

TP - Balles rebondissantes

Objectifs :

- ▷ Appliquer le principe de la programmation objet : classes, attributs, méthodes, objets.
- ▷ Ecrire la définition d'une classe
- ▷ Accéder aux attributs et méthodes d'une classe.
- ▷ Découvrir le module *Pygame*

1 Prise en main de Pygame

```
1 import pygame, sys
2 import time
3 from pygame.locals import *
4
5 LARGEUR = 640
6 HAUTEUR = 480
7 RAYON = 20
8
9 pygame.display.init()
10 fenetre = pygame.display.set_mode((LARGEUR, HAUTEUR))
11 fenetre.fill([0,0,0])
12
13 x = 300
14 y = 200
15 dx = 4
16 dy = -3
17 couleur = (45, 170, 250)
18
19 while True:
20     fenetre.fill([0, 0, 0])
21     pygame.draw.circle(fenetre, couleur, (x, y), RAYON)
22
23     x += dx
24     y += dy
25
26     pygame.display.update()
27
28     # routine pour pouvoir fermer proprement la fenetre Pygame
29     for event in pygame.event.get():
30         if event.type == pygame.QUIT:
31             pygame.display.quit()
32             sys.exit()
33
34     time.sleep(0.1)
```

1.1 Rajout d'un rebond sur les parois

Modifier le code précédent afin que la balle rebondisse sur chaque paroi (il suffit de modifier intelligemment les variables de vitesse *dx* et *dy*).

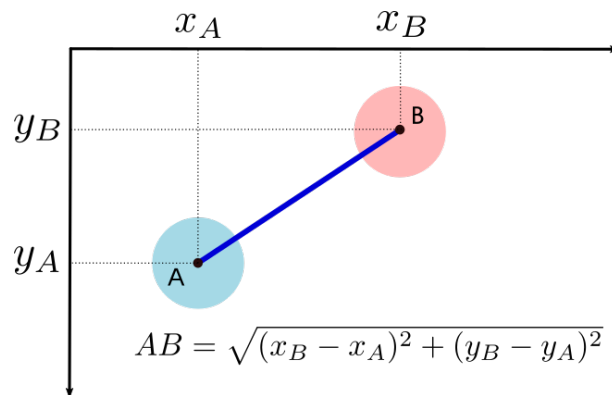
1.2 Rajout d'une deuxième balle

Attention au nommage des variables...

1.3 Gestion de la collision entre deux balles

1. À l'aide d'un schéma (papier-crayon!), mettez en évidence le test devant être réalisé pour détecter une collision.

Indice :



2. Implémenter ce test (en créant pour cela une fonction distance) et affichez "collision" en console lorsque les deux balles se touchent.
3. Pour donner l'illusion physique du rebond, échangez les valeurs respectives de dx et dy pour les deux balles.

1.4 Rajout d'une troisième balle et gestion du rebond avec les deux autres

Vraiment ? Peut-on continuer comme précédemment ?

2 La POO à la rescousse : création d'une classe Balle

2.1 La classe Balle

L'objectif est que la méthode constructeur dote chaque nouvelle balle de valeurs aléatoires : abscisse, ordonnée, vitesse, couleur...

- ▷ Pour l'aléatoire, utiliser `randint(a, b)` qui renvoie un nombre pseudo-aléatoire entre `a` et `b`. Il faut pour cela importer la fonction avec : `from random import randint`
- ▷ Vous pouvez aussi doter votre classe `Balle` d'une méthode `dessine` (qui affiche la balle), ainsi qu'une méthode `bouge` qui la fait bouger.
- ▷ Créer cette classe et instancier une balle.

2.2 Plusieurs balles

L'idée est de stocker dans une liste `sac_a_balles` un nombre déterminé de balles...

2.3 Collision de toutes les balles

Il «suffit» , dans la méthode constructeur, de tester la collision de la balle `self` avec chacune des balles de notre `sac_a_balles`

