Frame1,bg="white",borderwidth=2,relief=GROOVE)
Frame3.pack(side=LEFT,padx=10,pady=10)

# création d'un widget Label et d'un widget Button dans un widget Frame
Label(Frame1,text="RDV dentiste samedi à 15h").pack(padx=10,pady=10)
Button(Frame1,text="Effacer",fg='navy',command=Frame1.destroy).pack(padx=10,pady=10)

Label(Frame2,text="Réviser le contrôle d'info").pack(padx=10,pady=10)
Button(Frame2,text="Effacer",fg='navy',command=Frame2.destroy).pack(padx=10,pady=10)

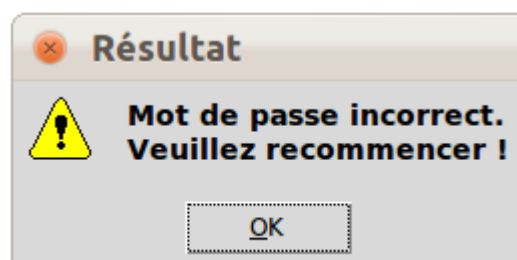
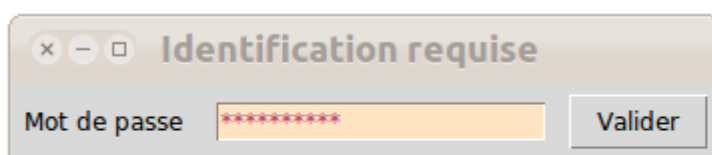
Label(Frame3,text="RDV dentiste à 10h",bg="white").pack(padx=10,pady=10)
Button(Frame3,text="Effacer",fg='navy',command=Frame3.destroy).pack(padx=10,pady=10)

Mafenetre.mainloop()

```

### Exemple n°3 : widgets Entry, Label, Button et boîte de dialogue MessageBox

Un script d'authentification :



```

# script mot_de_passe.py
# (C) Fabrice Sincère
from tkinter import *
from tkinter.messagebox import * # boîte de dialogue

```

```

def Verification():
    if Motdepasse.get() == 'python27':
        # le mot de passe est bon : on affiche une boîte de dialogue puis
on ferme la fenêtre
        showinfo('Résultat', 'Mot de passe correct.\nAu revoir !')
        Mafenetre.destroy()
    else:
        # le mot de passe est incorrect : on affiche une boîte de dialogue
        showwarning('Résultat', 'Mot de passe incorrect.\nVeuillez
recommencer !')
        Motdepasse.set('')

# Création de la fenêtre principale (main window)
Mafenetre = Tk()
Mafenetre.title('Identification requise')

# Création d'un widget Label (texte 'Mot de passe')
Labell = Label(Mafenetre, text = 'Mot de passe ')
Labell.pack(side = LEFT, padx = 5, pady = 5)

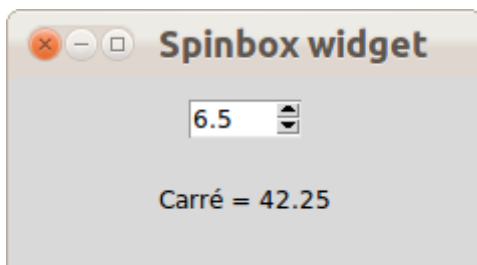
# Création d'un widget Entry (champ de saisie)
Motdepasse= StringVar()
Champ = Entry(Mafenetre, textvariable= Motdepasse, show='*', bg = 'bisque',
fg='maroon')
Champ.focus_set()
Champ.pack(side = LEFT, padx = 5, pady = 5)

# Création d'un widget Button (bouton Valider)
Bouton = Button(Mafenetre, text = 'Valider', command = Verification)
Bouton.pack(side = LEFT, padx = 5, pady = 5)

Mafenetre.mainloop()

```

#### Exemple n°4 : widgets Spinbox et Label



```

# script spinbox.py
#(C) Fabrice Sincère
from tkinter import *

def carre():
    """ Calcul du carré """
    Resultat.set("Carré = "+str(float(Valeur.get())**2))

# Création de la fenêtre principale (main window)
Mafenetre = Tk()
Mafenetre.title("Spinbox widget")

Valeur = StringVar()

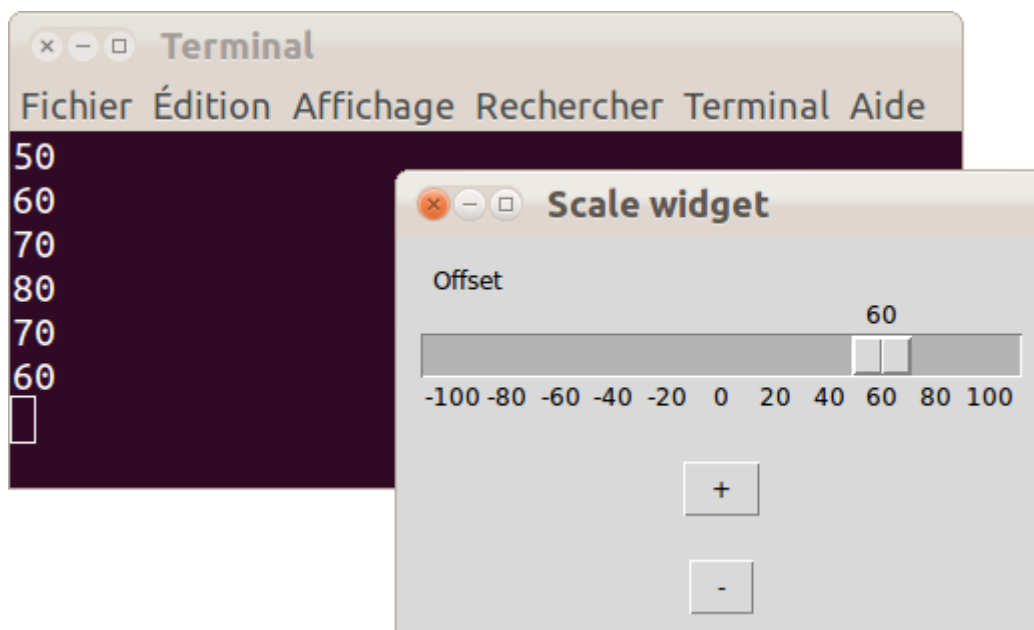
```

```
Valeur.set(2.0)
# Création d'un widget Spinbox
boite = Spinbox(Mafenetre,from_=0,to=10,increment=0.5,textvariable=Valeur,width=5,command=carre)
boite.pack(padx=30,pady=10)

# Création d'un widget Label
Resultat = StringVar()
carre()
Label(Mafenetre,textvariable=Resultat).pack(padx=30,pady=10)

Mafenetre.mainloop()
```

### Exemple n°5 : widgets Scale et Button



```
# script scale.py
#(C) Fabrice Sincère
from tkinter import *

def maj(nouvelleValeur):
    # nouvelle valeur en argument
    print(nouvelleValeur)
def plus():
    Valeur.set(str(int(Valeur.get())+10))
    print(Valeur.get())
def moins():
    Valeur.set(str(int(Valeur.get())-10))
    print(Valeur.get())

# Création de la fenêtre principale (main window)
Mafenetre = Tk()
Mafenetre.title("Scale widget")
```



```
Valeur = StringVar()
Valeur.set(50)
# Création d'un widget Scale
echelle = Scale(Mafenetre,from_=-100,to=100,resolution=10,orient=HORIZONTAL,\
length=300,width=20,label="Offset",tickinterval=20,variable=Valeur,command=maj)
echelle.pack(padx=10,pady=10)

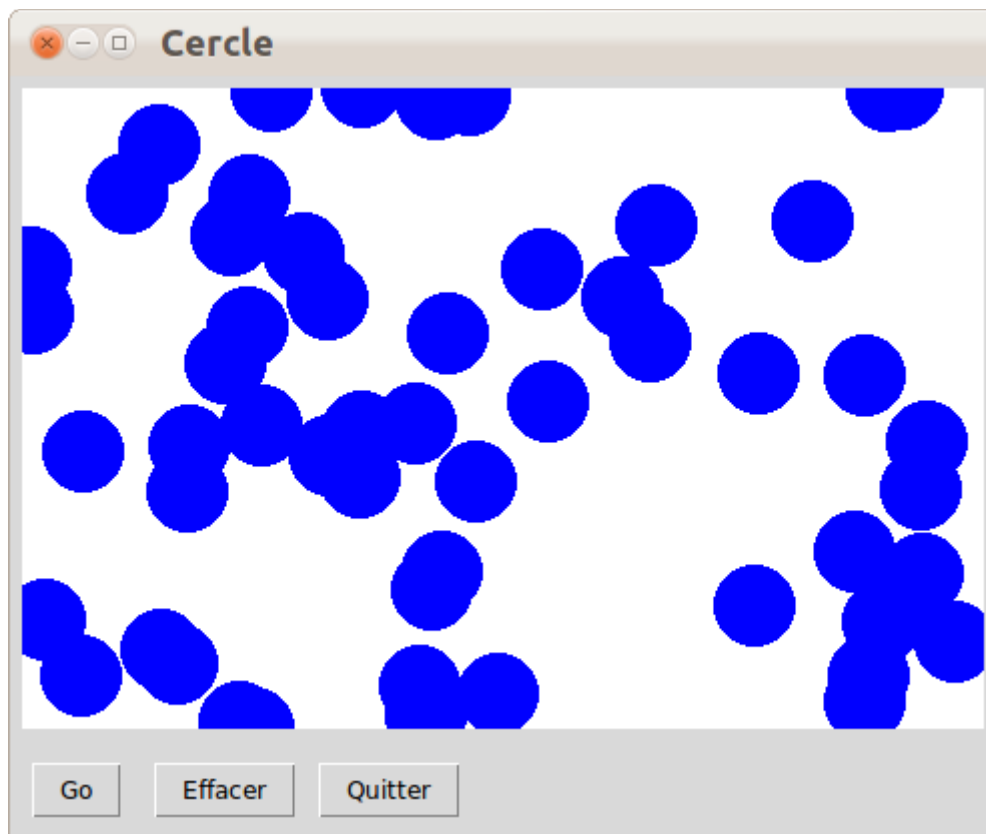
# Création d'un widget Button (bouton +)
Button(Mafenetre,text="+",command=plus).pack(padx=10,pady=10)

# Création d'un widget Button (bouton -)
Button(Mafenetre,text="-",command=moins).pack(padx=10,pady=10)

Mafenetre.mainloop()
```

### Exemple n°6 : widgets Canvas et Button

Le script `cercle.py` dessine, à chaque clic sur le bouton Go, un disque de rayon 20 pixels à une position aléatoire :



```
# script cercle.py
#(C) Fabrice Sincère
from tkinter import *
import random

def Cercle():
    """ Dessine un cercle de centre (x,y) et de rayon r """
```

```

x = random.randint(0,Largeur)
y = random.randint(0,Hauteur)
r = 20
Canevas.create_oval(x-r, y-r, x+r, y+r, outline='blue', fill='blue')

def Effacer():
    """ Efface la zone graphique """
    Canevas.delete(ALL)

# Création de la fenêtre principale (main window)
Mafenetre = Tk()
Mafenetre.title('Cercle')

# Création d'un widget Canvas (zone graphique)
Largeur = 480
Hauteur = 320
Canevas = Canvas(Mafenetre, width = Largeur, height =Hauteur, bg = 'white')
Canevas.pack(packx =5, packy =5)

# Création d'un widget Button (bouton Go)
BoutonGo = Button(Mafenetre, text = 'Go', command = Cercle)
BoutonGo.pack(side = LEFT, padx = 10, pady = 10)

# Création d'un widget Button (bouton Effacer)
BoutonEffacer = Button(Mafenetre, text = 'Effacer', command = Effacer)
BoutonEffacer.pack(side = LEFT, padx = 5, pady = 5)

# Création d'un widget Button (bouton Quitter)
BoutonQuitter = Button(Mafenetre, text = 'Quitter', command =
Mafenetre.destroy)
BoutonQuitter.pack(side = LEFT, padx = 5, pady = 5)

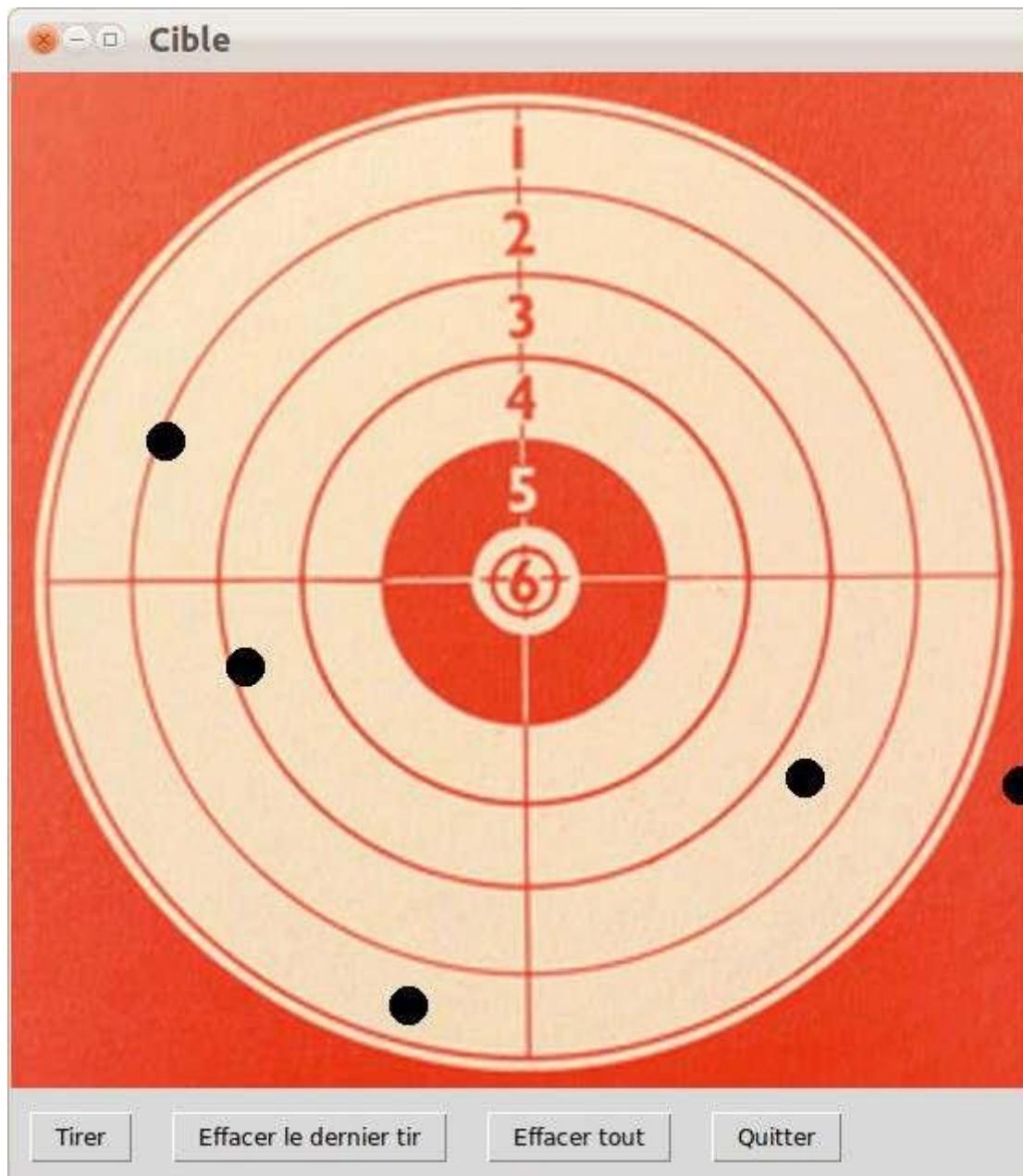
Mafenetre.mainloop()

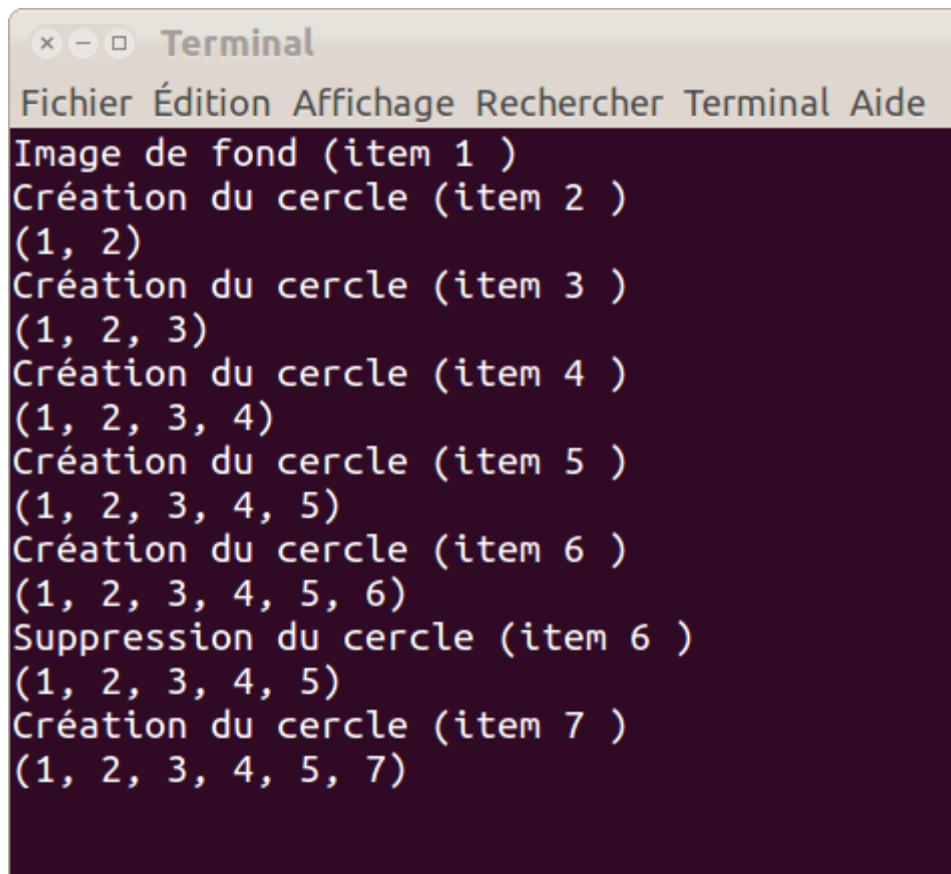
```

Ce code est détaillé [ici](#).

### Exemple n°7 : widgets Canvas et Button ; gestion des images

Ce script reprend le script `cercle.py` avec une image de fond (méthode `create_image()` de la classe `Canvas`) et la possibilité d'effacer la dernière action (pour cela, on se sert du numéro identifiant de chaque item d'un widget `Canvas`) :





```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
Image de fond (item 1 )
Création du cercle (item 2 )
(1, 2)
Création du cercle (item 3 )
(1, 2, 3)
Création du cercle (item 4 )
(1, 2, 3, 4)
Création du cercle (item 5 )
(1, 2, 3, 4, 5)
Création du cercle (item 6 )
(1, 2, 3, 4, 5, 6)
Suppression du cercle (item 6 )
(1, 2, 3, 4, 5)
Création du cercle (item 7 )
(1, 2, 3, 4, 5, 7)
```

```
# script cible.py
#(C) Fabrice Sincère
from tkinter import *
import random

def Cercle():
    """ Dessine un cercle de centre (x,y) et de rayon r """
    x = random.randint(0, Largeur)
    y = random.randint(0, Hauteur)
    r = 10

    # on dessine un cercle dans la zone graphique
    item = Canevas.create_oval(x-r, y-r, x+r, y+r, outline='black', fill='black')

    print("Création du cercle (item" , item , ")")
    # affichage de tous les items de Canevas
    print(Canevas.find_all())

def Undo():
    """ Efface le dernier cercle """
    if len(Canevas.find_all()) > 1:
        item = Canevas.find_all()[-1]
        # on efface le cercle
        Canevas.delete(item)
```

```

    print("Suppression du cercle (item" , item ,")")
    # affichage de tous les items de Canevas
    print(Canevas.find_all())

def EffacerTout():
    """ Efface tous les cercles """
    while len(Canevas.find_all()) > 1:
        Undo()

# Création de la fenêtre principale (main window)
Mafenetre = Tk()
Mafenetre.title('Cible')

# Image de fond
photo = PhotoImage(file="tk_cible.gif")

# Création d'un widget Canvas (zone graphique)
Largeur = 550
Hauteur = 550
Canevas = Canvas(Mafenetre,width = Largeur, height =Hauteur)
item = Canevas.create_image(0,0,anchor=NW, image=photo)
print("Image de fond (item",item,")")
Canevas.pack()

# Création d'un widget Button
BoutonGo = Button(Mafenetre, text ='Tirer', command = Cercle)
BoutonGo.pack(side = LEFT, padx = 10, pady = 10)

# Création d'un widget Button
BoutonEffacer = Button(Mafenetre, text ='Effacer le dernier tir', command = Undo)
BoutonEffacer.pack(side = LEFT, padx = 10, pady = 10)

# Création d'un widget Button
BoutonEffacerTout = Button(Mafenetre, text ='Effacer tout', command = EffacerTout)
BoutonEffacerTout.pack(side = LEFT, padx = 10, pady = 10)

# Création d'un widget Button (bouton Quitter)
BoutonQuitter = Button(Mafenetre, text ='Quitter', command = Mafenetre.destroy)
BoutonQuitter.pack(side = LEFT, padx = 10, pady = 10)

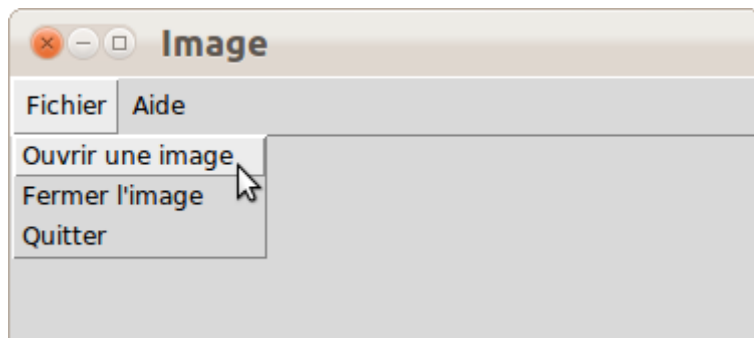
Mafenetre.mainloop()

```

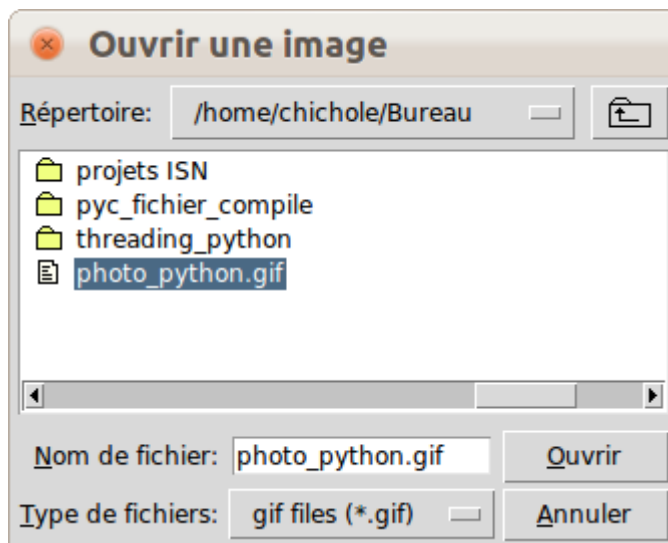
Télécharger l'image de fond [tk\\_cible.gif](#)

#### **Exemple n°8 : widgets Menu et Canvas ; gestion des images ; boîtes de dialogue `FileDialog` et `MessageBox`**

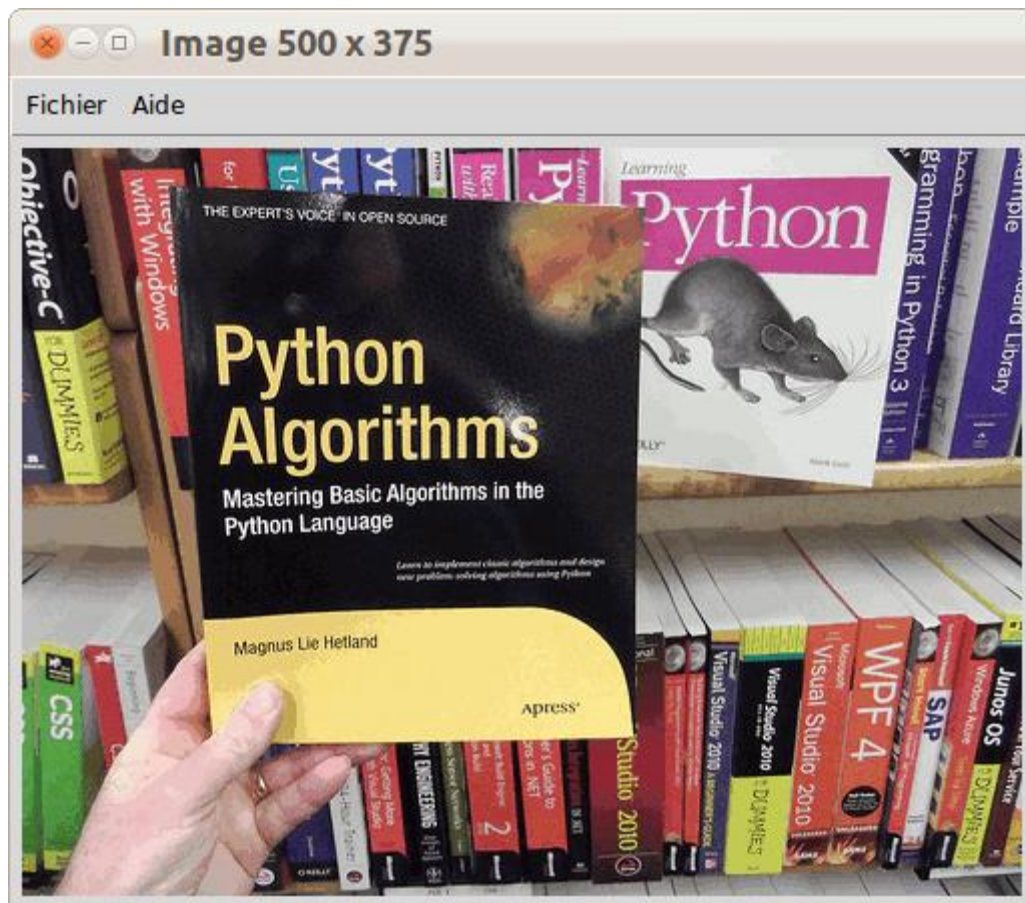
Le script suivant est un browser d'images (formats .gif .ppm .pgm), avec un widget `Menu` :



une boîte de dialogue `FileDialog` pour rechercher un fichier :



et un widget `Canvas` dans lequel sera affiché l'image :



```
# script lecture_gif.py
#(C) Fabrice Sincère
from tkinter import *
import tkinter.messagebox
import tkinter.filedialog

def Ouvrir():
    Canvas.delete(ALL) # on efface la zone graphique

    filename = tkinter.filedialog.askopenfilename(title="Ouvrir une image",filetypes=[('gif files', '.gif'),('all files','.*')])
    print(filename)

    photo = PhotoImage(file=filename)
    gifdict[filename] = photo # référence
    print(gifdict)

    Canvas.create_image(0,0,anchor=NW,image=photo)
    Canvas.config(height=photo.height(),width=photo.width())

    Mafenetre.title("Image "+str(photo.width())+" x "+str(photo.height()))

def Fermer():
    Canvas.delete(ALL)
    Mafenetre.title("Image")
```



```

def Apropos():
    tkinter.messagebox.showinfo("A propos","Tutorial Python Tkinter\n(C) Fabrice Sincère")

# Main window
Mafenetre = Tk()
Mafenetre.title("Image")

# Création d'un widget Menu
menubar = Menu(Mafenetre)

menufichier = Menu(menubar,tearoff=0)
menufichier.add_command(label="Ouvrir une image",command=Ouvrir)
menufichier.add_command(label="Fermer l'image",command=Fermer)
menufichier.add_command(label="Quitter",command=Mafenetre.destroy)
menubar.add_cascade(label="Fichier", menu=menufichier)

menuaide = Menu(menubar,tearoff=0)
menuaide.add_command(label="A propos",command=Apropos)
menubar.add_cascade(label="Aide", menu=menuaide)

# Affichage du menu
Mafenetre.config(menu=menubar)

# Création d'un widget Canvas
Canevas = Canvas(Mafenetre)
Canevas.pack(padx=5,pady=5)

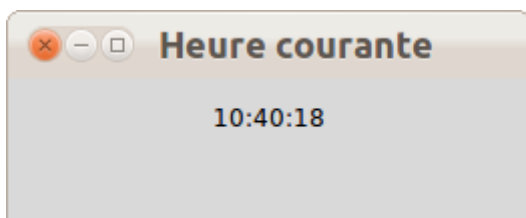
# Utilisation d'un dictionnaire pour conserver une référence
gifdict={ }

Mafenetre.mainloop()

```

### Exemple n°9 : gestion du temps

L'heure courante est mise à jour toutes les secondes :



Pour cela, on utilise la méthode `after()` qui appelle une fonction après une durée donnée en millisecondes :

```

# script heure.py
#(C) Fabrice Sincère
from tkinter import *

```



```

import time

def maj():
    # on arrive ici toutes les 1000 ms
    heure.set(time.strftime('%H:%M:%S'))
    Mafenetre.after(1000,maj)

Mafenetre = Tk()
Mafenetre.title("Heure courante")

# Création d'un widget Label
heure = StringVar()
Label(Mafenetre,textvariable=heure).pack(padx=10,pady=10)

maj()

Mafenetre.mainloop()

```

#### Exemple n°10 : widgets Canvas et Button ; gestion du temps

Le script `animation.py` est un exemple d'animation (affichage d'environ 20 disques par seconde) :



On se sert de la méthode `after()` pour actualiser la zone graphique toutes les 50 ms :

```

# script animation.py
#(C) Fabrice Sincère
from tkinter import *
import random

def Cercle():
    """ Dessine un cercle de centre (x,y) et de rayon r """
    global Arret
    x = random.randint(0, Largeur)
    y = random.randint(0, Hauteur)
    r = 10
    Canevas.create_oval(x-r, y-r, x+r, y+r, outline='red', fill='red')
    if Arret == False:
        # appel de la fonction Cercle() après une pause de 50 millisecondes
        Mafenetre.after(50, Cercle)

def Arrêter():
    """ Arrêt de l'animation """
    global Arret
    Arret = True

def Démarrer():
    """ Démarre l'animation """
    global Arret
    Canevas.delete(ALL)
    if Arret == True:
        Arret = False
        Cercle() # un seul appel à cette fonction

Arret = True

# Création de la fenêtre principale (main window)
Mafenetre = Tk()
Mafenetre.title('Animation')

# Création d'un widget Canvas
Largeur = 480
Hauteur = 320
Canevas = Canvas(Mafenetre, width = Largeur, height = Hauteur, bg = 'white')
Canevas.pack(padx = 5, pady = 5)

# Création d'un widget Button (bouton Démarrer)
BoutonGo = Button(Mafenetre, text = 'Démarrer', command = Démarrer)
BoutonGo.pack(side = LEFT, padx = 10, pady = 10)

# Création d'un widget Button (bouton Arrêter)
BoutonArrêter = Button(Mafenetre, text = 'Arrêter', command = Arrêter)
BoutonArrêter.pack(side = LEFT, padx = 5, pady = 5)

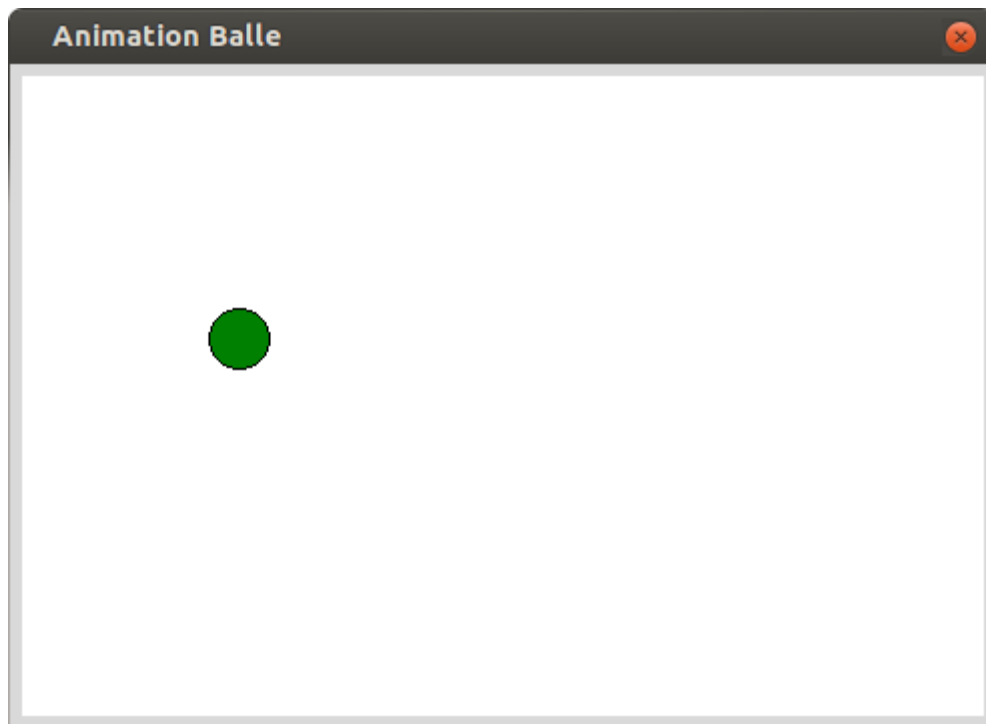
# Création d'un widget Button (bouton Quitter)
BoutonQuitter = Button(Mafenetre, text = 'Quitter', command =
Mafenetre.destroy)
BoutonQuitter.pack(side = LEFT, padx = 5, pady = 5)

Mafenetre.mainloop()

```

### Exemple n°11 : widget Canvas ; gestion du temps

Le script `animation_balle.py` est une animation qui gère le déplacement d'une balle et de ses rebonds sur les bords :



```
# script animation_balle.py
#(C) Fabrice Sincère

from tkinter import *
import math,random

LARGEUR = 480
HAUTEUR = 320
RAYON = 15 # rayon de la balle

# position initiale au milieu
X = LARGEUR/2
Y = HAUTEUR/2

# direction initiale aléatoire
vitesse = random.uniform(1.8,2)*5
angle = random.uniform(0,2*math.pi)
DX = vitesse*math.cos(angle)
DY = vitesse*math.sin(angle)

def deplacement():
    """ Déplacement de la balle """
    global X,Y,DX,DY,RAYON,LARGEUR,HAUTEUR

    # rebond à droite
    if X+RAYON+DX > LARGEUR:
        X = 2*(LARGEUR-RAYON)-X
        DX = -DX
```

```

# rebond à gauche
if X-RAYON+DX < 0:
    X = 2*RAYON-X
    DX = -DX

# rebond en bas
if Y+RAYON+DY > HAUTEUR:
    Y = 2*(HAUTEUR-RAYON)-Y
    DY = -DY

# rebond en haut
if Y-RAYON+DY < 0:
    Y = 2*RAYON-Y
    DY = -DY

X = X+DX
Y = Y+DY

# affichage
Canevas.coords(Balle,X-RAYON,Y-RAYON,X+RAYON,Y+RAYON)

# mise à jour toutes les 50 ms
Mafenetre.after(50,deplacement)

# Création de la fenêtre principale
Mafenetre = Tk()
Mafenetre.title("Animation Balle")

# Création d'un widget Canvas
Canevas = Canvas(Mafenetre,height=HAUTEUR,width=LARGEUR,bg='white')
Canevas.pack(padx=5,pady=5)

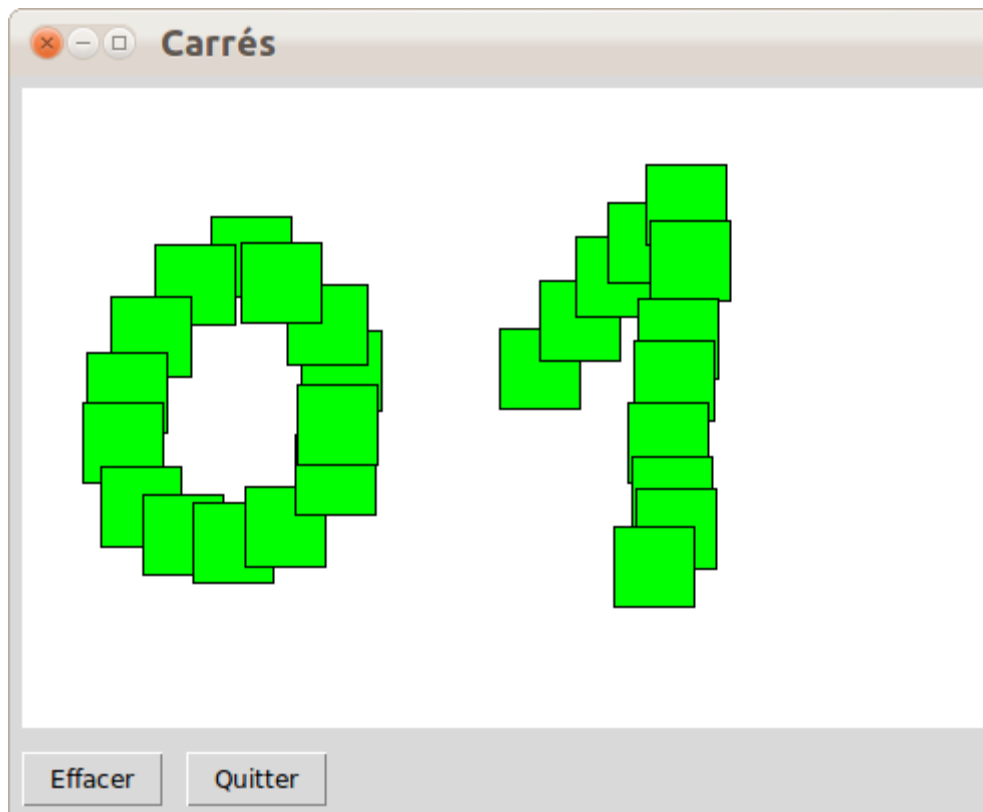
# Création d'un objet graphique
Balle = Canevas.create_oval(X-RAYON,Y-
RAYON,X+RAYON,Y+RAYON,width=1,fill='green')

deplacement()
Mafenetre.mainloop()

```

### Exemple n°12 : widgets Canvas et Button ; gestion de la souris

Le script `carre.py` dessine un carré à l'endroit du clic de la souris.  
Pour cela, on utilise l'événement associé au clic gauche de la souris.



```
# script carre.py
#(C) Fabrice Sincère
from tkinter import *

def Clic(event):
    """ Gestion de l'événement Clic gauche sur la zone graphique """
    # position du pointeur de la souris
    X = event.x
    Y = event.y
    # on dessine un carré
    r = 20
    Canvas.create_rectangle(X-r, Y-r, X+r, Y+r,
outline='black',fill='green')

def Effacer():
    """ Efface la zone graphique """
    Canvas.delete(ALL)

# Création de la fenêtre principale
Mafenetre = Tk()
Mafenetre.title('Carrés')

# Création d'un widget Canvas
Largeur = 480
Hauteur = 320
Canvas = Canvas(Mafenetre, width = Largeur, height =Hauteur, bg ='white')
# La méthode bind() permet de lier un événement avec une fonction :
# un clic gauche sur la zone graphique provoquera l'appel de la fonction
utilisateur Clic()
Canvas.bind('<Button-1>', Clic)
Canvas.pack(padx =5, pady =5)

# Création d'un widget Button (bouton Effacer)
```

```

Button(Mafenetre, text = 'Effacer', command = Effacer).pack(side=LEFT, padx =
5, pady = 5)

# Création d'un widget Button (bouton Quitter)
Button(Mafenetre, text = 'Quitter', command =
Mafenetre.destroy).pack(side=LEFT, padx=5, pady=5)

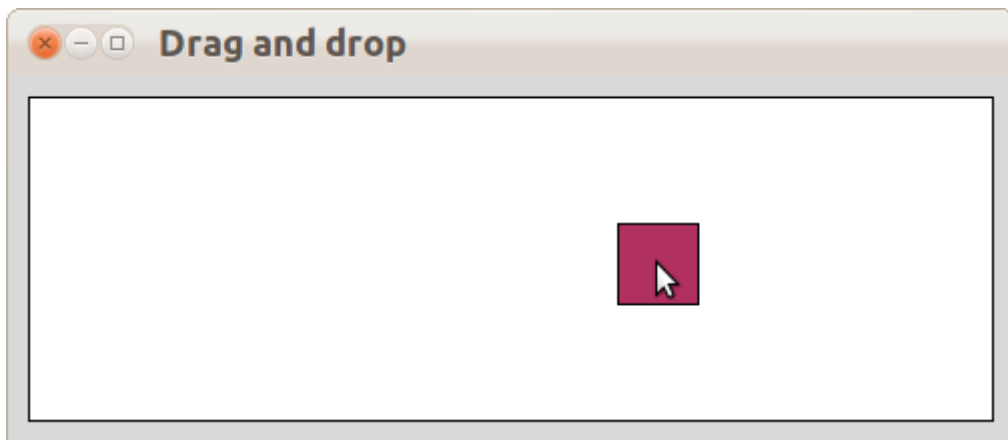
Mafenetre.mainloop()

```

Ce code est détaillé [ici](#).

### Exemple n°13 : widget Canvas ; gestion de la souris

Nous allons voir comment déplacer un objet graphique avec la souris (clic, drag and drop) :



```

# script drag_and_drop.py
#(C) Fabrice Sincère
from tkinter import *

def Clic(event):
    """ Gestion de l'événement Clic gauche """
    global DETECTION_CLIC_SUR_OBJET

    # position du pointeur de la souris
    X = event.x
    Y = event.y
    print("Position du clic -> ",X,Y)

    # coordonnées de l'objet
    [xmin,ymin,xmax,ymax] = Canevas.coords(Carre)

    print("Position objet -> ",xmin,ymin,xmax,ymax)
    if xmin<=X<=xmax and ymin<=Y<=ymax: DETECTION_CLIC_SUR_OBJET = True
    else: DETECTION_CLIC_SUR_OBJET = False
    print("DETECTION CLIC SUR OBJET -> ",DETECTION_CLIC_SUR_OBJET)

def Drag(event):
    """ Gestion de l'événement bouton gauche enfoncé """

```

```

X = event.x
Y = event.y
print("Position du pointeur -> ",X,Y)

if DETECTION_CLIC_SUR_OBJET == True:
    # limite de l'objet dans la zone graphique
    if X<0: X=0
    if X>Largeur: X=Largeur
    if Y<0: Y=0
    if Y>Hauteur: Y=Hauteur
    # mise à jour de la position de l'objet (drag)
    Canevas.coords(Carre,X-TailleCarre,Y-TailleCarre,X+TailleCarre,Y+TailleCarre)

DETECTION_CLIC_SUR_OBJET = False

# Création de la fenêtre principale
Mafenetre = Tk()
Mafenetre.title("Drag and drop")

# Création d'un widget Canvas
Largeur = 480
Hauteur = 160
TailleCarre = 20
Canevas = Canvas(Mafenetre,width=Largeur,height=Hauteur,bg='white')
# Création d'un objet graphique
Carre = Canevas.create_rectangle(0,0,TailleCarre*2,TailleCarre*2,fill='maroon')

# La méthode bind() permet de lier un événement avec une fonction
Canevas.bind('<Button-1>',Clic) # événement clic gauche (press)
Canevas.bind('<B1-Motion>',Drag) # événement bouton gauche enfoncé (hold down)

Canevas.focus_set()
Canevas.pack(padx=10,pady=10)

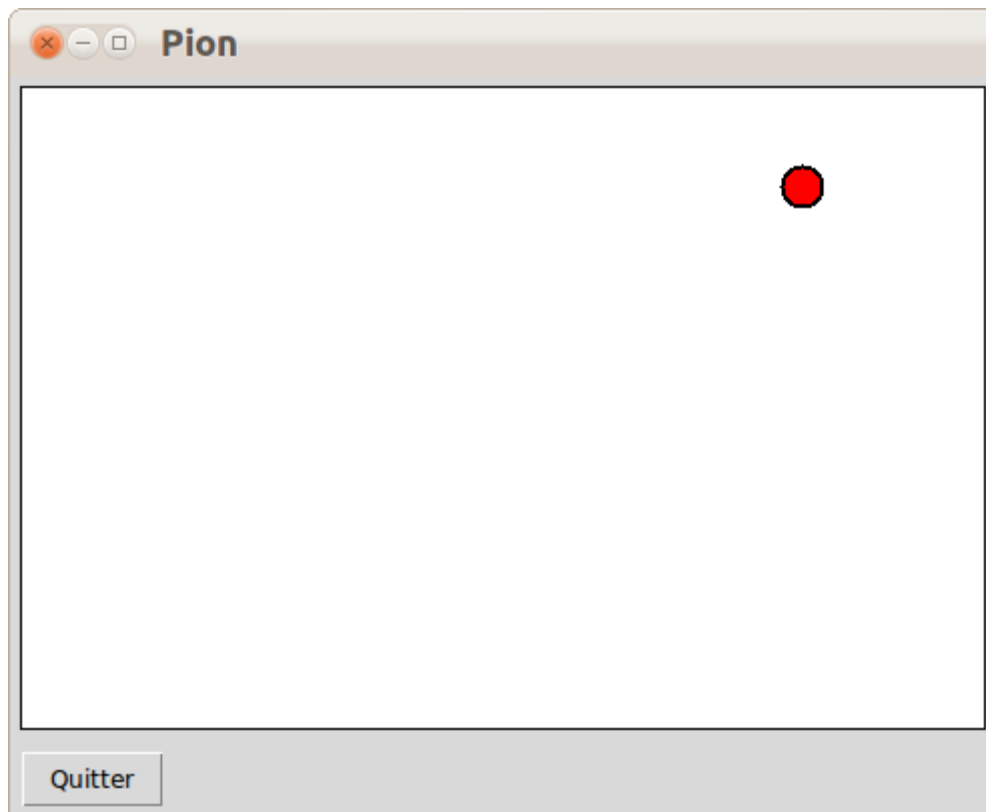
Mafenetre.mainloop()

```

#### Exemple n°14 : widgets Canvas et Button ; gestion du clavier

Le script `pion.py` gère le déplacement d'un pion avec le clavier.  
 Pour se faire, on utilise l'événement associé à l'appui d'une touche du clavier.

- touche a déplacement vers le haut
- touche q déplacement vers le bas
- touche l déplacement vers la gauche
- touche m déplacement vers la droite



```
# script pion.py
#(C) Fabrice Sincère
from tkinter import *

def Clavier(event):
    """ Gestion de l'événement Appui sur une touche du clavier """
    global PosX,PosY
    touche = event.keysym
    print(touche)
    # déplacement vers le haut
    if touche == 'a':
        PosY -= 20
    # déplacement vers le bas
    if touche == 'q':
        PosY += 20
    # déplacement vers la droite
    if touche == 'm':
        PosX += 20
    # déplacement vers la gauche
    if touche == 'l':
        PosX -= 20
    # on dessine le pion à sa nouvelle position
    Canvas.coords(Pion,PosX -10, PosY -10, PosX +10, PosY +10)

# Création de la fenêtre principale
Mafenetre = Tk()
Mafenetre.title('Pion')

# position initiale du pion
PosX = 230
PosY = 150

# Création d'un widget Canvas (zone graphique)
Largeur = 480
```



```

Hauteur = 320
Canevas = Canvas(Mafenetre, width = Largeur, height =Hauteur, bg ='white')
Pion = Canevas.create_oval(PosX-10,PosY-10,PosX+10,PosY+10,width=2,outline='black',fill='red')
Canevas.focus_set()
Canevas.bind('<Key>',Clavier)
Canevas.pack(padx =5, pady =5)

# Création d'un widget Button (bouton Quitter)
Button(Mafenetre, text ='Quitter', command =
Mafenetre.destroy).pack(side=LEFT,padx=5,pady=5)

Mafenetre.mainloop()

```

Ce code est détaillé [ici](#).

### *Symboles des quelques touches spéciales*

'Up', 'Down', 'Left', 'Right' (flèches directionnelles haut, bas, gauche, droite), 'Return' (touche Entrée), 'space' (barre Espace)...

### **Exemple n°15 : widgets Checkbutton et Button ; musiques et sons avec pygame**



Le module `pygame` est un module externe de création de jeux vidéo en 2D.

`pygame` contient un sous module `pygame.mixer` qui permet de charger et de lire des musiques ou des sons dans plusieurs formats (mp3, ogg, wav...).

La procédure d'installation de `pygame` se trouve [ici](#).

```

# script sons_pygame.py
#(C) Fabrice Sincère
# python version 3.2
# pygame version 1.9.2
from tkinter import *
import pygame

pygame.mixer.init()
pygame.mixer.music.load("chavmusic7.mp3")
# réglage volume
pygame.mixer.music.set_volume(0.3)

son1 = pygame.mixer.Sound("balla1.ogg")
son1.set_volume(0.5)
son2 = pygame.mixer.Sound("death1.wav")
son2.set_volume(1.0)

def PlaySon1():

```

```

    son1.play()
def PlaySon2():
    son2.play()
def Music():
    print(musique.get())
    if musique.get() == 1:
        # 1 (ON)
        # joue en boucle
        pygame.mixer.music.play(-1)
    else:
        # 0 (OFF)
        pygame.mixer.music.stop()

# Création de la fenêtre principale (main window)
Mafenetre = Tk()
Mafenetre.title("Checkbutton widget + Pygame.mixer")

# Création d'un widget Button
Button(Mafenetre,text="son 1",command=PlaySon1).pack(side=LEFT,padx=10,pady=10)

# Création d'un widget Button
Button(Mafenetre,text="son 2",command=PlaySon2).pack(side=LEFT,padx=10,pady=10)

# Création d'un widget Checkbutton
musique = IntVar()
musique.set(1) # ON
Checkbutton(Mafenetre,text="Musique de fond",variable=musique,command=Music).pack(side=LEFT,padx=10,pady=10)

Music()
Mafenetre.mainloop()

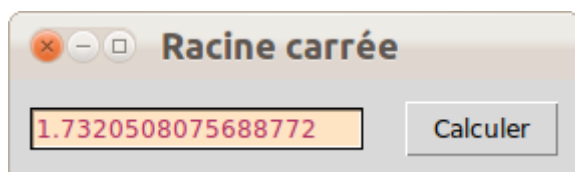
Télécharger le son death1.wav
Télécharger le son balla1.ogg
Télécharger la musique chavmusic7.mp3

```

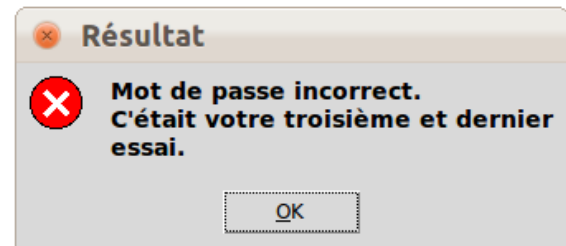
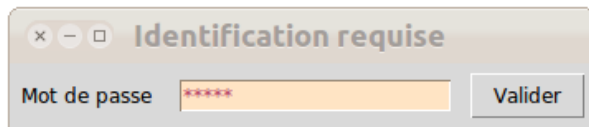
## Exercices

**Exercice 7.1 ★** En s'inspirant des scripts `de.py` et `mot_de_passe.py`, écrire une application avec interface graphique qui calcule la racine carrée d'un nombre.

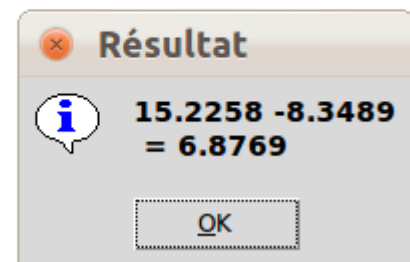
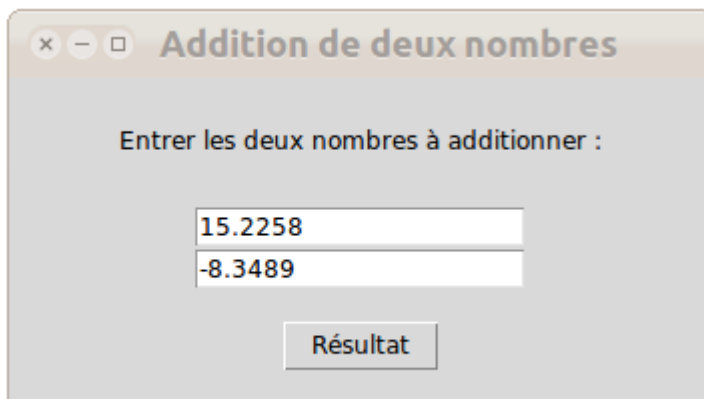
Par exemple, le calcul de  $\sqrt{3}$  donne :



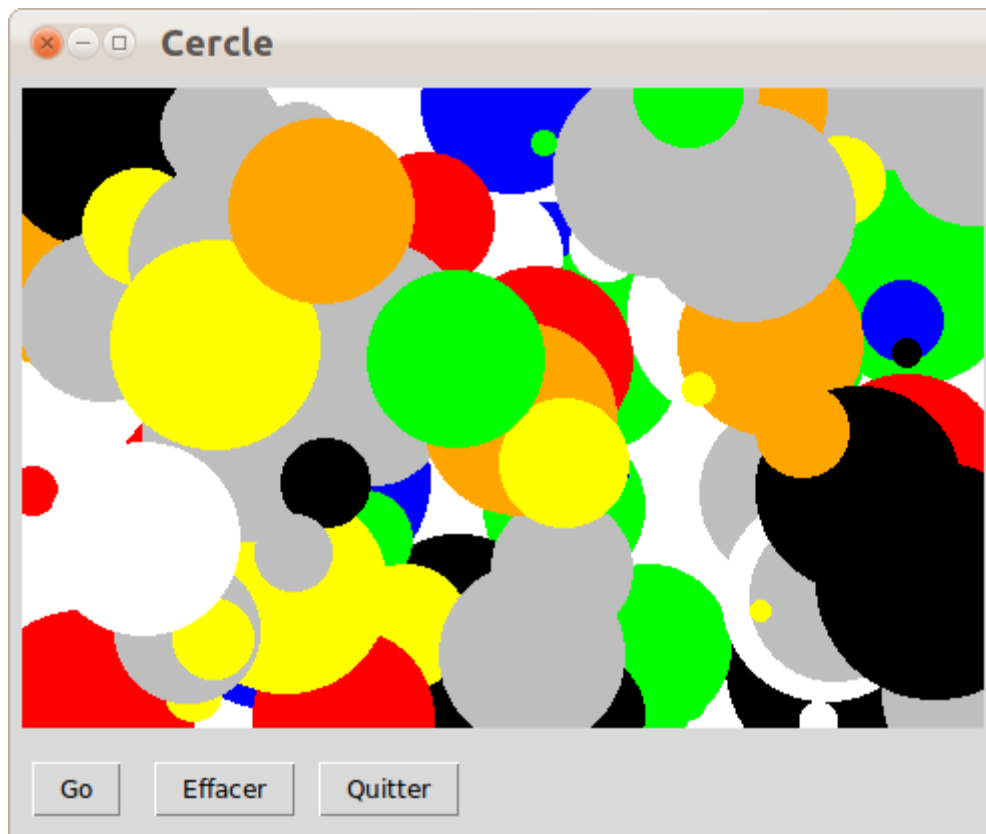
**Exercice 7.2 ★** Reprendre le script `mot_de_passe.py` de manière à limiter le nombre d'essais à trois.



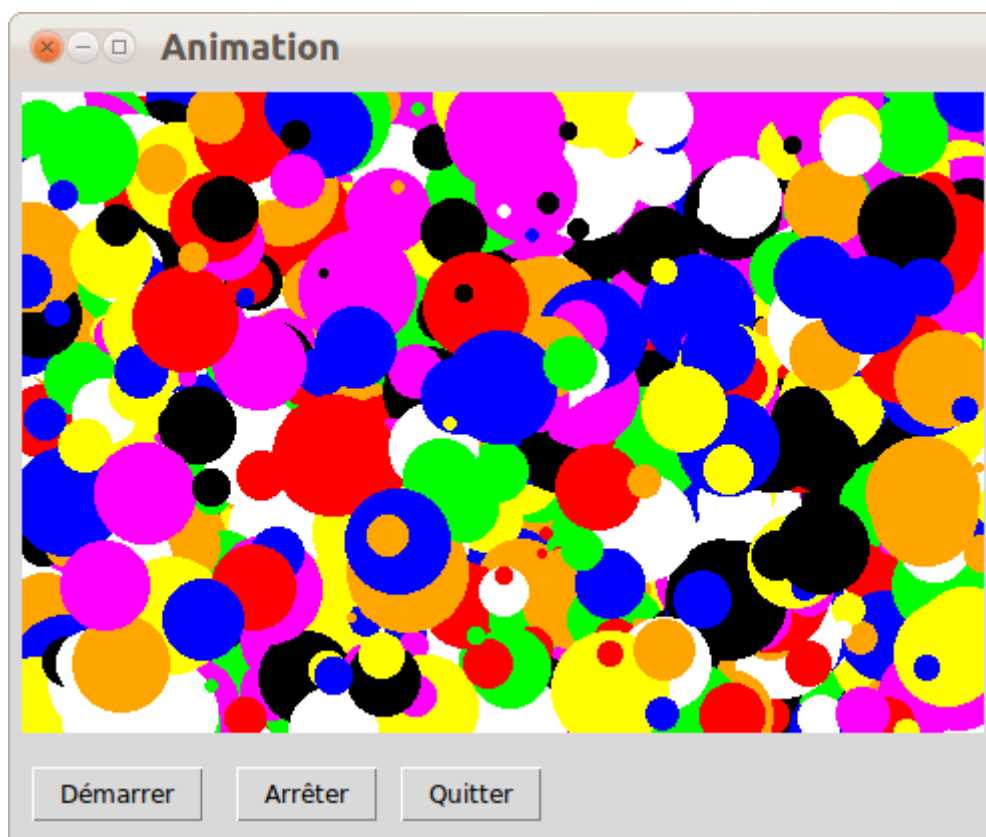
**Exercice 7.3 ★** En s'inspirant des scripts `de.py` et `mot_de_passe.py`, écrire une application avec interface graphique qui calcule l'addition ou la soustraction de deux nombres :



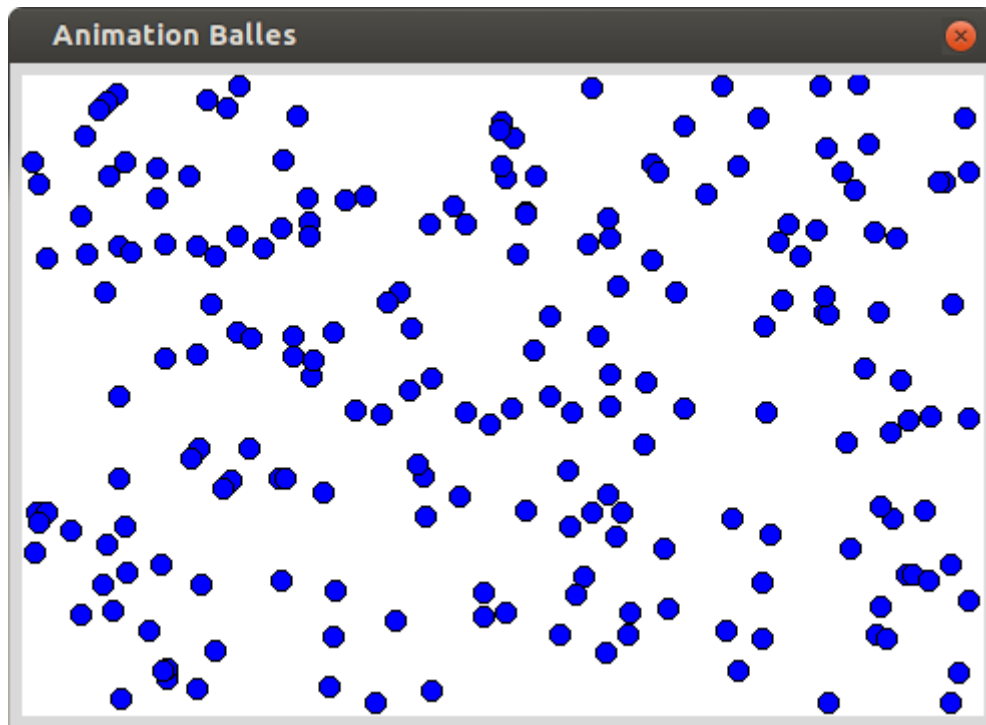
**Exercice 7.4 ★** A partir du script `cercle.py`, dessiner des disques de positions, rayons et couleurs aléatoires :



**Exercice 7.5** ★ A partir du script `animation.py`, faire une animation avec des disques de positions, rayons et couleurs aléatoires.



**Exercice 7.6 ★★** A partir du script `animation_balle.py`, faire une animation qui gère la trajectoire d'un nombre quelconque de balles :



Remarque : on ne tiendra pas compte des chocs entre balles.

**Exercice 7.7 ★★**

1) Reprendre le script `cible.py` et remplacer le disque noir par une image :



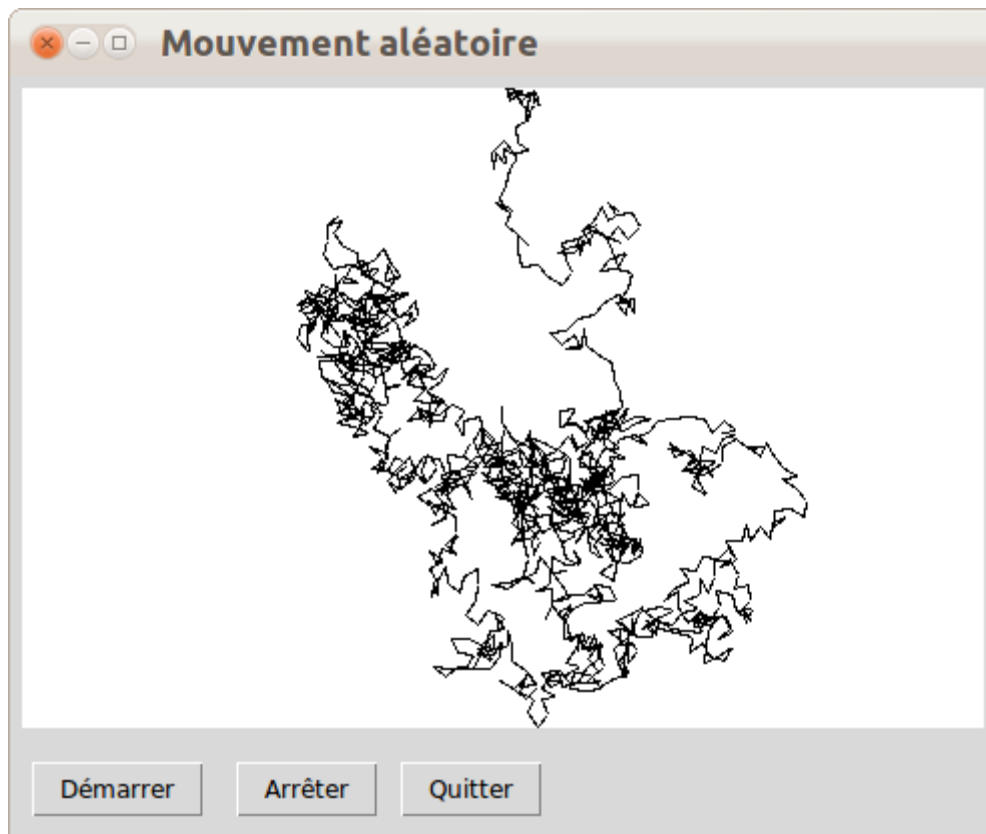
Télécharger l'image [impact.gif](#)

Remarque : l'image de l'impact doit avoir un fond transparent.

2) En s'inspirant du script `sons_pygame.py`, ajouter un effet sonore ([tk\\_coup\\_fusil.wav](#)).

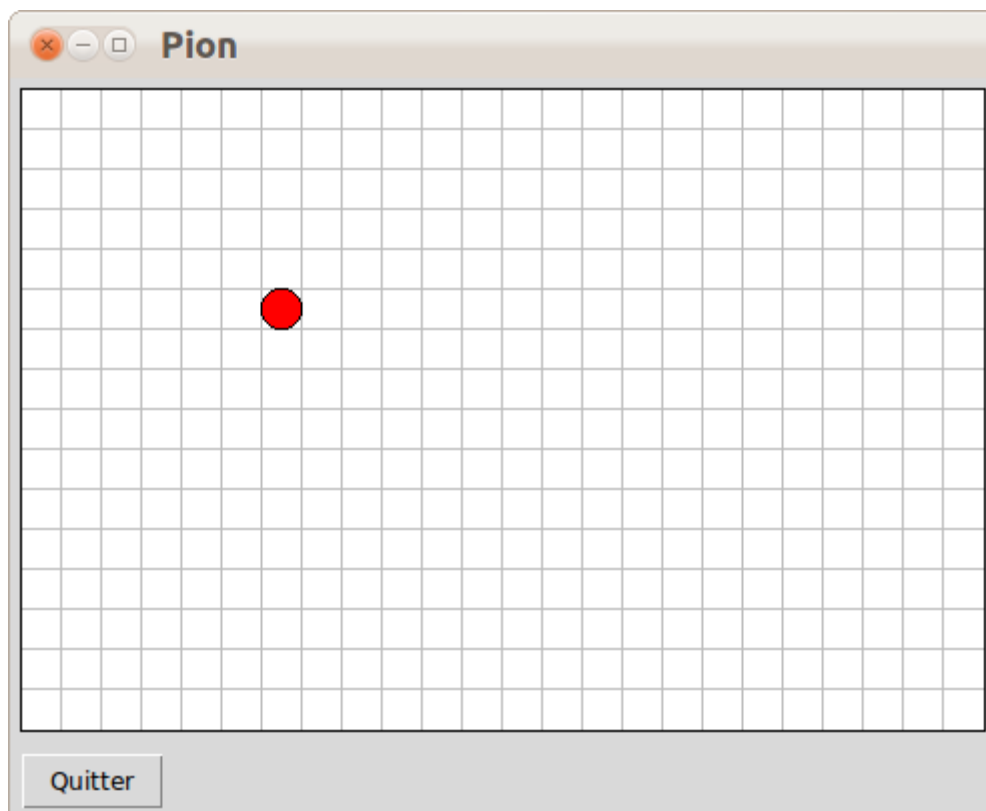
**Exercice 7.8 ★★** En s'inspirant du script `animation.py`, faire l'animation d'un mouvement aléatoire brownien.

On utilisera la méthode `create_line()` de la classe `Canvas`.



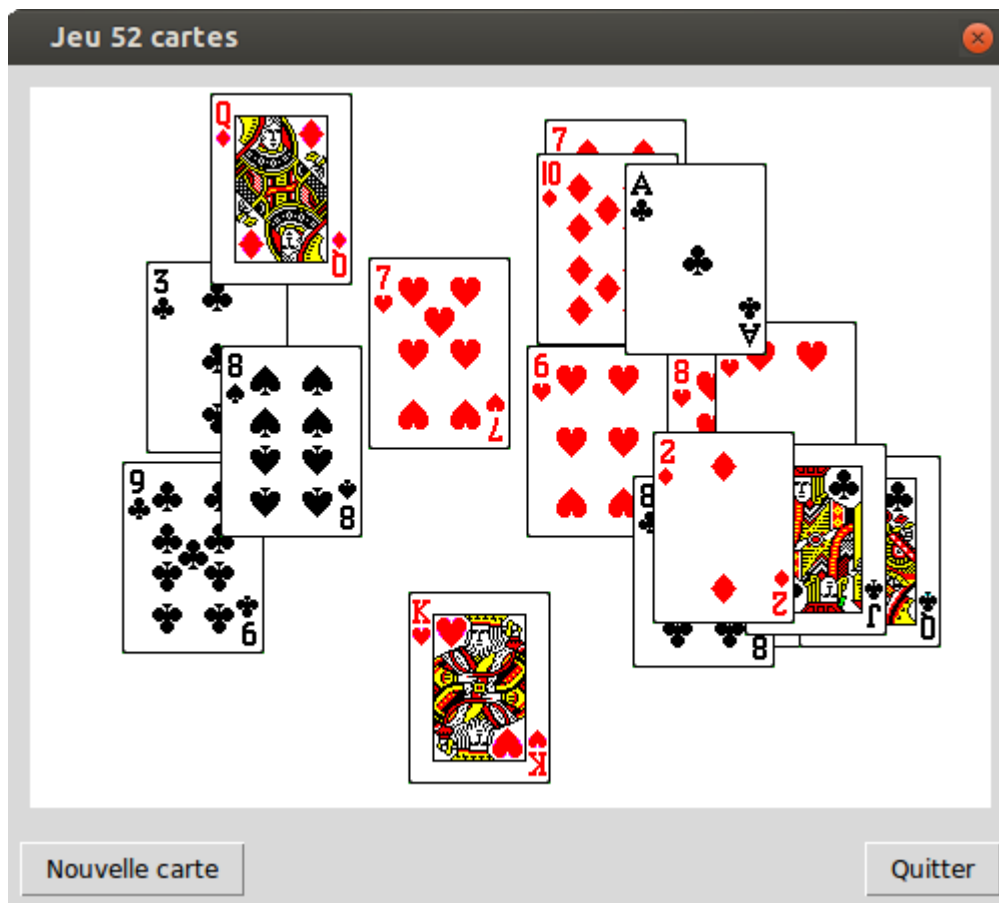
**Exercice 7.9 ★** Compléter le script `pion.py` de manière à dessiner une grille.

On utilisera la méthode `create_line()` de la classe `Canvas`.



### Exercice 7.10 ★★

Ecrire un script qui tire une carte et l'affiche à une position aléatoire.



Il faut se servir d'une référence comme dans le script `lecture_gif.py`

Télécharger les images des 52 cartes et un squelette du script [ici](#).

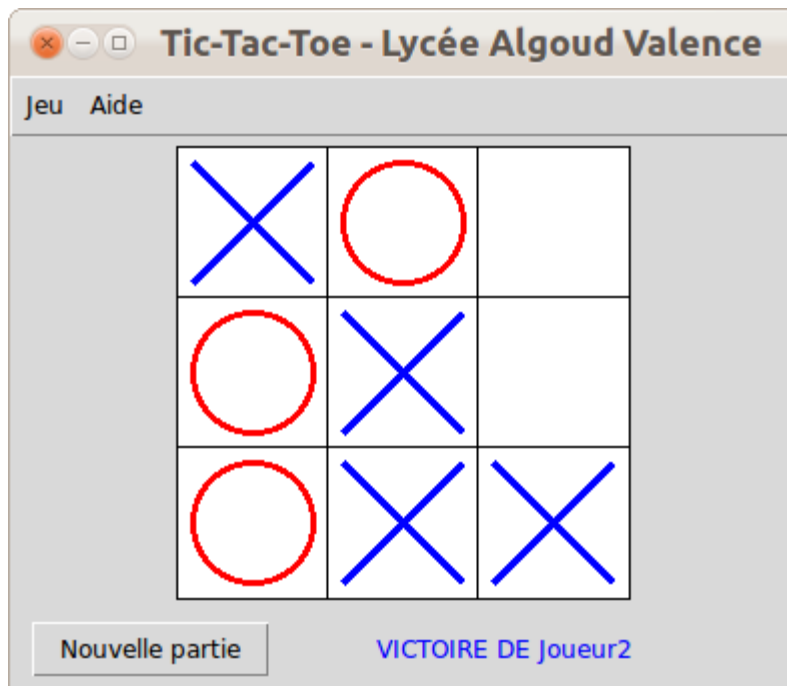
### Quelques idées de projets

#### Projet n°1 ★★★★★ Jeu Tic-Tac-Toe (jeu du morpion)

Un projet relativement simple pour un travail en binôme.

Le fichier exécutable est téléchargeable [ici](#).





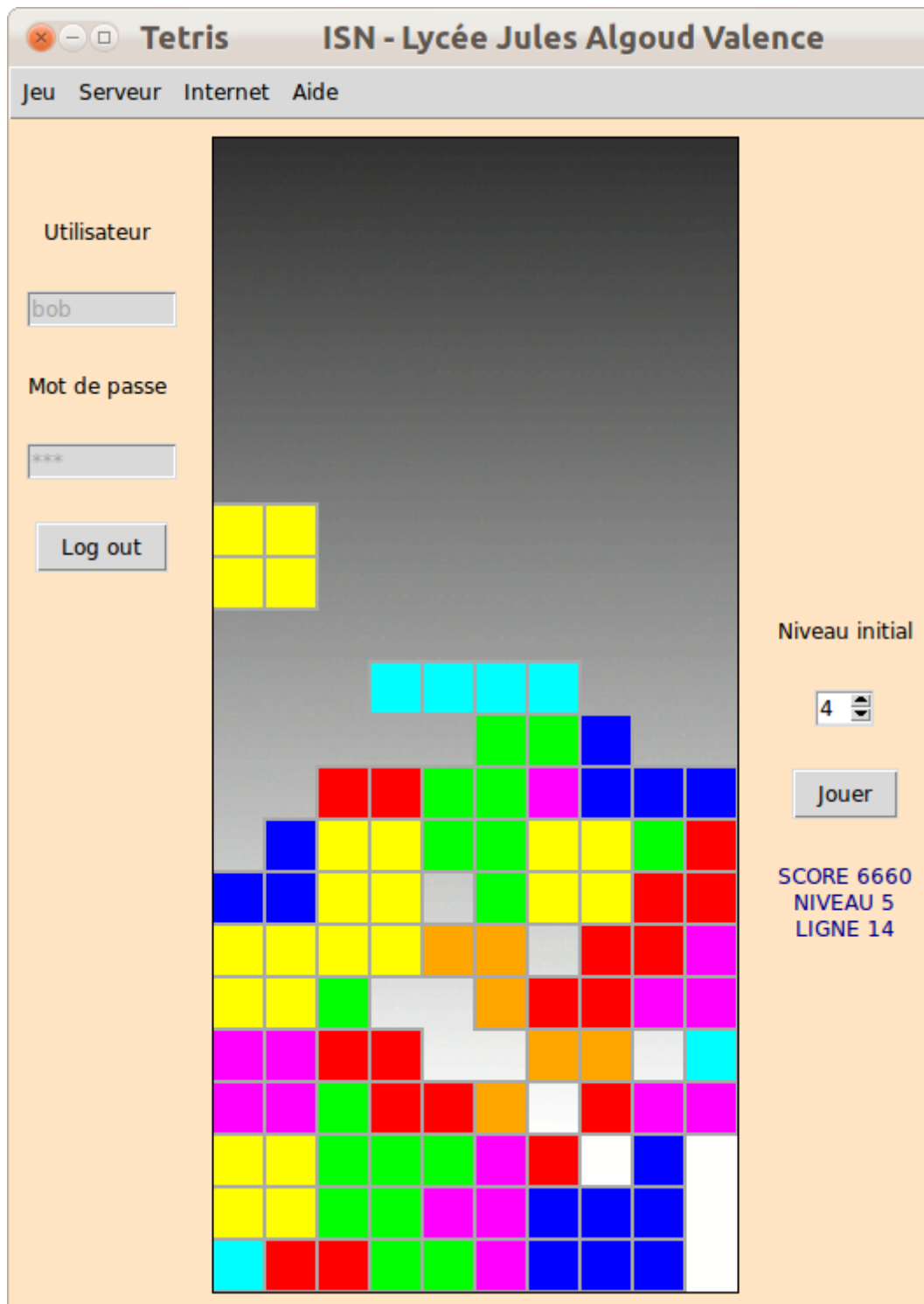
Remarque : dans ma version, le code source Python fait une bonne centaine de lignes.

On pourra compléter ce projet par une version en réseau (plus d'informations [ici](#)).

## Projet n°2 ★★★★★ Jeu de Tetris avec classement en ligne

Un gros projet à décomposer en plusieurs tâches :

- jeu en local avec Python
- [applications Web](#) (en PHP ou CGI-Python, base de données MySQL)
  - [nombre d'inscrits](#)
  - [inscription en ligne](#) (essayez !)
  - [classement en ligne](#)
  - [record](#)
  - [dernières parties](#)
  - [dernière version](#)



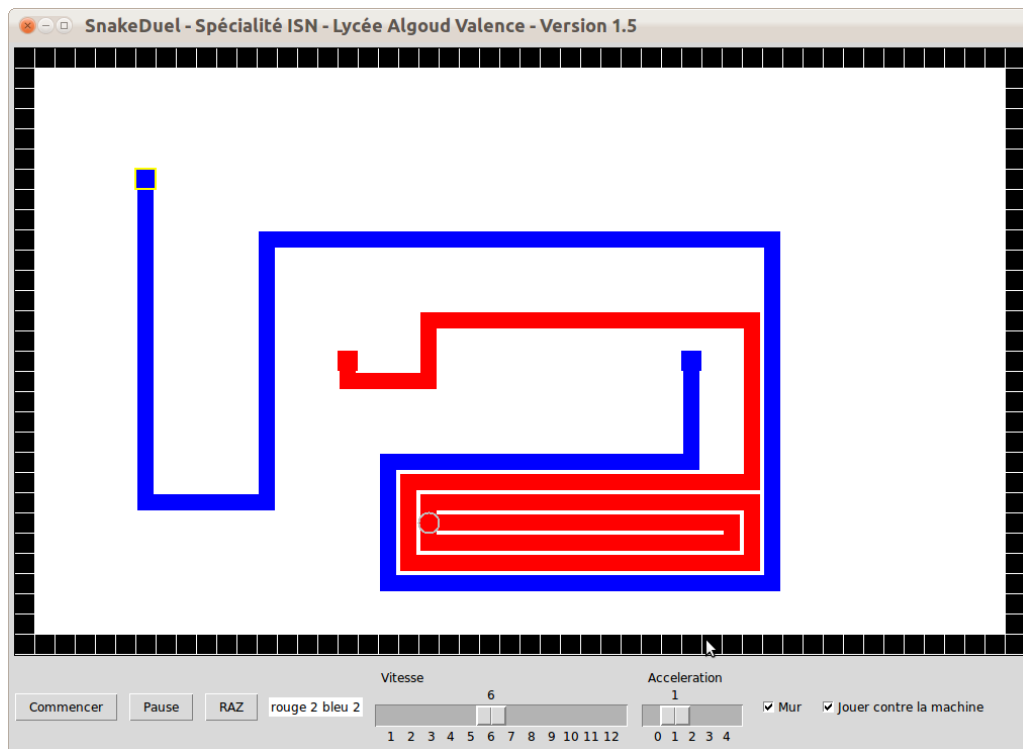
- Télécharger le fichier exécutable

Le fichier exécutable est téléchargeable [ici](#).

### Projet n°3 ★★★★★ Jeu de SnakeDuel

Un jeu qui se joue à deux, ou seul contre l'ordinateur.

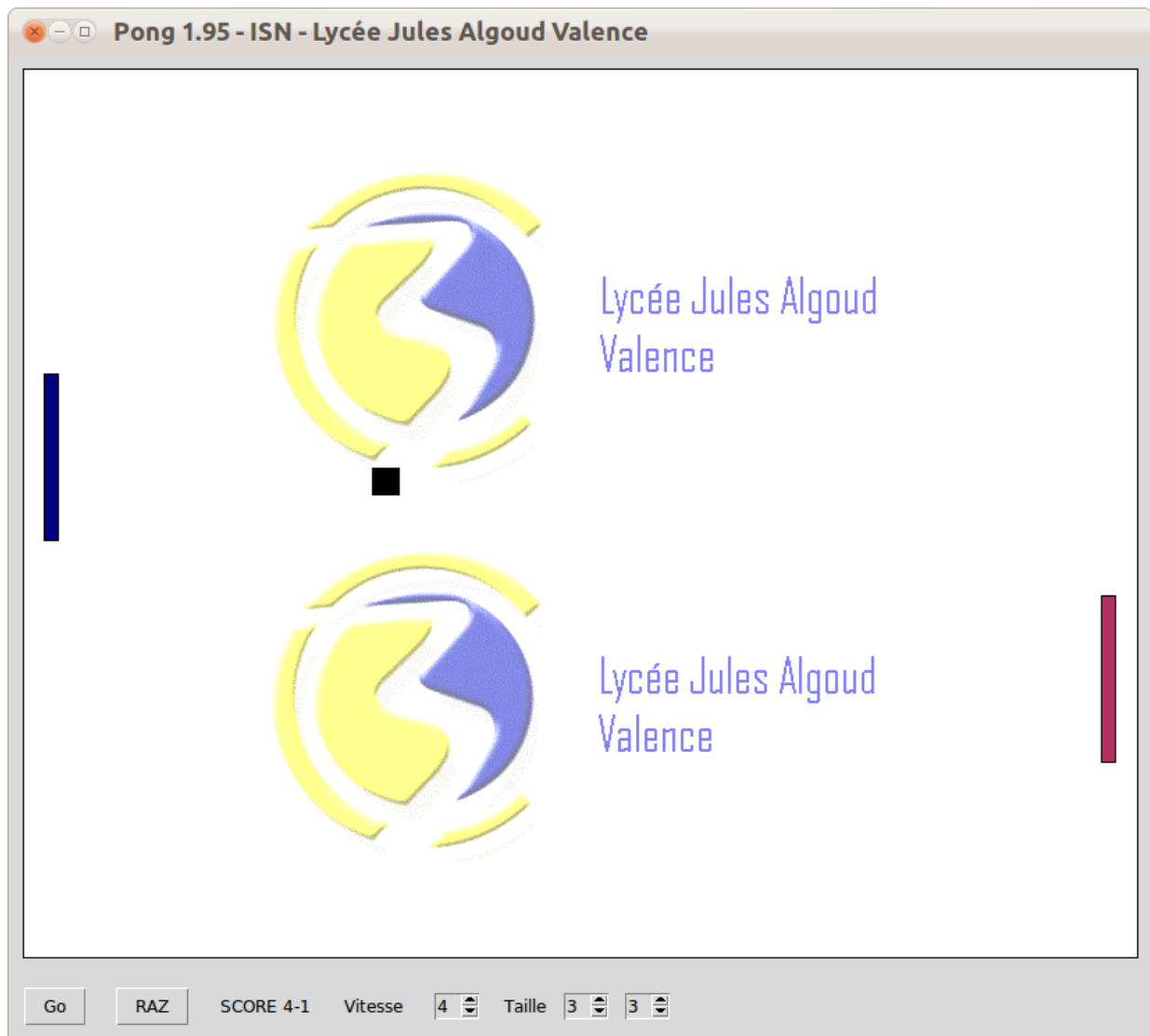
Le fichier exécutable est téléchargeable [ici](#).



#### Projet n°4 ★★★★★ Jeu de Pong

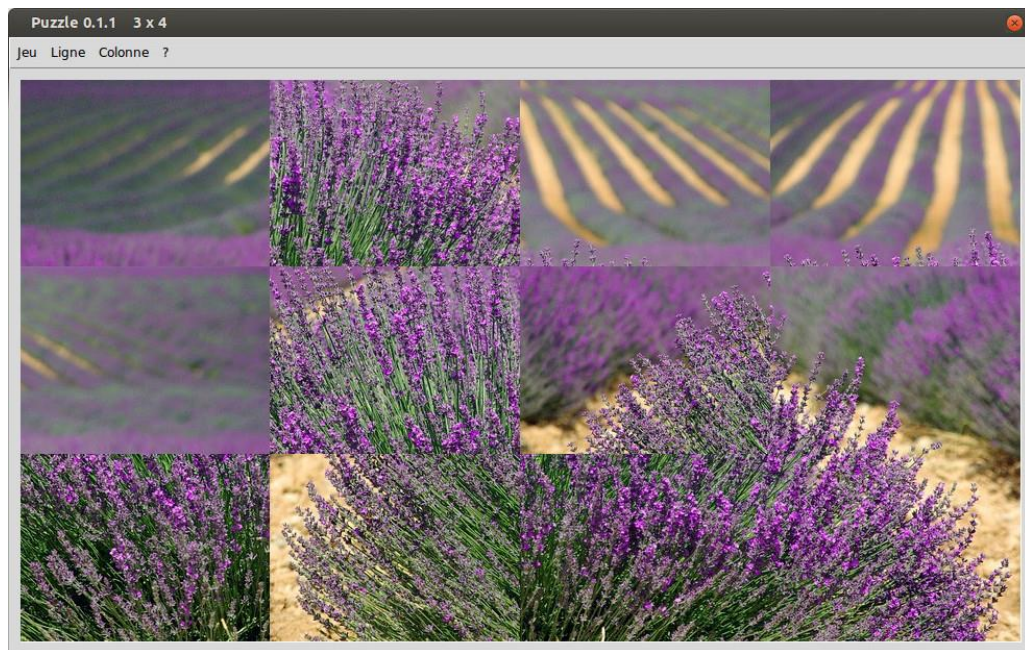
Un jeu qui se joue à deux.

Le fichier exécutable est téléchargeable [ici](#).

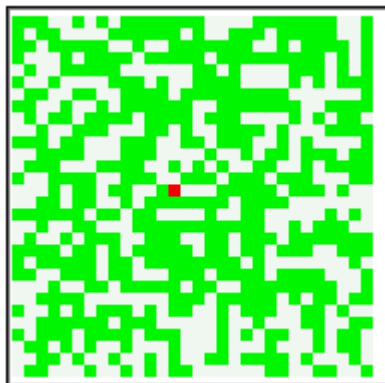


### Projet n°5 ★★★★★ Puzzle

Le fichier exécutable est téléchargeable [ici](#).



## Projet n°6 ★★★ Simulation de feux de forêt



Liens utiles :

- [cormas.cirad.fr](http://cormas.cirad.fr)
- [deptinfo-ensip.univ-poitiers.fr](http://deptinfo-ensip.univ-poitiers.fr)

### Programmes exécutables pour Windows

Pas besoin d'avoir Python sur votre machine !

Les programmes exécutables (extension `.exe`) des exercices et de la plupart des projets sont téléchargeables [ici](#) (7 Mo).

Décompresser ensuite l'archive.

Pour jouer à Tetris (par exemple), lancer le programme `tk_Tetris.exe`

Plus d'informations sur les jeux [ici](#).

Have fun !

## Remarques

- Testé avec succès sous Windows XP, Windows 7 et ... Linux/Ubuntu (avec l'émulateur Wine).

## Webographie

- [Création d'interface graphique avec le module tkinter \(niveau classe de seconde\)](#)
- [leducdenormandie.itslearning.com](http://leducdenormandie.itslearning.com)
- [fr.wikibooks.org \(tkinter\)](http://fr.wikibooks.org/tkinter)
- [fr.wikibooks.org \(fenêtres et graphismes\)](http://fr.wikibooks.org/fen%C3%AAtres_et_graphismes)
- [python.developpez.com](http://python.developpez.com)
- [docs.python.org](http://docs.python.org)
- [infohost.nmt.edu](http://infohost.nmt.edu)
- [effbot.org](http://effbot.org)
- [epydoc.sourceforge.net](http://epydoc.sourceforge.net)
- [www.tutorialspoint.com](http://www.tutorialspoint.com)
- [lmgty.com](http://lmgty.com)
- [Module pygame.mixer.Sound \(documentation\)](#)
- [Module pygame.mixer.music \(documentation\)](#)