

OBJECTIFS : L'objectif de ce TP est de découvrir d'autres fonctionnalités graphiques de la bibliothèque *Tkinter* de python en concevant « une animation de balle rebondissante ».

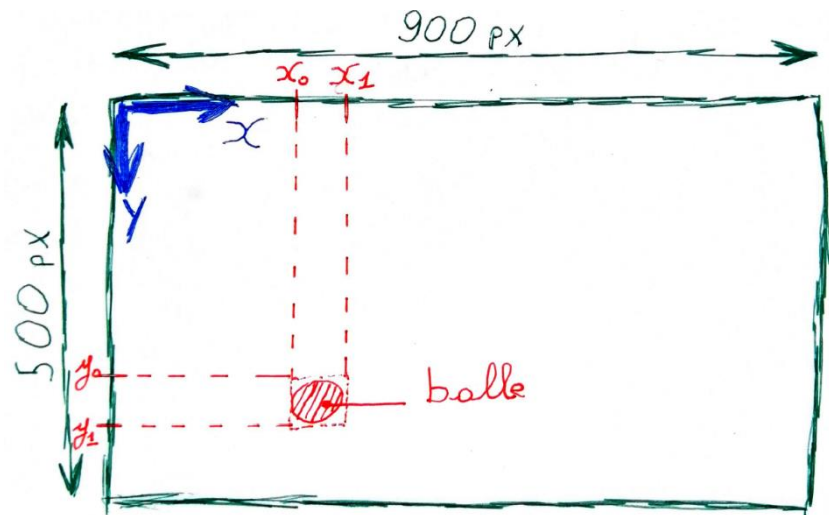
L'évaluation de ce travail est basée sur le rendu du fichier *.py* qui sera constitué.

1. CONCEPTION DU JEU SUR PAPIER :

On commence par définir les bases de ce que l'on veut faire sur papier :

Animation imaginée : voir *.gif* sur *nsibrantly.fr*

« Une balle se déplace automatiquement en diagonale et rebondit sur les côtés de la fenêtre quand on fait un clic gauche avec la souris. Elle s'arrête lorsque l'on fait un clic droit ».



Que devra réaliser le code python à écrire ?

- Partie graphique :

- Créer une fenêtre Tkinter : \Rightarrow fonction *creer_fenetre()* pour créer un objet Tk nommé *fenetre*.
- Créer 1 widget Tkinter : une zone graphique (objet *Canvas*) :
 \Rightarrow fonction *creer_widgets()* pour créer 1 objets Tk nommé *zone_graphique*
- Créer 1 item graphique dans le Canvas : un cercle de diamètre 30 px pour faire office de balle
 \Rightarrow fonction *graphique_jeu()* pour créer 1 items Tk nommés *balle*

- Partie animation :

- Créer 2 évènements liés aux clics gauche et droit de la souris
 \Rightarrow fonctions *go()* qui modifient les coordonnées du palet et *arret()*.
- Gérer le mouvement de la balle rebondissante : \Rightarrow fonction *deplace()*

2. Création de la partie Graphique :

Infos :

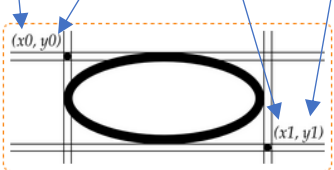
- Pour créer un cercle dans un Canevas, on utilise la méthode `create_oval()` appliquée à l'objet canvas. Cela donne :

Numéro de cet objet géométrique inséré dans le *canvas* stocké ici dans la variable *balle*

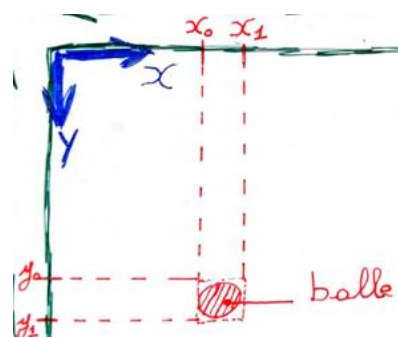
```
balle = zone_graphique.create_oval(100 , 100 , 120 , 120 , fill = "white")
```

Nom canvas

Nom méthode



Couleur de remplissage



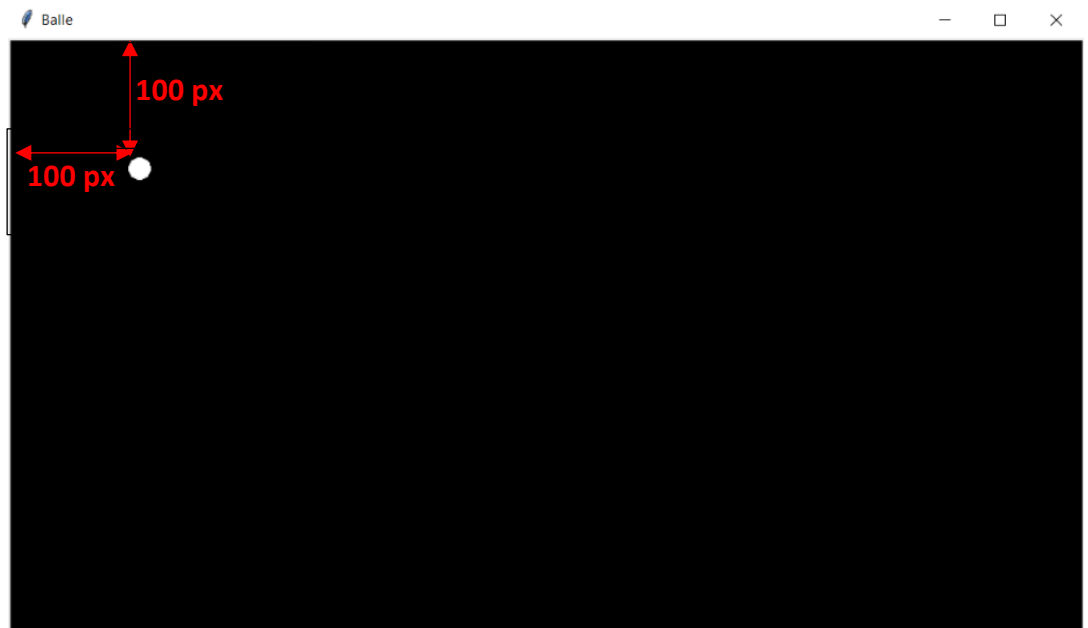
Ici, pour un cercle de 20px de diamètre :

- $x_0 = 100$
- $y_0 = 100$
- $x_1 = 120$
- $y_1 = 120$

⇒ Ouvrir un fichier et l'enregistrer sous le nom `tp12.py` .

Le code qui suit permet de constituer la partie graphique de l'animation. Il est incomplet.

⇒ Le compléter pour obtenir le résultat donné ci-contre :



```
# Modules -----
from tkinter import Tk , Canvas

# Fonctions -----
def creer_fenetre() :
    [redacted]

def creer_widgets() :
    [redacted]

def graphique_jeu() :
    [redacted]

# Main -----
dx = 5
dy = 5

fenetre = creer_fenetre()
zone_graphique = creer_widgets()
balle = graphique_jeu()

fenetre.mainloop()
```

3. Création de la partie Animation :

3.1. Création de l'évènement clic gauche qui crée le mouvement de la balle :

La fonction callback de l'évènement sera appelée *go()*. Pour démarrer vous pouvez partir du code ci-dessous et de la compléter.

```
def go(event) :
    deplace()

def deplace() :
    x0,y0,x1,y1 = zone_graphique.coords(balle)
    if x1 + dx > 900 or x0 + dx < 0 : dx = -dx
    [redacted]
    zone_graphique.coords(balle, x0+dx, y0+dy, x1+dx, y1+dy)
```

⇒ A vous de poursuivre.

3.2. Création de l'évènement clic droit qui stoppe le mouvement de la balle :

La fonction callback de l'évènement sera appelée *arret()*.

⇒ Finaliser le code et tester son bon fonctionnement.

----- FIN de ce Tp -----

Transférer le fichier tp12.py **par l'intermédiaire de l'onglet transfert** du site <https://nsibranly.fr> en utilisant le code : **tp12** .