

Dernière mise à jour	Informatique	Denis DEFAUCHY
30/08/2022	8 – for if while	TD 8-2 – Sujet

Informatique

8

for if while

TD 8-2
Balle rebondissante

Dernière mise à jour	Informatique	Denis DEFAUCHY
30/08/2022	8 – for if while	TD 8-2 – Sujet

Objectif

L'objectif de ce sujet est de mettre en œuvre de manière ludique les principes de l'algorithmique. Vous allez créer un petit programme qui permettra à une balle de rebondir indéfiniment sur les bords d'une fenêtre d'affichage graphique.

Prise en main

Téléchargez le code élèves disponible en ligne ici ([LIEN](#)).

Un code y est présent, il a pour le but de vous simplifier la tâche en ne travaillant que sur l'aspect algorithmique et non sur l'affichage :

- Import des librairies d'affichage
- Définition de 5 fonctions d'affichage qui ne seront qu'utilisées par la suite
- Zone de programmation dans laquelle vous aurez à coder, appelée « Zone de code »

Dans la partie « Création de la zone de mouvement », quelques lignes sont définies, vous ne devriez pas à avoir à les modifier (sauf si vous le souhaitez) :

- Définition des dimensions de la fenêtre d'affichage : $x \in [-200, 200]$ et $y \in [-200, 200]$
- Dimensionnement de l'épaisseur des bordures de la zone de mouvement, affichées à l'extérieure de la zone de 200 par 200
- Définition de la taille de la fenêtre d'affichage
- Initialisation de la figure qui servira à afficher la balle
- Création d'une variable « Delai » que vous pourrez modifier (plus elle est petite, plus la balle ira vite), correspondant à une attente entre l'affichage de deux images

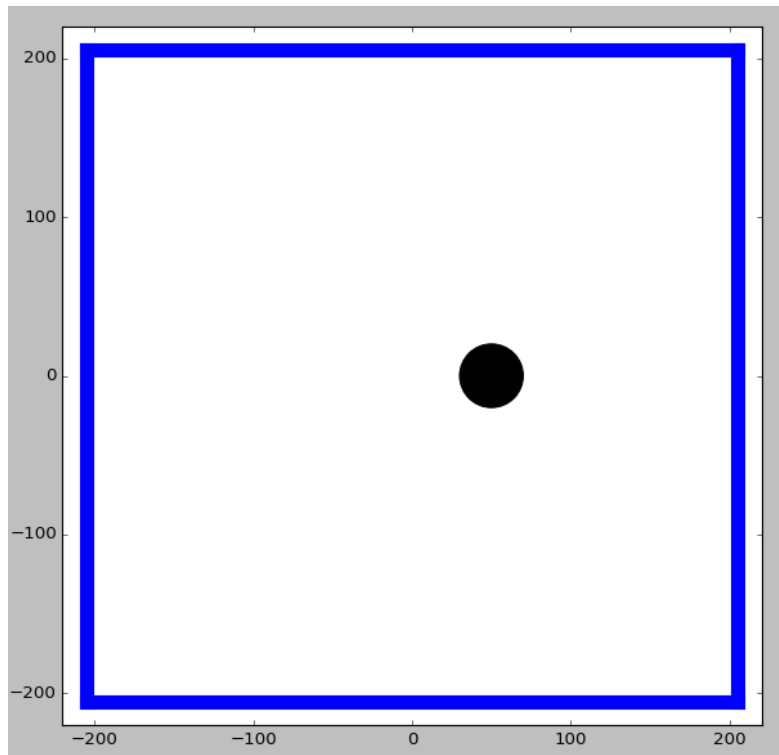
Dans la partie « Mouvement de la balle », quelques lignes sont définies :

- Coordonnées du centre de la balle X , Y et de son rayon R
- Effacement du schéma s'il était rempli précédemment
- Création des bordures dans la zone d'affichage
- Création de la balle dans la zone d'affichage
- Affichage de la situation

C'est ici que vous allez jouer !

Dernière mise à jour	Informatique	Denis DEFAUCHY
30/08/2022	8 – for if while	TD 8-2 – Sujet

Lancez le code fourni, vous devriez voir apparaître l'image suivante :



Si une erreur vous indique que Matplotlib n'est pas installé, tapez dans la console : « conda install matplotlib » et exécutez cette ligne. L'installation se fera automatiquement. Sinon, cherchez bien, une fenêtre s'est ouverte !!!

Rebonds d'une balle

En ne travaillant que dans la partie « Mouvement de la balle » :

- Mettre en place un algorithme qui déplace la balle automatiquement en diagonale (+1 ou -1 verticalement et horizontalement à chaque itération, par exemple $\begin{cases} X += dx \\ Y += dy \end{cases}$)
- Gérer les rebonds à chaque rencontre d'un bord. Attention, la balle ne doit pas mordre le bord et l'algorithme doit fonctionner quel que soit le rayon R et la dimension de la zone de mouvement.

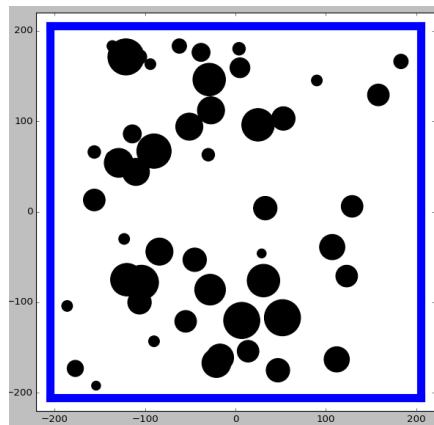
Remarque : Pour réaliser une boucle infinie, utilisez la commande « `while True :` ». Pensez à casser votre programme avec « Ctrl+i » avant chaque exécution afin de reprendre la main sur la console.

Dernière mise à jour	Informatique	Denis DEFAUCHY
30/08/2022	8 – for if while	TD 8-2 – Sujet

Plusieurs balles

Si vous avez fini la première partie, essayez d'adapter votre code afin de :

- Placer aléatoirement n balles dans la zone de déplacement (sans mordre les bordures) de rayons aléatoires choisis dans une plage, par exemple entre 5 et 10
- Les faire se déplacer dès l'instant initial dans les 4 directions aléatoirement
- Gérer les rebonds



On ne gèrera pas les collisions.

Envie de plus ? Agar.io

A chaque fois que deux balles se touchent, elles n'en font plus qu'une dont la surface vaut la somme des deux (on arrondira le rayon à l'entier le plus proche), et la plus grosse l'emporte sur la position et la direction du déplacement.

Remarques :

- On se place dans un mode où la vitesse reste constante quelle que soit la taille !
- Si lors d'une fusion, une balle dépasse du cadre, on la replacera automatiquement dans le cadre

Dernière mise à jour	Informatique	Denis DEFAUCHY
30/08/2022	8 – for if while	TD 8-2 – Sujet

Fou de programmation ?

On définit la couleur d'une des balles rouge ('r'), et on peut la déplacer en touchant les flèches du clavier. Ce programme devient un jeu où le but est d'être le dernier vivant.

Un peu d'aide pour la gestion des touches :

```
fig.canvas.mpl_connect('key_press_event', f_mouvement)

def f_mouvement(event): # Rq: à la suite d'un bug identifié, bien fermer
    la figure entre 2 lancements du jeu
    if event.key == ' ': # Fin du jeu
        if C.count('r') != 0: # Au cas où la rouge vient d'être mangée !
            C.remove('r')
    elif event.key == 'down':
        dY[ind_rouge] = -1
    elif event.key == 'up':
        dY[ind_rouge] = 1
    elif event.key == 'left':
        dX[ind_rouge] = -1
    elif event.key == 'right':
        dX[ind_rouge] = 1
    elif event.key == 'enter': # Triche
        R[ind_rouge] += 1
```

Avec `ind_rouge` l'indice (en variable globale) de la balle rouge dans les différentes listes.