

Exercice 5 points

1. **2 points** Construire une classe `Fraction` pour représenter un nombre réel. Cette classe doit contenir deux attributs `numérateur` et `denominateur`.

```
class Fraction :  
    def __init__(self, num, den) :  
        self.numérateur = num  
        self.denominateur = den
```

2. **1.5 points** 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.

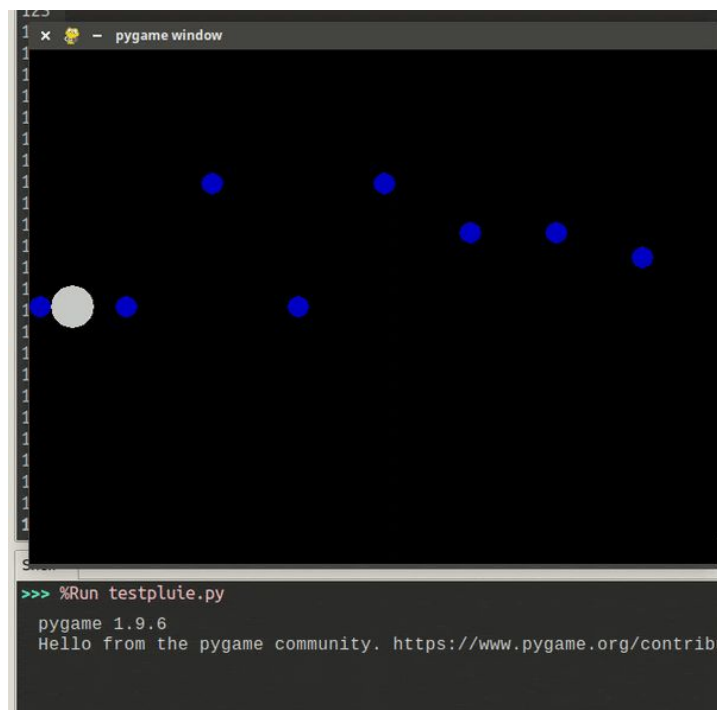
```
class Fraction :  
    def __init__(self, num, den) :  
        self.numérateur = num  
        self.denominateur = den  
  
    def valeur_dec(self):  
        if self.denominateur == 0 :  
            return None  
        else :  
            return self.numérateur / self.denominateur
```

3. **1.5 points** 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.

```
class Fraction :  
    def __init__(self, num, den) :  
        self.numérateur = num  
        self.denominateur = den  
  
    def valeur_dec(self):  
        if self.denominateur == 0 :  
            return None  
        else :  
            return self.numérateur / self.denominateur  
  
    def __str__(self):  
        if self.denominateur == 1 :  
            return str(self.numérateur)  
        else :  
            return str(self.numérateur)+"/"+str(self.denominateur)
```

Problème 15 points

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

Ouvrez-le avec Thonny.

Question 1 : méthode constructeur de la classe Goutte

2 points 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):  
        self.x = posx  
        self.y = 10  
        self.dy = randint(5,10)  
        self.color = (0,0,200)  
        self.taille = taille_goutte  
        self.alive = True
```

Question 2 : déplacement de l'objet

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

Question 3 : rebond de l'objet

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

```
def move(self):  
    #Q1  
    self.y += self.dy  
  
    #Q2  
    if self.y < self.taille or self.y > ecran[1]-self.taille :  
        self.dy = -self.dy
```

Question 4 : compléter le range

1 point

```
liste_gouttes = []  
for k in range(nb_gouttes):  
    liste_gouttes.append(Goutte(10+k*ecran[0]//nb_gouttes))
```

Question 5 : affichage et mouvement de toutes les gouttes

2 points Toutes les gouttes doivent être affichées et mises en mouvement.

```
for g in liste_gouttes:  
    g.move()  
    g.draw()
```

Question 6 : affichage du vaisseau du joueur

1 point Le vaisseau du joueur doit s'afficher.

```
player.draw()
```

Question 7 : mouvement du vaisseau au clavier

2 points Chaque mouvement sera de 5 pixels.

```
keys=pygame.key.get_pressed()

if keys[K_LEFT]:
    player.x -= 5
if keys[K_RIGHT]:
    player.x += 5
if keys[K_UP]:
    player.y -= 5
if keys[K_DOWN]:
    player.y += 5
```

Question 8 : collision des gouttes et du vaisseau

2 points 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.

```
for g in liste_gouttes :
    if ((g.x-player.x)**2+(g.y-player.y)**2)**0.5 <
g.taille+player.taille :
    g.alive = False
```

Question 9 : test de fin de jeu

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

```
if len(liste_gouttes) == 0 :
    pygame.display.quit()
    print("durée :", time.time()-chrono)
    sys.exit()
```

Question 10 : affichage du temps mis par le joueur

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

```
print("durée :", time.time()-chrono)
```

Intégralité du code

