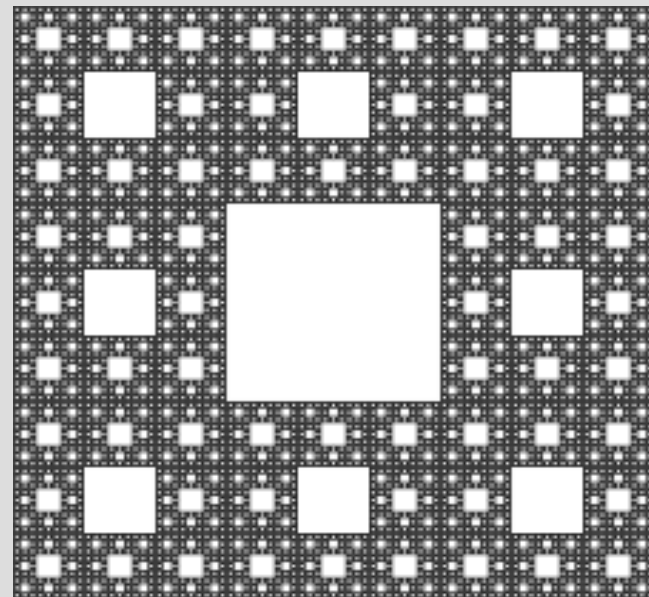
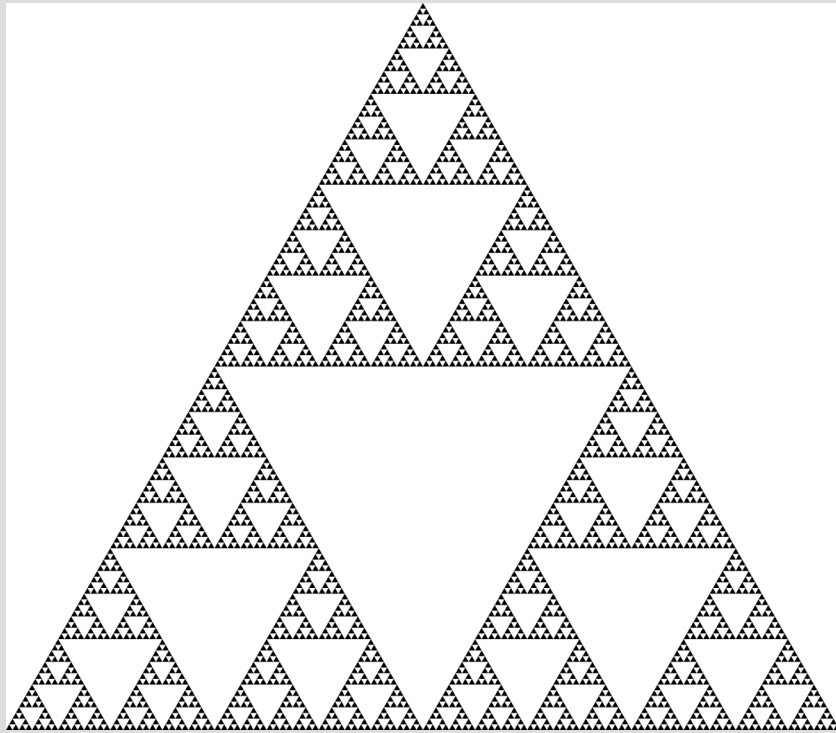


Les fractales de Sierpinski



TERENCE SURENDRA Darvin et KAZADI NTAMBWE Aristote
L2 Mathématiques
Mr Cardinal
Année 2018/2019

Sommaire

- ◆ I. Qui est Sierpinski ?
- ◆ II. Qu'est ce qu'une fractale
- ◆ III. Présentation des programmes
- ◆ IV. Conclusion

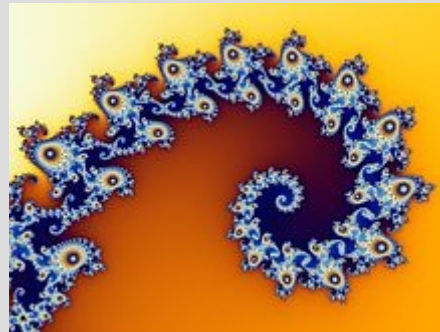
I. Qui est Sierpinski ?



Wacław Sierpinski est un mathématicien polonais né le 14 mars 1882 et mort le 21 octobre 1969. Il est connu pour ses contributions à la théorie des ensembles, la théorie des nombres, la théorie des fonctions et la topologie.

II. Qu'est ce qu'une fractale ?

Une figure fractale est un objet mathématique, telle une courbe ou une surface, dont la structure est invariante par changement d'échelle. De nombreux phénomènes naturels comme le tracé des lignes de côtes possèdent des formes fractales approximatives. Un objet fractal est un objet dont chaque élément est aussi un objet fractal.



Ensemble de Mandelbrot

III. Présentation de nos programmes

Pour ces programmes, nous avons importé le module « turtle » qui permet de déplacer une tortue sur l'écran.

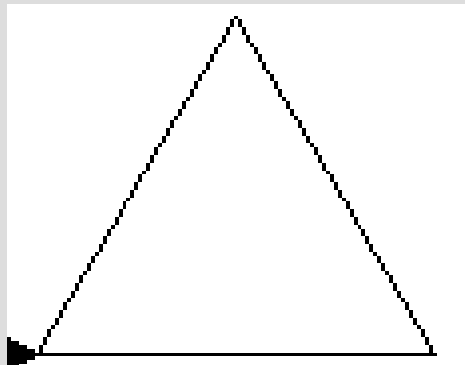
```
from turtle import *
```

Cela va nous permettre d'utiliser les fonctions « forward (fd) » et « backward » qui vont faire avancer ou reculer la tortue d'une certaine distance, « right » et « left » qui vont faire tourner la tortue d'un angle à droite ou à gauche, « penup » et « pendown » qui vont permettre de poser ou lever la tortue et « reset » qui permet de repartir à 0.

III. Présentation de nos programmes

Nous avons travaillé dans un premier temps sur le triangle de Sierpinski : grâce à votre aide, nous avons tout d'abord défini une fonction « triangle » qui va créer un simple triangle.

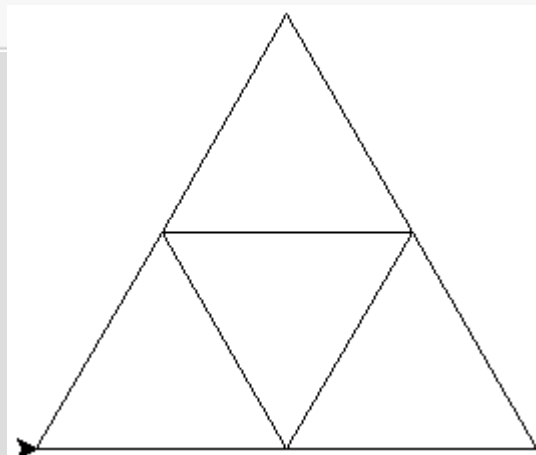
```
def triangle(1):  
    for i in range(3):  
        forward(1)  
        left(120)
```



III. Présentation de nos programmes

Nous avons ensuite défini une fonction « triple » qui va créer trois triangles orientés vers le bas dans le triangle précédent.

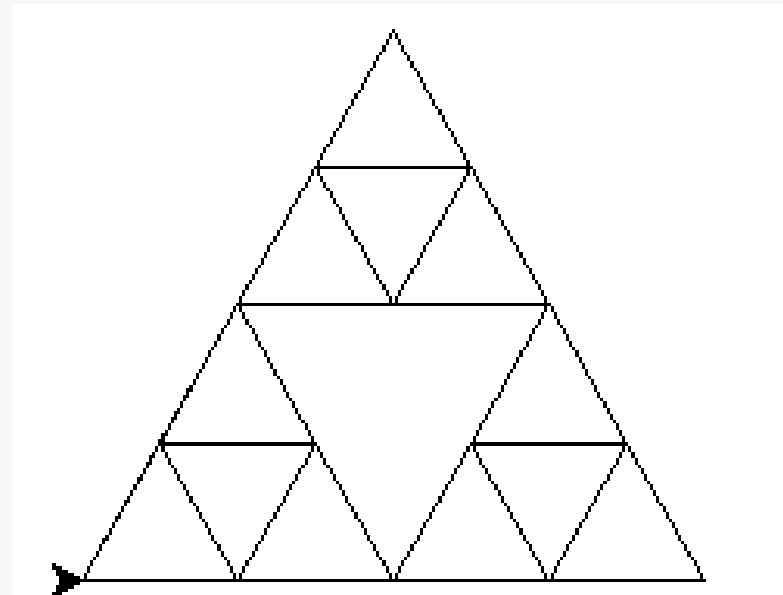
```
def triple(l):  
    triangle(l/2)  
    fd(l/2)  
    triangle(l/2)  
    left(120)  
    fd(l/2)  
    left(-120)  
    triangle(l/2)  
    left(-120)  
    fd(l/2)  
    left(120)
```



III. Présentation de nos programmes

Enfin, nous avons défini la fonction « `sierpinski2` » qui va reproduire 3 fois le triangle précédent afin de faire un grand triangle.

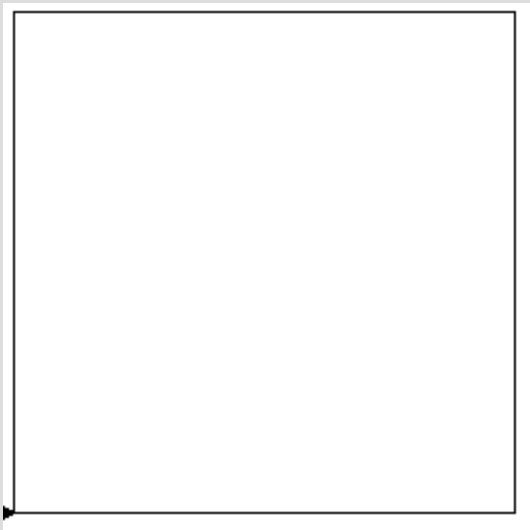
```
def sierpinski2(l,n):  
    if n==0:  
        triangle(l)  
    if n==1:  
        triple(l)  
    if n>1:  
        sierpinski2(l/2,n-1)  
        fd(l/2)  
        sierpinski2(l/2,n-1)  
        fd(-l/2)  
        left(60)  
        fd(l/2)  
        left(-60)  
        sierpinski2(l/2,n-1)  
        left(-120)  
        fd(l/2)  
        left(120)
```



III. Présentation de nos programmes

Dans un second temps, nous avons travaillé sur le tapis de Sierpinski : nous avons tout d'abord défini une fonction « carré » qui va créer un carré.

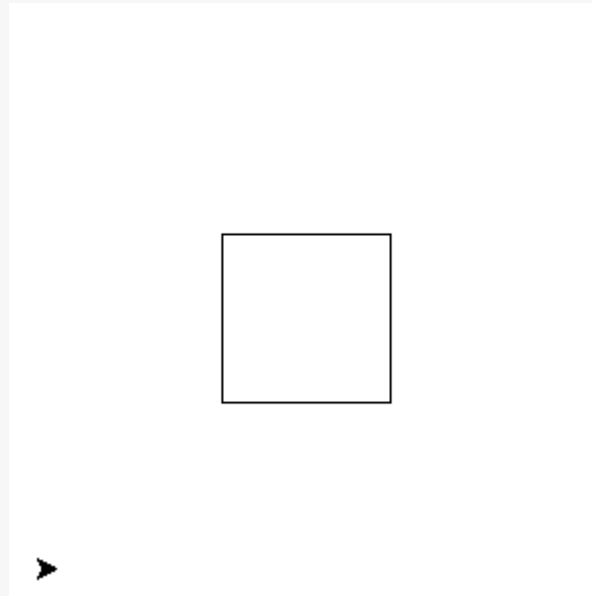
```
def carré(1):  
    for i in range(4):  
        fd(1)  
        left(90)
```



III. Présentation de nos programmes

Nous avons ensuite défini une fonction « petitcarré2 » qui va créer un petit carré au milieu du carré précédent sans faire le contour du carré précédent.

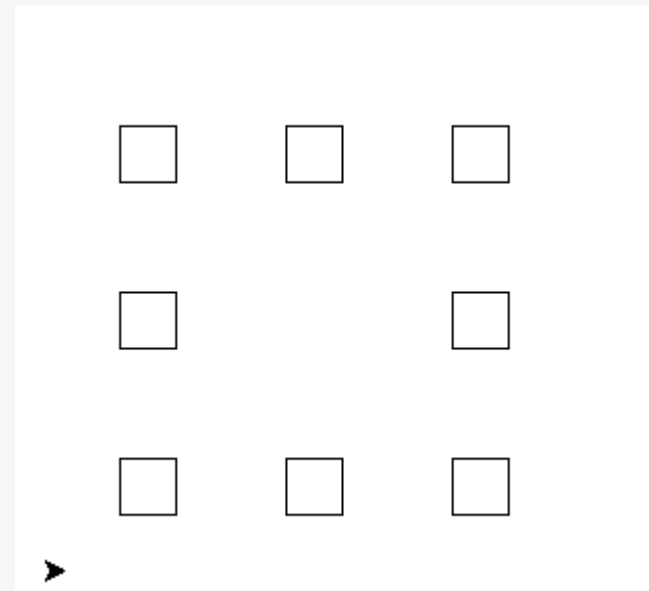
```
def petitcarré2(l):  
    penup()  
    fd(l/3)  
    left(90)  
    fd(l/3)  
    left(-90)  
    pendown()  
    for i in range(4):  
        fd(l/3)  
        left(90)  
    penup()  
    left(-90)  
    fd(l/3)  
    left(-90)  
    fd(l/3)  
    left(180)  
    pendown()
```



III. Présentation de nos programmes

Nous avons aussi défini une fonction « tapis » qui va dessiner les petits carrés autour du carré du milieu sans faire les contours du grand carré et du carré du milieu.

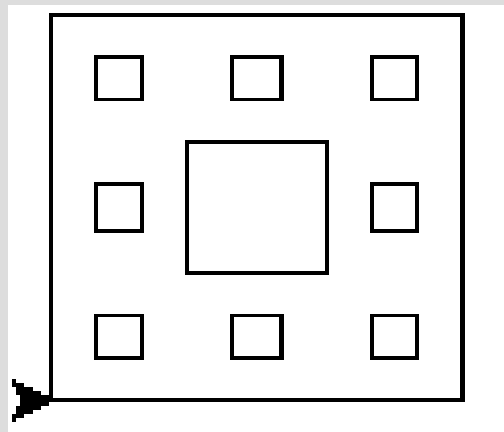
```
def tapis(l,n):  
    if n==1:  
        petitcarré2(l)  
    if n>1:  
        for p in range(4):  
            for i in range(2):  
                tapis(l/3,n-1)  
                penup()  
                fd(l/3)  
                pendown()  
            penup()  
            fd(l/3)  
            pendown()  
            left(90)
```



III. Présentation de nos programmes

Et nous avons enfin défini une fonction « tapis2 » qui va dessiner les contours du carré du milieu et du grand carré.

```
def tapis2(l,n):  
    carré(l)  
    for a in range(n):  
        tapis(l,n-a)
```



IV. Conclusion

Ce projet a été un projet très intéressant. Il nous a permis de réfléchir sur un programme itératif.

On peut voir que le principe de ces deux programmes est à peu près similaire et qu'on peut adapter ces programmes à plusieurs formes géométriques différentes en modifiant légèrement ces programmes.

Ce projet nous a aussi permis de réutiliser des mathématiques, d'utiliser ce qu'on a fait dans les notebooks et de travailler en groupe.