

Conseils pratiques :

- Travaillez dans Thonny. Quand votre code vous semble correct, collez-le dans le cadre bleu de réponse.
- Lorsque votre document est prêt à être rendu, cliquez sur «Rendre» dans Google Classroom.

### Exercice

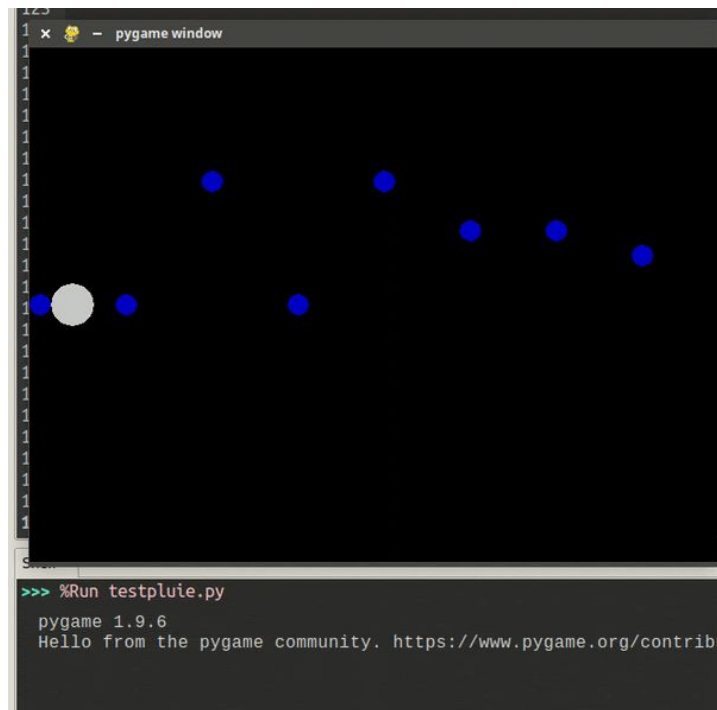
1. Construire une classe `Fraction` pour représenter un nombre réel. Cette classe doit contenir deux attributs `numérateur` et `denominateur`.

2. Rajouter une méthode `valeur_dec` qui renvoie la valeur décimale de la fraction si celle-ci est calculable, ou bien `None` si le dénominateur vaut 0.

3. Rajouter une méthode `__str__` qui renvoie une chaîne de caractères de la forme `"12/25"` lors d'un appel par `print()`, ou bien simplement `"12"` si le fraction vaut 12/1.

## Problème

Le but de cet exercice est de coder le jeu ci-dessous.



Une boule blanche (le «vaisseau») doit attraper en un minimum de temps les gouttes de pluie. Lorsqu'elles ont toutes été attrapées, le jeu se ferme et le temps apparaît.

Un squelette de code vous est proposé. Il est téléchargeable à cette adresse :

[https://github.com/glassus/nsi/tree/master/Terminale/DS/DS01/jeu\\_pluie\\_a\\_completer.py](https://github.com/glassus/nsi/tree/master/Terminale/DS/DS01/jeu_pluie_a_completer.py)

Ouvrez-le avec Thonny.

### Question 1 : méthode constructeur de la classe Goutte

Il faut ici compléter la méthode constructeur. L'objet doit être doté des attributs suivants :

- `x` , égal au paramètre `posx`
- `y` , égal par défaut à une valeur que vous devez définir
- `dy`, qui représente la vitesse verticale de l'objet. Cette vitesse doit être un nombre aléatoire entre 5 et 10
- `color` , qui sera égal à la couleur de votre choix
- `taille` , qui sera égal à `taille_goutte`
- `alive` , qui sera un booléen indiquant si la goutte est vivante ou pas. Par défaut, elle sera vivante.

```
class Goutte :  
    def __init__(self, posx):
```

**Question 2 : déplacement de l'objet**

Il faut faire en sorte que l'objet se déplace verticalement de  $dy$  à chaque tour de boucle.

**Question 3 : rebond de l'objet**

L'objet doit rebondir sur le mur du haut et du bas.

```
def move(self):  
    #Q2 et Q3 sont à coller ici
```

**Question 4 : compléter le range****Question 5 : affichage et mouvement de toutes les gouttes**

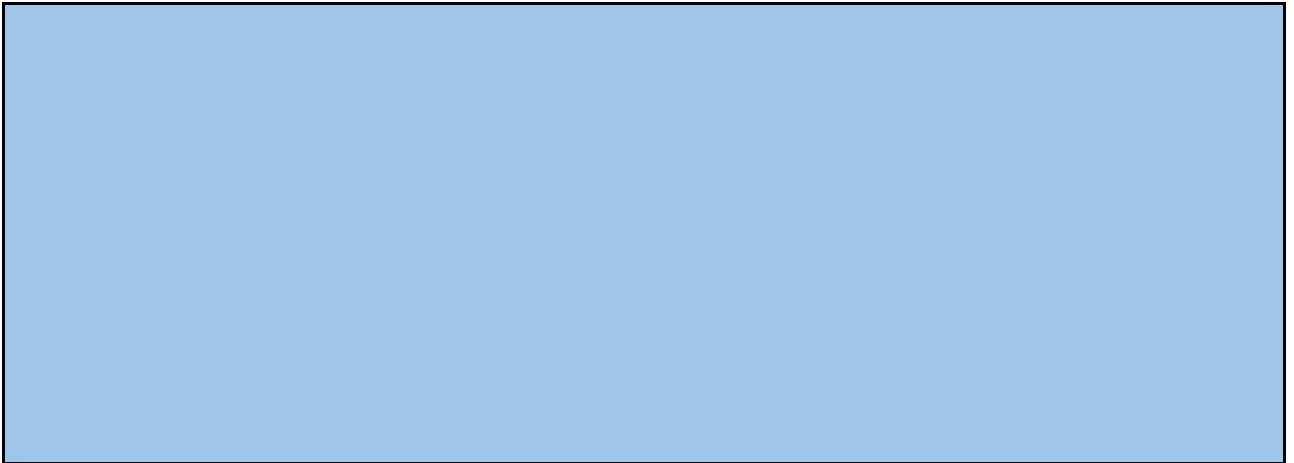
Toutes les gouttes doivent être affichées et mises en mouvement.

**Question 6 : affichage du vaisseau du joueur**

Le vaisseau du joueur doit s'afficher.

**Question 7 : mouvement du vaisseau au clavier**

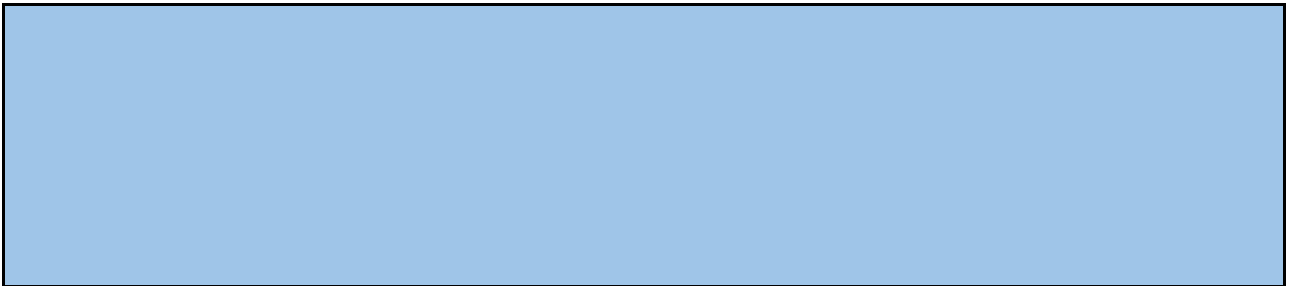
Chaque mouvement sera de 5 pixels.

**Question 8 : collision des gouttes et du vaisseau**

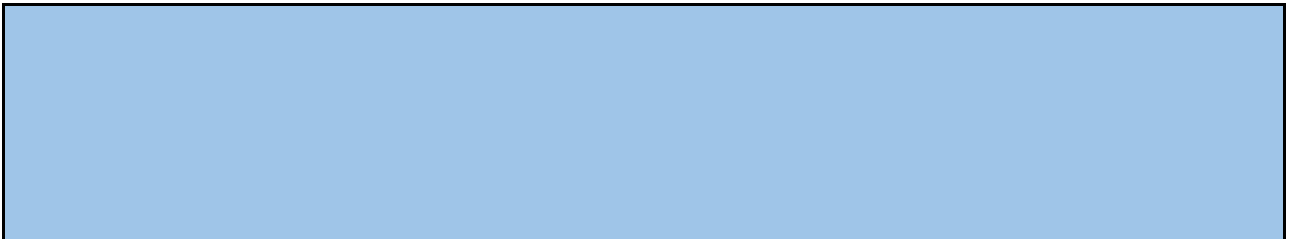
Il faut ici gérer la collision entre le vaisseau du joueur et les gouttes. En cas de collision, passer l'attribut `alive` de la goutte touchée à `False`.

**Question 9 : test de fin de jeu**

Il faut compléter le `if` pour que le jeu s'arrête quand il doit s'arrêter.

**Question 10 : affichage du temps mis par le joueur**

Calcul et affichage du temps que le joueur a mis pour toucher toutes les gouttes.



---

## Intégralité du code

