Mini_projet 2 : tapis de Sierpinsky

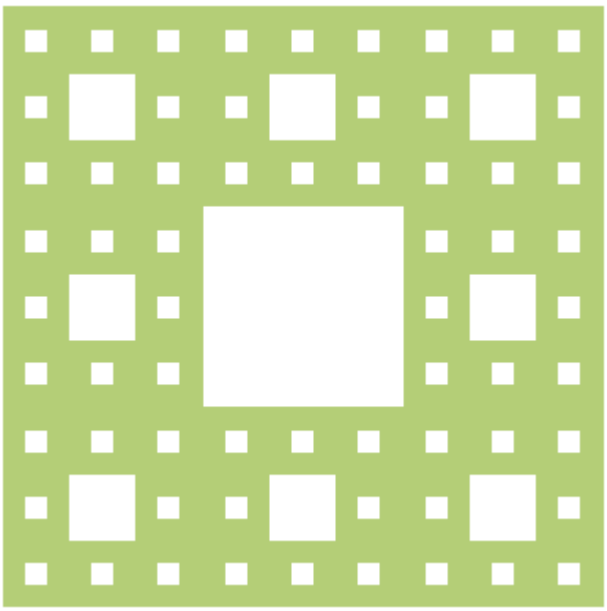
Récupérer le fichier **sierpinsky.py** donné en ressources.

Les différentes étapes du mini-projet 2

Etape 1 : Compléter la classe **Carre** sachant que cette classe doit avoir les caractéristiques suivantes :

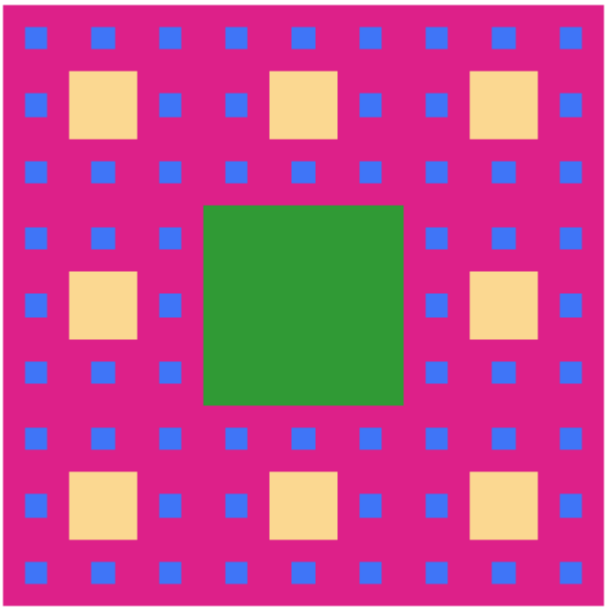
```
Classe Eleve :

    Attributs privés :
        __cote
        __abscisse
        __ordonnee

    Accesseurs :
        get__cote()
        get__abscisse()
        get__ordonnee

    Méthode:
        __init__(self, cote, abscisse, ordonnee)
        afficher(self)
```

Etape 2 : Compléter le code de la fonction récursive **tapis(n, longueur, a, o)** qui doit tracer un carré de côté **c = longueur//3** au coin inférieur de coordonnées **(a+c ; o+c)** puis faire plusieurs appels récursifs (on poura utiliser deux boucles **for** imbriquées) pour obtenir le tapis de Sierpinsky comme ci-dessous pour **n = 3** . Il faut compléter le dernier **pass** dans le code principal pour appeler une première fois la fonction **tapis(...)** .

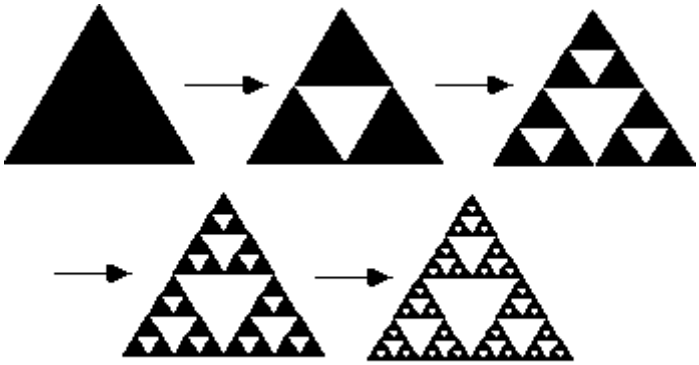


Etape 3 : Compléteter la boucle **for** dans le code principal, pour remplir de façon aléatoire la liste des couleurs. Puis utiliser cette liste de couleur, dans la fonction récursive **tapis(...)** afin que tous les carrés de même taille aient la même couleur, mais une couleur différente des autres carrés de taille différentes. Voici un exemple de tapis que l'on peut obtenir.

Remarque : Dans la liste de couleur, les couleurs sont stockées en tant que tuple de trois valeurs entières comprises entre 0 et 255, représentant le code RVB de la couleur.



*** Etape 4 (bonus) :** Addapter le code du tapis de Sierpinsky, pour obtenir le **triangle de Sierpinsky**.



*** Etape 5 (bonus) :** Addapter ce code, pour obtenir le **pentagone de Sierpinsky**.