

Fractale de Von Koch

Projet à rendre pour le 30 octobre 2018

Données :

Un segment $[AB]$ de longueur a , n le numéro d'étape.

Le segment $[AB]$ seul représente l'étape 0.

Initialisation :

Construire A_1 et A_3 qui partagent le segment $[AB]$ en 3 parties égales (dans l'ordre A, A_1, A_3, B). Construire alors A_2 tel que A_2 soit sommet d'un triangle équilatéral $A_1A_2A_3$.

Tracer alors les segments $[AA_1], [A_1A_2], [A_2A_3], [A_3B]$.

Gommer le segment central $[A_1A_3]$.

Reproduire le processus sur chacun des 4 segments $[AA_1], [A_1A_2], [A_2A_3], [A_3B]$ jusqu'à l'étape n .

PARTIE DESSIN PAPIER (2 feuilles A4 blanches format paysage):

A l'aide d'un compas à la mine bien taillée, construire les étapes 1, 2, 3, et 4.

Choisir a de manière judicieuse...

Pour l'étape 2, faire un schéma détaillé qui indique les angles orientés de la figure.

L'objet que vous venez de construire s'appelle une fractale.

PREMIERS CALCULS :

A chaque étape, donner

- le nombre de segments S_n
- la longueur de chaque segment a_n
- la longueur totale de la figure obtenue T_n .

En déduire une expression de S_n, a_n et T_n de la figure obtenue à l'étape n en fonction de n .

PARTIE ALGORITHMIQUE :

Écrire l'algorithme qui permet de construire à l'écran l'étape 1 à partir des points A et B , en utilisant les instructions av (avance), td (tourne à droite) et tg (tourne à gauche), $pendown$ et $penup$ pour activer/désactiver un tracé éventuel. On suppose que la souris est au départ au point A et dirigée vers le point B .

...

On transforme cette série d'instructions en une fonction qui s'appelle $graine(a)$.

On a donc :

$graine(a)$

...

Écrire maintenant l'algorithme qui permet de construire l'étape 2 en utilisant les instructions de base déjà fournies, et la fonction graine(a).

```
graine(...)  
tg 60  
...  
...
```

On transforme cette série d'instructions en une fonction qui s'appelle VonKoch(a, etape), avec etape = 2 et où VonKoch(a,1) contient rigoureusement les instructions de la graine.

On a donc :

```
VonKoch(a, etape)          (etape = 2)  
  graine(...)  
  tg 60  
  ...  
  ...
```

ou encore :

```
VonKoch(a, etape)          (etape = 2)  
  VonKoch(..., etape-1)  
  tg 60  
  ...  
  ...
```

Écrire alors une fonction qui permet de tracer l'étape n.

```
VonKoch(a, etape)          (etape = n, n>=1)  
  Si etape == 1  
    graine(...)  
  Sinon  
    VonKoch(..., etape-1)  
    tg 60  
    ...  
    ...  
  FinSi
```

PROGRAMMATION :

- Programmer avec Snap! l'algorithme précédent.
- Programmer avec Snap! la fractale de Von Koch (flocon de neige) en appliquant 3 fois l'algorithme précédent aux 3 côtés d'un triangle équilatéral.

Enregistrer les images de la scène pour obtenir de jolis flocons de Von Koch.

RETOUR SUR LA PARTIE CALCULS :

On s'intéresse au flocon de Von Koch.

A chaque étape, donner

- la longueur totale de la figure obtenue U_n
- l'aire du flocon f_n obtenu à l'étape n : commencer par calculer f_1, f_2, f_3 puis en déduire f_n . Expliquer votre raisonnement.