Le triangle de Sierpinski

I. Le jeu du chaos (algorithme de constructions géométriques « à la main »)

On dispose d'un dé cubique équilibré dont les faces sont numérotées de 1 à 6.

Sur une feuille, placer trois points quelconques A, B, C non alignés.
Le triangle ABC est le « triangle de base ».

Choisir un point M au hasard.

Lancer le dé.

- Si le numéro obtenu est 1 ou 2, placer le milieu de $[AM]$;
- Si le numéro obtenu est 3 ou 4, placer le milieu de $[BM]$;
- Si le numéro obtenu est 5 ou 6, placer le milieu de $[CM]$.

Le milieu précédemment construit est renommé M.

Répéter 20 fois l'opération.

Attention de toujours repartir du dernier point placé et non pas du premier point choisi !

Que remarque-t-on au sujet de la disposition des points ?

On a utilisé un dé pour obtenir des résultats aléatoires.
Les points semblent-ils suivre un modèle précis ?

Dans le paragraphe suivant, on va utiliser un programme pour voir ce qui arrive lorsqu'on trace des milliers de points.

II. Le jeu du chaos ; programme Python

On va réaliser la figure précédente en utilisant un programme sur calculatrice.

Pour des raisons esthétiques, on décide de prendre le triangle ABC équilatéral et on se place dans un repère orthonormé du plan de sorte que les points A, B, C ont pour coordonnées respectives $(0;0)$, $(1;0)$, $\left(\frac{1}{2};\frac{\sqrt{3}}{2}\right)$.

On travaille dans la fenêtre graphique définie par $0 \leqslant x \leqslant 1$ et $0 \leqslant y \leqslant 1$.

On utilise le résultat :

Soit M et N deux points du plan muni d'un repère.
Le milieu du segment $[MN]$ a pour coordonnées $\left(\frac{x_M + x_N}{2} ; \frac{y_M + y_N}{2}\right)$.

Dans le programme, on utilise la formule des coordonnées d'un milieu sous la forme $\left(0,5\left(x_{\mathrm{M}}+x_{\mathrm{N}}\right);0,5\left(y_{\mathrm{M}}+y_{\mathrm{N}}\right)\right)$.

```
import numpy as np
import matplotlib.pyplot as plt
from random import randint

def midpoint(point1, point2):
    return [(point1[0] + point2[0])/2, (point1[1] + point2[1])/2]

v1 = [0,0]
v2 = [1,0]
v3 = [.5,np.sqrt(3)/2]

def sier(N):
    curr_point = [0,0]
    for _ in range(N):
            val = randint(0,2)
            if val == 0:
                    curr_point = midpoint(curr_point, v1)
            if val == 1:
                    curr_point = midpoint(curr_point, v2)
            if val == 2:
                    curr_point = midpoint(curr_point, v3)
            plt.scatter(curr_point[0],curr_point[1],s=1,c='b')
    return plt.show()
```

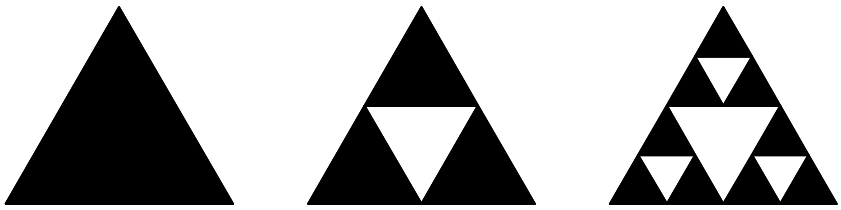
III. Le triangle de Sierpinski

Le triangle de Sierpiński, aussi appelé par Mandelbrot le joint de culasse de Sierpiński, est une fractale, du nom de Wacław Sierpiński.
Wacław Sierpiński (1882-1969) est un mathématicien polonais qui l'étudia en 1915.

Les fractales sont des objets géométriques complexes qui permettent notamment de désigner des formes qui existent depuis toujours dans la nature (chou romanesco, alvéoles pulmonaires, côtes bretonnes, flocons de neige). Le mot « fractale » a été créé par Benoît Mandelbrot à partir de la racine latine « fractus » signifiant à la fois irrégulier et brisé. Suite aux travaux de Mandelbrot et grâce au développement de l'informatique, on a assisté dans les années quatre-vingt à l'émergence d'une nouvelle branche des mathématiques : la géométrie fractale. Depuis leur invention en 1975 par Benoît Mandelbrot, les fractales font l'objet de nombreuses études.

Il est défini de la manière suivante : on part d'un triangle équilatéral que l'on partage en quatre triangles équilatéraux et dont on enlève le triangle central. On recommence l'opération pour les triangles restants et ainsi de suite. Les figures ci-dessous donnent le rang 0 et les trois premières.

On partage un triangle équilatéral en quatre triangles équilatéraux obtenus en traçant les segments joignant les milieux des côtés et on enlève le triangle central. Chaque triangle non noirci est alors divisé en quatre triangles équilatéraux selon le même procédé et on noircit le triangle central comme précédemment.



Triangle initial Étape 1 Étape 2

Pour aller plus loin : site de l'université de Boston (Boston University Arts & Sciences Mathematics & Statistics)
The Sierpinski triangle http://math.bu.edu/DYSYS/chaos-game/node2.html

Fin du cours sur triangle Sierpinski

Pour quitter le mode édition de programme, appuyer sur [2nde] , puis sur [mode] .

Introduction :

Le triangle de Sierpiński, aussi appelé par Mandelbrot le joint de culasse de Sierpiński, est une fractale, du nom de Wacław Sierpiński.
Wacław Sierpiński (1882-1969) est un mathématicien polonais qui l'étudia en 1915.

Les fractales sont des objets géométriques complexes qui permettent notamment de désigner des formes qui existent depuis toujours dans la nature (chou romanesco, alvéoles pulmonaires, côtes bretonnes, flocons de neige). Le mot « fractale » a été créé par Benoît Mandelbrot à partir de la racine latine « fractus » signifiant à la fois irrégulier et brisé. Suite aux travaux de Mandelbrot et grâce au développement de l'informatique, on a assisté dans les années quatre-vingt à l'émergence d'une nouvelle branche des mathématiques : la géométrie fractale. Depuis leur invention en 1975 par Benoît Mandelbrot, les fractales font l'objet de nombreuses études.

Le triangle de Sierpinski

Il est défini de la manière suivante : on part d'un triangle équilatéral que l'on partage en quatre triangles équilatéraux et dont on enlève le triangle central. On recommence l'opération pour les triangles restants et ainsi de suite. Les figures ci-dessous donnent le rang 0 et les trois premières

```
: FnOff
: ClrDraw
: PlotsOff
: AxesOff
: 0 → Xmin
: 1 → Xmax
: 0 →Ymin
: 1 → Ymax
: Prompt N
: rand → X
: rand → Y
: For(K,1,N)
: rand →N
: If N =< 3^-1   // utilisez direct la touche [x-1]
: Then
: .5X → X
: .5Y → Y
: End
```

: If 3^-1<N and N=< 2/3
: Then
: .5(X+.5 →X
: .5(Y+1 →Y
: End
: If N>2/3
: Then
: .5(X+1 → X
: .5Y → Y
: End
: Pt-On(X,Y
: End

Aide pour trouver quelques mots-clés et opérateurs

Commande	Chemin
EffDessin ou ClrDraw	DRAW – 1: EffDessin ou 1 : ClrDraw
→	STO>
Xmin, Xmax, Ymin, Ymax	VARS – 1:Window – (1, 2, 4 et 5)
Prompt	PRGM – E/S ou I/O – 2:Prompt
For(	PRGM – 4:For(
NbrAléat ou rand	MATH – PRB – 1: NbrAléat ou 1 : rand
If	PRGM – 1:If
<	TEST – 5:<
Then	PRGM – 2:Then
Else	PRGM – 3:Else
End	PRGM – 7:End
Pt-Aff(ou Pt-On(	DRAW – POINTS – 1: Pt-Aff(ou 1: Pt-On(

*

On peut parir de n’importe quell triangle.

On va partir du triangle ABC en prenant les coordonnées (0 ;0), B(0 ;1) …

Un travail en complexe gagnerait des ligne dans le programme.

Coquillage de Sierpinski

Premièrement, placez un point à un endroit de votre choix sur l’image ci-dessous où se trouvent déjà les points A, B et C.

Deuxièmement, placez une série de points selon les résultats aléatoires obtenus avec le dé que vous lancez :
- si vous obtenez 1 ou 2, vous placez un point à mi-chemin entre le dernier point posé et le point A ;
- si vous obtenez 3 ou 4, vous placez un point à mi-chemin entre le dernier point posé et le point B ;
- si vous obtenez 5 ou 6, vous placez un point à mi-chemin entre le dernier point posé et le point C.

Continuez jusqu’à ce que vous ayez placé 20 points sur l’image.

Attention de toujours repartir du dernier point que vous avez placé et non pas du premier point que vous avez choisi !

Programme pour calculatrice TI

Il faut premièrement enregistrer le format du graphique pour que le programme fonctionne quelque soit le graphique utilisé précédemment.
Pour cela, allez dans "fenêtre", puis saisir 0 comme valeur de "Xmin" et "Ymin", ainsi que la valeur 1 pour "Xmax" et "Ymax". Les autres valeurs n'ont pas d'importance.
Allez maintenant dans la section "dessin" (2nde prgm) et sélectionnez l'onglet "STO" à droite, puis prenez "EnrBDG".
Vous vous retrouvez maintenant sur l'écran de calcul avec " EnrBDG " et cliquez sur "1". Sélectionnez "entrer" ; "Done" apparaît à l'écran.

Le vendredi 6 janvier 2017 j'ai montré ce que ça donnait sur calculatrice.

Objectif : découvrir un ensemble de points compliqué qui s'appelle une fractale

Un élève m'a demandé : « Ça sert à quoi ? »

Version écrite à partir du 7-10-2017

Version pour TI-83 Premium CE

Vous pouvez maintenant recopier le programme qui est le suivant :

```
: RappelBDG 1 (il se trouve à la même section que EnrBDG)
: 0 → R
: Repeat R
: Prompt N
: 0.25 → X
: 0.5 → Y
: {X,Y} → L4
: {0,0} → L1
: {1,0} → L2
: {0.5,1} → L3
 : For(I,2,N)
: nbrAléatEnt(0,3) → C  [faire math puis PRB puis choix 5 ; ne remplir que les deux premières entrées ; appuyer sur entrer sans remplir la dernière entrée)
: If  C = 0
: Then
: L1 → L5
: Else
: If  C = 1
: Then
: L2 → L5
: Else
: L3 → L5
: End
: End
:  0.5 * L5 + 0.5 * L4  → L4
: L4(1) → X
: L4(2) → Y
: Pt-Aff(X,Y)
: GetKey → R   (Cela permet, avec le "Repeat R" au début, d'arrêter le programme lorsque l'on appuie sur une touche.)
: End
```

GetKey se trouve dans la section E/S rang 7 de la touche Prgm .

Remarque très importante :

On écrit L_1, L_2, L_3, L_4, L_5 dans le programme grâce aux touches de la calculatrice au-dessus des touches [1] , [2] , [3] , [4] , [5] .

Version pour TI-83 Premium CE

Vous pouvez maintenant recopier le programme qui est le suivant :

: RappelBDG 1 (il se trouve à la même section que EnrBDG)
: 0 → R
: Repeat R
: 0.5 → K
: 0.25 → X
: 0.5 → Y
: 10000 → N
: {X,Y} → L4
: {0,0} → L1
: {1,0} → L2
: {0.5,1} → L3
: For(I,2,N)
: nbrAléatEnt(0,3) → C [faire [math] puis PRB puis choix 5 ; ne remplir que les deux premières entrées ; appuyer sur [entrer] sans remplir la dernière entrée)
: If C = 0
: Then
: L1 → L5
: Else
: If C = 1
: Then
: L2 → L5
: Else
: L3 → L5
: End
: End
: $K*L5+(1-K)*L4 \rightarrow L4$
: L4(1) → X
: L4(2) → Y
: Pt-Aff(X,Y)
: GetKey → R (Cela permet, avec le "Repeat R" au début, d'arrêter le programme lorsque l'on appuie sur une touche.)
: End

GetKey se trouve dans la section E/S rang 7 de la touche [Prgm] .

Remarque très importante :

On écrit L1, L2, L3, L4, L5 dans le programme grâce aux touches de la calculatrice au-dessus des touches [1] , [2] , [3] , [4] , [5] .

Version ancienne :

Il faut premièrement enregistrer le format du graphique pour que le programme fonctionne quelque soit le graphique utilisé précédemment.
Pour cela, allez dans "fenêtre", puis saisir 0 comme valeur de "Xmin" et "Ymin", ainsi que la valeur 1 pour "Xmax" et "Ymax". Les autres valeurs n'ont pas d'importance.
Allez maintenant dans la section "dessin" (2nde prgm) et sélectionnez l'onglet "STO" à droite, puis prenez "StoreGBD".
Vous vous retrouvez maintenant sur l'écran de calcul avec "StoreGBD" et cliquez sur "1". Sélectionnez "entrer" ; "Done" apparaît à l'écran.

Vous pouvez maintenant recopier le programme qui est le suivant :

: RecallGBD 1 (il se trouve à la même section que StoreGBD)
: 0 → R
: Repeat R
: 0.5 → K
: 0.25 → X
: 0.5 → Y
: 10000 → N
: {X,Y} → L4
: {0,0} → L1
: {1,0} → L2
: {0.5,1} → L3
: For(I,2,N)
: entAléat(0,3) → C [faire math puis PRB puis choix 5 : entAléat(; séparer le 0 et le 3 par une virgule et non par un point)
: If C = 0
: Then
: L1 → L5
: Else
: If C = 1
: Then
: L2 → L5
: Else
: L3 → L5
: End
: End
: $K*L5+(1-K)*L4 \rightarrow L4$
: L4(1) → X
: L4(2) → Y
: Pt-On(X,Y)
: GetKey → R (Cela permet, avec le "Repeat R" au début, d'arrêter le programme lorsque l'on appuie sur une touche.)
: End

Remarque très importante :

On doit taper L1, L2, L3 grâce aux touches de la calculatrice au-dessus des touches 1, 2, 3…

Théo Manfredi

Bonjour monsieur,
voici donc le programme des fractales ainsi qu'un programme assez puissant pour obtenir les diviseurs naturels d'un entier.
J'ai cependant eu des petits soucis avec Géogébra donc je n'ai pas l’horloge pour le moment mais je vous l'enverrai ce week-end.
Je préfère détailler les étapes pour réaliser les programmes pour éviter que vous n'ayez de problèmes. De plus, ma calculatrice est en Anglais.

Fractales :

Il faut premièrement enregistrer le format du graphique pour que le programme fonctionne quelque soit le graphique utilisé précédemment.
Pour cela, allez dans "fenêtre", puis saisir 0 comme valeur de "Xmin" et "Ymin", ainsi que la valeur 1 pour "Xmax" et "Ymax". (Les autres valeurs n'ont pas d'importance)
Allez maintenant dans la section "dessin" (2nde + prgm) et sélectionnez l'onglet "STO" à droite, puis prenez "StoreGBD".
Vous vous retrouvez maintenant sur l'écran de calcul avec "StoreGBD" et cliquez sur "1". Sélectionnez "entrer" ; "Done" apparaît à l'écran.

Vous pouvez maintenant recopier le programme qui est le suivant :

: RecallGBD 1 (il se trouve à la même section que StoreGBD)
: 0 → R
: Repeat R
: 0.5 → K
: 0.25 → X
: 0.5 → Y
: 10000 → N
: {X,Y} → L4
: {0,0} → L1
: {1,0} → L2
: {0.5,1} → L3
: For(I,2,N)
: randInt(0,3) → C
: If C=0
: Then
: L1 → L5
: Else
: If C=1
: Then
: L2 → L5
: Else
: L3 → L5
: End
: End
: $K*L5+(1-K)*L4 \rightarrow L4$
: L4(1) → X
: L4(2) → Y
: Pt-On(X,Y)
: GetKey → R (Cela permet, avec le "Repeat R" au début, d'arrêter le programme lorsque l'on appuie sur une touche.)
: End

Diviseurs :

```
Input "N=",N
1-->dim(L1)
1-->dim(L2)
0-->L1(1)
0-->L2(1)
For(I,1,iPart(√(N)
If remainder(N,I)=0
Then
augment(L1,{I})-->L1
augment({N/I},L2)-->L2
End
End
augment(L1,L2)-->L1
```
(Ce programme ne nécessite pas d'affichage de valeurs car cette dernière ligne va automatiquement afficher la Liste 1)

La liste L1 est composée de tous les diviseurs naturels du nombre N entré et comporte un 0 au début et à la fin de la liste (cela peut être corrigé mais alourdiraitle programme).

Si vous rencontrez des erreurs, n'hésitez pas à me contacter.
Merci aussi de me prévenir lorsque vous recevez ce mail pour que je puisse valider votre adresse mail.

Cordialement,
Théo MANFREDI, TS2

Voici ce que j'ai trouvé sur Internet :

TI-83 Manuel d'utilisation - Texas Instruments

Ce programme dessine un fractal célèbre, le triangle de Sierpinski, et le mémorise sous forme d'image. Pour commencer, appuyez sur ~ ~ 1. Nommez le programme SIERPINS et appuyez sur Í. L'éditeur de programme s'affiche. PROGRAM:SIERPINS :FnOff :ClrDraw :PlotsOff :AxesOff :0!Xmin:1!Xmax :0!Ymin:1!Ymax Choix des paramètres WINDOW :rand!X:rand!Y :For(K,1,3000) :rand!N Début du groupe For :If N 1à3 :Then :.5X!X :.5Y!Y :End Groupe If/Then :If 1à3

PROGRAM:SIERPINS :FnOff :ClrDraw :PlotsOff :AxesOff :0!Xmin:1!Xmax :0!Ymin:1!Ymax Choix des paramètres WINDOW :rand!X:rand!Y :For(K,1,3000) :rand!N Début du groupe For :If N 1à3 :Then :.5X!X :.5Y!Y :End Groupe If/Then :If 1à3

```
:FnOff :ClrDraw
:PlotsOff :AxesOff
:0!Xmin:1!Xmax
:rand!X
:rand!Y
:For(K,1,3000)
:rand!N
:If N 1à3
:Then
:.5X!X
:.5Y!Y
:End
```

:If 1à3
:Then
:.5(.5+X)!X
:.5(1+Y)!Y
:End
:If 2à3
:Then :.5(1+X)!X
:.5Y!Y
:End
:Pt-On(X,Y) Dessin d'un point :End Fin du groupe For :StorePic 6

Après avoir exécuté ce programme, vous pouvez rappeler et afficher le dessin à l'aide de l'instruction RecallPic 6.

Nicolas Leroy m'a écrit :

Programme triangle de Sierpinski

Store GBD = Enr BDG
Recall GBD = Rappel BDG
Ran α Int = nBC Aléat Ent
Se trouve dans la section « Prob »de la touche math. Ne remplir que les deux premières entrées. Appuyer sur entrer sans remplir la derniére entrée.
Attention à bien utiliser les touches L_1, L_2, L_3, L_4, L_5 de la calculatrice.
Pt-on =Pt Aff
Get key se trouve dans la section E/S rang 7 de la touche Prgrm

Voici ce que m'a écrit Nicolas Leroy sur une feuille :

Programme triangle de Sierpinski

Store GBD = Enr BDG

Recall GBD = Rappel BDG

RandInt = nbr AléatEnt
se trouve dans la section « PROB » de la touche math . Ne remplir que les deux premières entrées.
Appuyer sur entrer sans remplir la dernière entrée.

Attention à bien utiliser les touches L1, L2, L3, L4, L5 de la calculatrice

Pt-on = Pt-Aff

get key se trouve dans la section E/S rang 7 de la touche programme.

Le 23-6-2017

APMEP

Depuis leur invention en 1975 par Benoît Mandelbrot, les fractales font l'objet de nombreuses études. Leur utilisation dans l'enseignement présente de nombreux avantages : elle peut se faire à tous les niveaux du collège, du lycée voire de l'université, elle permet comme l'expliquent les auteurs de [LW] de faire des liens entre différentes parties des mathématiques, de donner des exemples concrets d'application des mathématiques dans le monde extérieur, de susciter l'intérêt des élèves par la beauté et la complexité des figures fractales. Ces avantages ont été exploités plusieurs fois dans la littérature comme en témoignent plusieurs articles pédagogiques ([BAFP], [Ga], [GG] par exemple). Dans ce travail, nous avons cherché à les utiliser afin d'enseigner l'algorithmique au lycée. Nous présentons des activités qui allient géométrie et informatique, et qui sont basées sur les fractales de Sierpinski. Ces activités sont en conformité avec les programmes de mathématiques d'autant plus que ces derniers introduisent depuis la réforme de 2009 l'algorithmique dès la classe de seconde. Après avoir présenté et défini les fractales de Sierpinski, nous précisons les prérequis et les modalités des activités proposées. Nous élaborons des algorithmes permettant la construction du triangle de Sierpinski. Nous expliquons et employons la méthode récursive très simple à implémenter et bien adaptée à la géométrie fractale. Nous choisissons de traduire nos algorithmes en Python, mais en réalité les programmes obtenus sont facilement adaptables à tout langage de programmation admettant la récursivité. Enfin, nous finissons par des activités classiques sur un tableur de type Excel conduisant au calcul du périmètre et de l'aire des fractales.

I. Présentation des fractales de Sierpinski

1. Un peu d'histoire

Les fractales sont des objets géométriques complexes qui permettent notamment de désigner des formes qui existent depuis toujours dans la nature (chou romanesco, alvéoles pulmonaires, côtes bretonnes, flocons de neige). Le mot « fractale » a été créé par Benoît Mandelbrot [Ma] à partir de la racine latine « fractus » signifiant à la fois irrégulier et brisé. Suite aux travaux de Mandelbrot et grâce au développement de l'informatique, on a assisté dans les années quatre-vingt à l'émergence d'une nouvelle branche des mathématiques : la géométrie fractale.

Plusieurs mathématiciens précurseurs ont contribué à la naissance de la géométrie fractale :

En 1500, le graveur et géomètre Albrecht Dürer invente la première figure fractale connue sous le nom de pentagone de Dürer.  _ Il s'agit de placer dans un pentagone six pentagones adjacents en laissant de côté le pentagone central et en recommençant le partage pour les pentagones restants.

Vers 1700, Leibniz introduit et étudie la propriété d'autosimilarité. Cette propriété est l'une des plus importantes propriétés théoriques des objets fractals.

En 1904, Von Koch propose une courbe désormais connue sous le nom de flocon de Von Koch. Cette courbe a permis d'établir l'existence de fonctions continues mais non dérivables en tout point, fonctions pathologiques, considérées d'abord comme des monstres mathématiques et qui se sont avérées utiles pour définir de manière rigoureuse les règles du calcul infinitésimal.

En 1915, le mathématicien polonais Sierpinski introduit le triangle qui porte désormais son nom et que l'on va définir plus bas.

En 1925, apparaît la première représentation graphique d'un ensemble de Julia. Rappelons que les ensembles de Julia sont des sous-ensembles du plan complexe associés à des suites complexes (zn)(zn) satisfaisant une relation de récurrence de la forme $z_{n+1}=z_n^2+c$

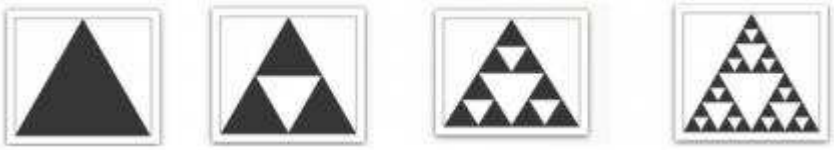
Abandonnés pendant de nombreuses années, les travaux de Julia ont été remis à l'honneur essentiellement grâce à Benoît Mandelbrot puis à Adrien Douady. Ce dernier a notamment établi les propriétés fondamentales de l'ensemble de Mandelbrot qu'il a lui-même défini et baptisé, et qui est une mine de problèmes ouverts difficiles.

D'autres mathématiciens ont donné leur nom à des fractales (on peut citer Peano, Cantor, Hilbert).

Notons que les fractales ont de nombreuses applications au sein des mathématiques et également dans des domaines aussi variés que la biologie, la physique, l'astrophysique, la géophysique, l'étude des phénomènes naturels, l'industrie, les finances, l'art (voir aussi [Ta]). Enfin, elles interviennent dans la théorie du chaos qui présente elle-même de nombreuses applications en chimie et en mécanique des fluides.

2. Le triangle de Sierpinski

Il est défini de la manière suivante : on part d'un triangle équilatéral que l'on partage en quatre triangles équilatéraux et dont on enlève le triangle central. On recommence l'opération pour les triangles restants et ainsi de

suite. Les figures ci-dessous donnent le rang 0 et les trois premières étapes : 

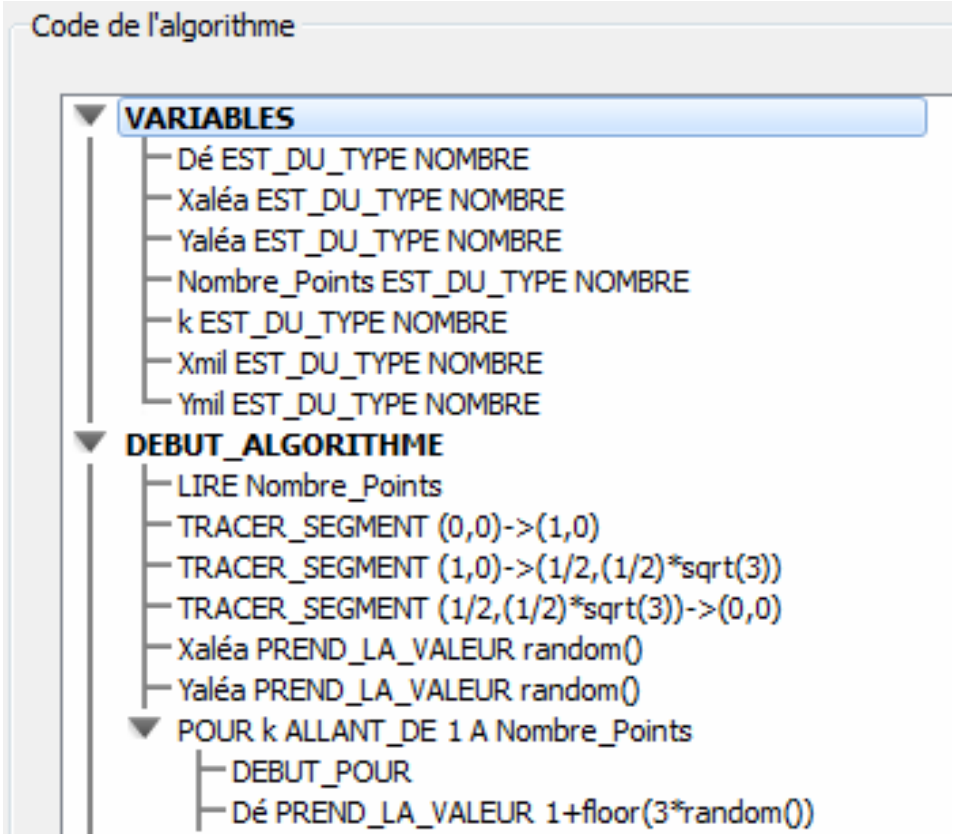
II.

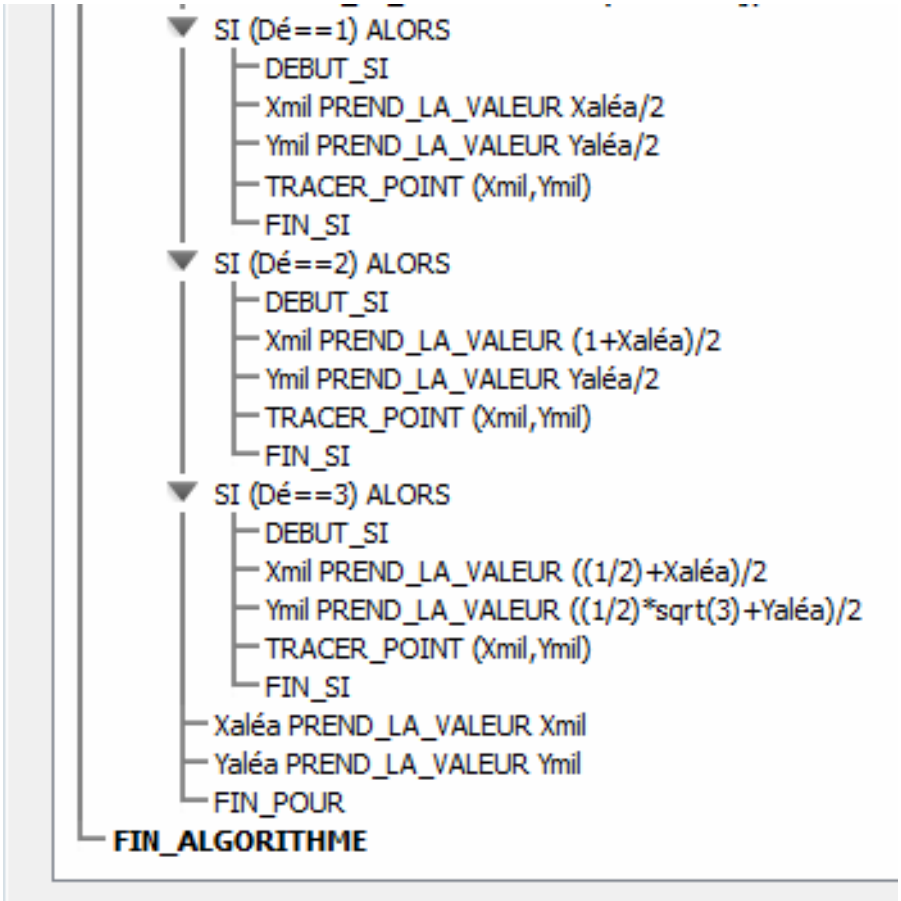
Doc de Boris Hanus

Réaliser un algorithme qui :

- Construit un triangle équilatéral (de coté 1)
- Choisit un point M de coordonnées aléatoire dans la fenêtre graphique (le carré de coté 1) et qui, en fonction d'un choix aléatoire (équiprobable), va construire le milieu de ce point M et d'un de 3 sommets du triangle.
- Le milieu précédemment construit est renommé M.
- Répéter n fois cette opération.

Code de l'algorithme





Le 23-6-2017

J’ai trouvé sur le site le programme

http://tout82.free.fr/forum/sujet.php?sujet=3138

```
:FnOff
:ClrDraw
:PlotsOff
:AxesOff
:0->Xmin
:1->Xmax
:0->Ymin
:1->Ymax
:rand->X
:rand->Y
:For(K,1,3E3    // 3E3=3000
:rand->N
:If N =< 3^-1   // utilisez direct la touche [x-1]
:Then
:.5X->X
:.5Y->Y
:End
:If 3^-1<N and N=< 2/3
:Then
:.5(X+.5->X
:.5(Y+1->Y
:End
:If N>2/3
:Then
:.5(X+1->X
:.5Y->Y
:End
:Pt-On(X,Y
:End
```

	Commandes		Explication
Ligne	Français	Anglais	
1	EffDessin	ClrDraw	Efface le dessin
2	0→Xmin	0→Xmin	Précise la valeur minimale de X pour le graphique (0)
3	1→Xmax	1→Xmax	Précise la valeur maximale de X pour le graphique (1)
4	0→Ymin	0→Ymin	Précise la valeur minimale de Y pour le graphique (0)
5	1→Ymax	1→Ymax	Précise la valeur maximale de Y pour le graphique (1)
6	Prompt A	Prompt A	Demande à l’utilisateur d’entrer un A (entre 0 et 1)
7	Prompt B	Prompt B	Demande à l’utilisateur d’entrer un B (entre 0 et 1)
8	For(C, 1, 3000)	For(C, 1,	Fera la séquence, jusqu’à son « END» (de la ligne 9 à la ligne
9	NbrAléat→N	rand→N	Place un nombre aléatoire (toujours entre 0 et 1) dans la
10	If N<.3333	If N<.3333	Si la valeur aléatoire est plus petite que .3333
11	Then	Then	Alors faire la séquence jusqu’à la ligne 13
12	.5(A+1) →A	.5(A+1) →A	Modifie la valeur de A
13	.5B→B	.5B→B	Modifie la valeur de B
14	Else	Else	Sinon (donc si la valeur de N n’est pas inférieure à .3333)
15	If N<.6666	If N<.6666	Si la valeur aléatoire est plus petite que .6666
16	Then	Then	Alors faire la séquence jusqu’à la ligne 18

17	.5A→A	.5A→A	Modifie la valeur de A
18	.5B→B	.5B→B	Modifie la valeur de B
19	Else	Else	Sinon (si la valeur de N n'est pas inférieure à .6666), faire la séquence jusqu'à la ligne 22
20	.5(A+.5) →A	.5(A+.5) →A	Modifie la valeur de A
21	.5(B+1) →B	.5(B+1) →B	Modifie la valeur de B
22	End	End	Termine la séquence commencée à 15
23	End	End	Termine la séquence commencée à 10
24	Pt-Aff(A,B)	Pt-On(A,B)	Affiche un point à la coordonnée A, B
25	End	End	Termine la séquence du « for», commencée à 8

Site conseillé : The Sierpinski triangle.

http://math.bu.edu/DYSYS/chaos-game/node2.html

Le mardi 8-8-2017

I. Le jeu du chaos (algorithme de constructions géométriques « à la main »)

Au départ :

On dispose d'un dé cubique dont les faces sont numérotées de 1 à 6.

Sur une feuille, placer trois points quelconques A, B, C non alignés.

Placer ensuite point quelconque sur la feuille.

Ensuite :

Placer une série de points selon les résultats aléatoires obtenus avec le dé que l'on lance :
• si vous obtenez 1 ou 2, vous placez un point à mi-chemin entre le dernier point posé et le point A ;
• si vous obtenez 3 ou 4, vous placez un point à mi-chemin entre le dernier point posé et le point B ;
• si vous obtenez 5 ou 6, vous placez un point à mi-chemin entre le dernier point posé et le point C.
o
Continuer jusqu'à ce que vous ayez placé 20 points sur l'image.

Attention de toujours repartir du dernier point que vous avez placé et non pas du premier point que vous avez choisi !

Que remarquez-vous au sujet de la disposition des points ?
b) Vous avez utilisé un dé pour obtenir des résultats aléatoires.
Ces points semblent-ils suivre un modèle précis ?
Maintenant, avec l'aide de votre enseignant et des annexes de l'activité, vous pouvez utiliser la force des outils technologiques pour voir ce qui arrive lorsqu'on trace des milliers de points.
Étonnant, n'est-ce pas ?

Le mardi 3 octobre 2017

J'enlève :

Commande	Chemin
EffDessin ou ClrDraw	DRAW – 1: EffDessin ou 1 : ClrDraw
→	STO>
Xmin, Xmax, Ymin, Ymax	VARS – 1:Window – (1, 2, 4 et 5)
Prompt	PRGM – E/S ou I/O – 2:Prompt
For(	PRGM – 4:For(
NbrAléat ou rand	MATH – PRB – 1: NbrAléat ou 1 : rand
If	PRGM – 1:If
<	TEST – 5:<
Then	PRGM – 2:Then
Else	PRGM – 3:Else
End	PRGM – 7:End
Pt-Aff(ou Pt-On(	DRAW – POINTS – 1: Pt-Aff(ou 1: Pt-On(

Le 4-4-2020

http://bopace.github.io/python/2016/06/09/python-turtle-sierpinski/

```
import numpy as np
import pylab
from random import randint

def midpoint(point1, point2):
    return [(point1[0] + point2[0])/2, (point1[1] + point2[1])/2]

curr_point = [0,0]  # our seed value for the chaos game
                    # It can fall anywhere inside the triangle

# our equilateral triangle vertices
v1 = [0,0]
v2 = [1,0]
v3 = [.5,np.sqrt(3)/2]

# Plot 5000 points
for _ in range(5000):
    # choose a triangle vertex at random
    # set the current point to be the midpoint
    # between the previous current point and
    # the randomly chosen vertex
    val = randint(0,2)
```

```
    if val == 0:
        curr_point = midpoint(curr_point, v1)
    if val == 1:
        curr_point = midpoint(curr_point, v2)
    if val == 2:
        curr_point = midpoint(curr_point, v3)
    # plot the new current point
    pylab.plot(curr_point[0],curr_point[1],'m.',markersize=2)


pylab.show()
```

Le lendemain 5-4-2020, j'ai utilisé le site :
https://www.codingame.com/playgrounds/17176/recueil-dexercices-pour-apprendre-python-au-lycee/cours---representation-graphique-avec-matplotlib
et ça marche.

Site Numworks Loupiot

```
def triangle(N_i):
  rgb=color(0,0,0)
  x,y=0.5,0
  n=0
  while n<N_i:
    A=randint(0,2)
    if A==0:
      mx,my=0,0
    if A==1:
      mx,my=1,0
    if A==2:
      mx,my=0.5,1
    x,y=(x+mx)/2,(y+my)/2
    set_pixel(int(x*320),222-int(y*222),rgb)
    n+=1
```

APMEP

Algorithmique et programmation graphique des fractales de Sierpinski
Angela Gammella-Mathieu [1] & Nicolas Mathieu [2]
Introduction
Depuis leur invention en 1975 par Benoît Mandelbrot, les fractales font l'objet de nombreus

https://www.quirin.science/python/un-peu-daleatoire-ii
Un peu d'aléatoire II

Un peu d'aléatoire II

```
import random as rd
```

```
import numpy as np
import matplotlib.pyplot as plt

## Jeu du chaos

def polygone(n):
    x= [2*k*np.pi / n for k in range(n)]
    return np.cos(x),np.sin(x)

# x,y = polygone(10000)
# plt.plot(x,y,linestyle='',marker = '.')
# plt.axis('equal')
# plt.show()

def chaos(n,N):
    px,py = polygone(n)
    abs = [0]
    ord = [0]
    for _ in range(N):
        sommet = rd.randint(0,n-1)
        x = (abs[-1] + px[sommet])/2
        y = (ord[-1] + py[sommet])/2
        abs.append(x); ord.append(y)
    plt.clf()
    plt.axis('equal')
    plt.plot(abs,ord,marker=',',linestyle='')
    plt.draw(); plt.show()

def chaos2(n,N):
    px,py = polygone(n)
    abs = [0]
    ord = [0]
    mem=-1
    sommet = rd.randint(0,n-1)
    for _ in range(N):
        while sommet == mem:
            sommet = rd.randint(0,n-1)
        mem = sommet
        x = (abs[-1] + px[sommet])/2
        y = (ord[-1] + py[sommet])/2
        abs.append(x); ord.append(y)
    plt.clf()
    plt.axis('equal')
    plt.plot(abs,ord,marker=',',linestyle='')
    plt.draw(); plt.show()
```

Très bon site :

odingame.com/playgrounds/17176/recueil-dexercices-pour-apprendre-python-au-lycee/cours---representation-graphique-avec-matplotlib

```
import matplotlib.pyplot as plt
import numpy as np

plt.plot([1,3,4],[2,1,6])

plt.show()
```

Pour visualiser : bac à sable

https://www.codingame.com/playgrounds/17176/recueil-dexercices-pour-apprendre-python-au-lycee/cours---representation-graphique-avec-matplotlib

Le 11-5-2020

```
import numpy as np
import matplotlib.pyplot as plt
from random import randint

def midpoint(point1, point2):
    return [(point1[0] + point2[0])/2, (point1[1] + point2[1])/2]


# our equilateral triangle vertices
v1 = [0,0]
v2 = [1,0]
v3 = [.5,np.sqrt(3)/2]

def sier(N):
    curr_point = [0,0]

    for _ in range(N):
            val = randint(0,2)
            if val == 0:
                    curr_point = midpoint(curr_point, v1)
            if val == 1:
                    curr_point = midpoint(curr_point, v2)
            if val == 2:
                    curr_point = midpoint(curr_point, v3)
            plt.scatter(curr_point[0],curr_point[1],s=1,c='b')
    return plt.show()

sier(2000)
```

```
import numpy as np
import pylab
from random import randint

def midpoint(point1, point2):
    return [(point1[0] + point2[0])/2, (point1[1] + point2[1])/2]

curr_point = [0,0]  # our seed value for the chaos game
                    # It can fall anywhere inside the triangle

# our equilateral triangle vertices
v1 = [0,0]
v2 = [1,0]
v3 = [.5,np.sqrt(3)/2]

# Plot 5000 points
for _ in range(5000):
    # choose a triangle vertex at random
    # set the current point to be the midpoint
    # between the previous current point and
    # the randomly chosen vertex
```

```
    val = randint(0,2)
    if val == 0:
        curr_point = midpoint(curr_point, v1)
    if val == 1:
        curr_point = midpoint(curr_point, v2)
    if val == 2:
        curr_point = midpoint(curr_point, v3)
    # plot the new current point
    pylab.plot(curr_point[0],curr_point[1],'m.',markersize=2)

pylab.show()
```

II. Le jeu du chaos (programme sur calculatrice TI)

On va réaliser la figure précédente en utilisant un programme sur calculatrice.
On se place dans un repère du plan de sorte que les points A, B, C ont pour coordonnées respectives $(0\,;0)$, $(1\,;0)$, $(0{,}5\,;1)$.
On travaille dans la fenêtre graphique définie par $0 \leqslant x \leqslant 1$ et $0 \leqslant y \leqslant 1$.

On utilise le résultat :

Soit M et N deux points du plan muni d'un repère.
Le milieu du segment $[\text{MN}]$ a pour coordonnées $\left(\dfrac{x_\text{M}+x_\text{N}}{2}\,;\dfrac{y_\text{M}+y_\text{N}}{2}\right)$.

Dans le programme, on utilise la formule des coordonnées d'un milieu sous la forme $\left(0{,}5\left(x_\text{M}+x_\text{N}\right)\,;0{,}5\left(y_\text{M}+y_\text{N}\right)\right)$.

Attention, les variables A, B, C du programme suivant n'ont rien à voir avec les points A, B, C.

gne	Commande		Explication
	Français	Anglais	
	EffDessin	ClrDraw	Efface le dessin
	0→Xmin	0→Xmin	Précise la valeur minimale de X pour le graphique (0)
	1→Xmax	1→Xmax	Précise la valeur maximale de X pour le graphique (1)
	0→Ymin	0→Ymin	Précise la valeur minimale de Y pour le graphique (0)
	1→Ymax	1→Ymax	Précise la valeur maximale de Y pour le graphique (1)
	Prompt A	Prompt A	Demande à l'utilisateur d'entrer un A (entre 0 et 1)
	Prompt B	Prompt B	Demande à l'utilisateur d'entrer un B (entre 0 et 1)
	Prompt N	Prompt N	Demande à l'utilisateur d'entrer une valeur de N (nombre de points à placer)
	For(I,1,N)	For(I,1,N)	Fera la séquence, jusqu'à son « End » (de la ligne 10 à la ligne 26), N foi
	nbrAléatEnt(1,3) → C		Place un entier aléatoire égal à 1, 2 ou 3 dans la variable C
	If C = 1	If C = 0	Si la valeur aléatoire est égale à 1
	Then	Then	Alors faire la séquence jusqu'à la ligne 13
	0.5A → A	.5(A+1) → A	Modifie la valeur de A
	0.5B → B	.5B → B	Modifie la valeur de B
	End	End	Termine la séquence commencée à 11
	If C = 2	If C = 1	Si la valeur aléatoire est égale à 2
	Then	Then	Alors faire la séquence jusqu'à la ligne 18
	0.5(A+1) → A	.5A → A	Modifie la valeur de A
	0.5B → B	.5B → B	Modifie la valeur de B
	End		Termine la séquence commencée à 16
	If C = 3	If C = 2	Si la valeur aléatoire est égale à 3

	Then	Then	Alors faire la séquence jusqu'à la ligne 18
	0.5(A+0.5) →A	.5(A+.5) →A	Modifie la valeur de A
	0.5(B+1) →B	.5(B+1) →B	Modifie la valeur de B
	End	End	Termine la séquence commencée à 21
	Pt-Aff(A,B)	Pt-On(A,B)	Affiche le point de coordonnées A, B
	End	End	Termine la séquence du « For », commencée à 9

Aide pour trouver quelques mots-clés et opérateurs

Commande	Chemin
EffDessin ou ClrDraw	DRAW – 1: EffDessin ou 1 : ClrDraw
→	STO>
Xmin, Xmax, Ymin, Ymax	VARS – 1:Window – (1, 2, 4 et 5)
Prompt	PRGM – E/S ou I/O – 2:Prompt
For(	PRGM – 4:For(
nbrAléatEnt	math choisir PROB 5 : nbrAléatEnt
If	PRGM – 1:If
Then	PRGM – 2:Then
Else	PRGM – 3:Else
End	PRGM – 7:End
Pt-Aff(ou Pt-On(	DRAW – POINTS – 1: Pt-Aff(ou 1: Pt-On(

Commande nbrAléatEnt :

math choisir PROB 5 : nbrAléatEnt

borninf :
bornsup :
n :
coller

Ici :
borninf : 1
bornsup : 2
n : 1

Entrer des valeurs de A et B comprises entre 0 et 1 par exemple (0,4 et 0,7).
Faire tourner le programme pour différentes valeurs de N ($N = 50$, $N = 100$, $N = 1000$, $N = 3000$).

```
import numpy as np
import matplotlib.pyplot as plt
from random import randint

def midpoint(point1, point2):
    return [(point1[0] + point2[0])/2, (point1[1] + point2[1])/2]


# our equilateral triangle vertices
v1 = [0,0]
v2 = [1,0]
v3 = [.5,np.sqrt(3)/2]

def sier(N):
    curr_point = [0,0]
    for _ in range(N):
            val = randint(0,2)
            if val == 0:
                    curr_point = midpoint(curr_point, v1)
            if val == 1:
                    curr_point = midpoint(curr_point, v2)
            if val == 2:
                    curr_point = midpoint(curr_point, v3)
            plt.scatter(curr_point[0],curr_point[1],s=1,c='b')
    return plt.show()
```

Le 16-10-2020

Je change

On va réaliser la figure précédente en utilisant un programme sur calculatrice.
On se place dans un repère orthonormé du plan de sorte que les points A, B, C ont pour coordonnées respectives $(0;0)$, $(1;0)$, $\left(\frac{1}{2};\frac{\sqrt{3}}{2}\right)$.

On va réaliser la figure précédente en utilisant un programme sur calculatrice.
On se place dans un repère orthonormé du plan de sorte que les points A, B, C ont pour coordonnées respectives $(0;0)$, $(1;0)$, $\left(\frac{1}{2};\frac{\sqrt{3}}{2}\right)$.

Que remarque-t-on au sujet de la disposition des points ?

On a utilisé un dé pour obtenir des résultats aléatoires.
Les points semblent-ils suivre un modèle précis ?

Maintenant, avec l’aide de votre enseignant et des annexes de l’activité, vous pouvez utiliser la force des outils technologiques pour voir ce qui arrive lorsqu’on trace des milliers de points. Étonnant, n’est-ce pas ?