

Fractales Sonores

Introduction

Les fractales sont des entités topologiques dont les propriétés sont similaires indépendamment de l'échelle à laquelle elles sont observées. Certaines sont parfaitement régulières et d'autres incluent des éléments aléatoires. Les images fractales donnent parfois des effets très impressionnants en révélant l'allure des processus mathématiques sous jacents. Les fractales de Mandelbrot peuvent être fascinantes mais des systèmes très simples exhibent déjà des propriétés fabuleuses. Le flocon de Koch par exemple est une ligne fermée sur un plan et occupe donc une surface finie mais le périmètre de cet objet a une longueur infinie! La construction d'un flocon de Koch se fait en appliquant une récurrence simple en remplaçant chaque segment de ligne d'une forme de base par un patron. La longueur de la ligne résultante est infinie puisque pour mesurer un segment quelconque, il est nécessaire d'appliquer la récurrence... L'aire de l'objet est inscrit dans un cadre et la forme est fermée, elle est donc nécessairement finie et il existe en fait une approche analytique à son calcul.

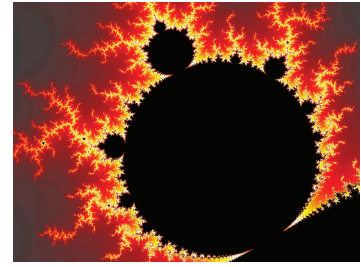
Une forme plus simple des structures de Koch est la ligne de Koch. Le processus de construction est similaire mais le cas de base est une ligne plutôt qu'un triangle. (Les figures continuent sur les pages suivantes). Pour dessiner une ligne de Koch, on écrit un programme récursif qui transforme les coordonnées de paires de points successifs. Étant donnée une ligne, on calcule le niveau 1 de la ligne de Koch. Puis, à partir de chaque segment de la ligne de niveau 1, on calcule la ligne de niveau 2 en créant de nouveaux points de la même manière. Ainsi de suite jusqu'à ce que le niveau cible soit atteint. À partir de la paire de points (A,B), on trouve la coordonnée des points C, D et E comme suit: C est projeté d'une distance h à angle droit à partir du centre du segment (A,B), D est au premier tiers et E est au second tiers sur le segment.

Votre travail consiste à faire une variation sur ce thème. Plutôt que de faire le dessin d'une ligne de Koch, vous allez faire un programme qui génère de la «musique de Koch». La sortie de votre programme sera un fichier MIDI qui jouera de la musique au piano. Dans ce contexte, l'axe vertical (la hauteur) donnera la note à jouer. L'axe horizontal représente le temps qui passe. Pour nous simplifier un peu la vie, le calcul du point C est modifié tel qu'illustré à la figure «Musique de Koch».

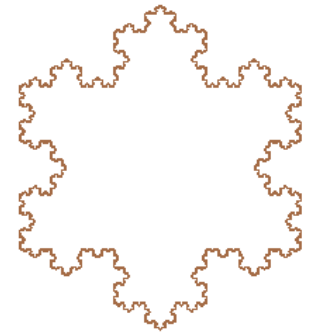
La récursion

Écrivez une fonction récursive `musique(A,B,niveau,niveauMax)` qui agit comme suit. Si `niveauMax == niveau`, alors la fonction émet une note dont la durée est la distance entre A et B sur l'axe horizontal. La hauteur de ce son est celle du point A. La durée de cette note correspond à la longueur du segment (A,B) sur l'axe horizontal. Si `niveau < niveauMax`, alors la fonction trouve la position des points C, D et E et s'invoque récursivement sur les segments (A,D), (D,C), (C,E) et (E,B).

La hauteur du son. Un piano de concert présente 88 touches, soit 7 groupes de 12 sons séparés par un demi-ton chacun. Vers le centre du clavier se trouve la touche du



Mandelbrot



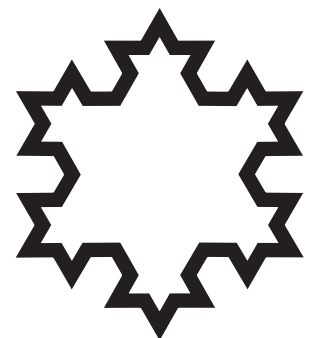
Flocon de Koch



Base et Patron



Niveau 1



Niveau 2

Do-moyen (middle-C en anglais) son numéro midi est 0x3C. Nous allons faire progresser notre fractale sur une partie de cette étendue s'étalant sur 48 sons à partir de deux octaves plus bas (midi 0x24) et jusqu'à deux octaves plus haut (midi 0x54) que le Do moyen. Ainsi, initialement la récurrence sera invoquée sur les points A et B dont la valeur sera 0x24. La hauteur (h sur le dessin «Musique de Koch») aura donc la valeur $0x54 - 0x3C == 0x30$ (décimal 48).

La valeur de h . Votre récurrence utilisera la fonction suivante pour trouver h . $h = \text{trunc}(48/2^n)$ où $\text{trunc}()$ est la fonction de troncature qui 'oublie' les éléments fractionnaires d'un nombre et où n est le niveau de la récurrence commençant avec la valeur 0 tel que montré sur les dessins à droite. Ainsi, au niveau 0, $h=48$ alors que au niveau 3, $h=6$ et au niveau 5, $h=1$.

Le volume du son est fixé à 0x60, le volume mitoyen.

La durée de la croche est fixée à 128 ticks MIDI tel que prescrit dans l'exemple sur le site web donné dans la section suivante.



Niveau 0



Niveau 1



Niveau 2

Format MIDI

Le format MIDI peut être assez compliqué mais nous n'utiliserons que des fonctionnalités de base. Consultez le site web <http://www.skytopia.com/project/articles/midi.html> pour les détails. Utilisez le format MIDI 0 afin de garder les choses le plus simple possible (ou MIDI 1 si cela vous tente).

Si ce travail vous semble trop facile, vous êtes invités à ajouter une piste MIDI supplémentaire où vous ajouterez un rythme de batterie pour accompagner votre musique.

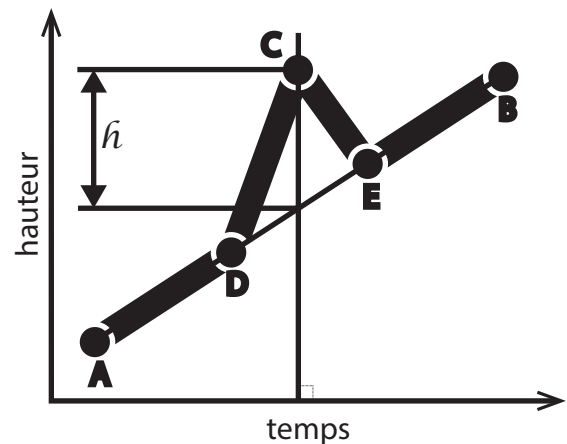
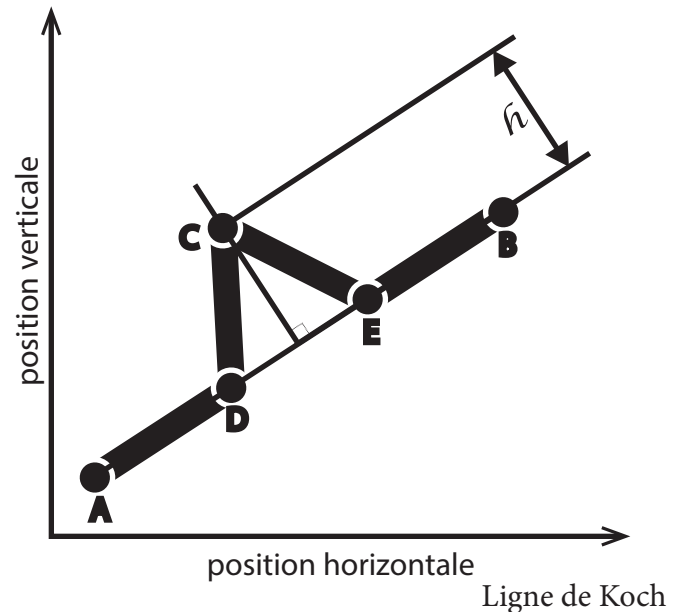
Votre programme

Votre programme doit se nommer 'fractales.py' et générer le fichier musique.midi.

Définissez une fonction récursive musique(A, B, niveau, niveauMax) avec (A,B) les points du segment à traiter, niveau est le niveau courant de la ligne/musique de Koch et niveauMax est le niveau maximal de la récursion. Cette fonction calcule tous les points.

Lorsque tous les points auront été calculés, il vous faudra les écrire dans le fichier MIDI dans un fichier en écriture «binaire» tel que décrit ci-après.

Votre programme doit définir un Main qui invoque votre méthode récursive de manière à produire une pièce musicale dont le niveau maximal est 5 et dont la distance sur l'axe horizontal (le temps) séparant A et B est 77 760 ticks MIDI.



Fichiers binaires

Pour être fonctionnels, les fichiers MIDI doivent être enregistrés sur disque dans le mode binaire et non pas dans le mode texte. Encodez donc les données de la piste MIDI dans un tableau avant de les enregistrer sur disque. Pour cela utilisez les fonction `pack_into` du module `struct` avec (par exemple) le format `'T'`. L'accès au fichier devrait se faire dans le 'mode binaire'. L'usage de ce mode signale à python de désactiver toute interprétation des octets lus et écrits au fichier; comme par exemple l'insertion de caractères de fin de ligne.

Créez le fichier MIDI avec la fonction `open` et le mode `'w+b'`. Cela assure que le fichier est créé s'il n'existait pas, qu'il est réinitialisé autrement, qu'on pourra y écrire et y lire et que python n'interprètera pas les octets en transit.

On écrit un tampon d'octets avec `write(tampon)`.

Les petits caractères

Ce travail se fait seul ou en équipe de deux.

Avertissement au sujet du plagiat

Tout travail montrant des évidences de plagiat se méritera la note de zéro; les individus dont les travaux sont copiés étant responsables au même titre que les individus qui dupliquent le travail d'autrui. Vous pouvez utiliser des informations obtenues sur Internet (vous ne pouvez pas dupliquer du code obtenu sur Internet), dans un livre ou dans une autre source : citez toutes vos sources! Soyez sérieux, faites votre travail, consultez les démonstrateurs. Si vous vous trouvez dans une discussion avec des collègues et que vous croyez que vous avez échangé suffisamment d'idées pour que vos solutions et analyses se ressemblent, dites-le franchement dans le rapport.

Fichiers à remettre sur Studium

Créez un répertoire dont le nom est matricule-ift2015H16-TP1 (remplacez 'matricule' par votre matricule bien sûr) avec la structure donnée ci-bas. Ensuite, utilisez l'utilitaire zip (ou la commande compress sur un mac) pour créer un fichier d'archive avec l'extension .zip (évitez les .rar et autres variations sur le thème svp). La remise consiste en un seul fichier, l'archive .zip résultante. Structure du répertoire à remettre :

matricule-ift2015H16-TP1/

équipiers.txt : ce fichier texte nomme les membres de l'équipe.

explications.txt : (fichier optionnel) Si vous désirez transmettre de l'information au correcteur, vous pouvez vous expliquer dans ce fichier texte.

code/ : ce répertoire comprend vos fichiers python.

Fonctionnalité

Un code est fonctionnel s'il remplit les spécifications et qu'il ne fait pas autre chose. Un programme qui ne répond pas aux tâches demandées n'est pas fonctionnel bien sûr. Mais un code qui enjolive sa sortie ne l'est pas non plus ! Par exemple : si un programmeur décide d'augmenter son interface d'entrées/sorties il devient

impossible d'utiliser son travail dans une chaîne d'outils prédéterminée et cela réduit la fonctionnalité de ses programmes. Lorsque vous produisez du code devant remplir des spécifications assurez-vous de remplir exactement les fonctionnalités.

Robustesse

Évidemment plus un programme comprends d'erreurs, moins il est robuste.

De plus, un programme d'ordinateur doit se comporter de manière consistante malgré les variations dans les données qui lui sont fournies en entrée et malgré les conditions variables d'exécution. En particulier lorsqu'une situation incongrue est détectée pendant l'exécution, le programme doit refuser de générer des résultats erronés et aviser l'utilisateur qu'une situation imprévue s'est produite. La politique « *crash early, crash hard* » est tout à fait bienvenue dans la plupart des cas (sauf évidemment si vous écrivez du code pour piloter un avion.)

Beauté

Un programme d'ordinateur n'est pas qu'une suite d'instructions destinées à être exécutée par un ordinateur ! C'est un objet abstrait destiné à être modifié par des humains. Un programme doit pouvoir être :

corrigé de ses erreurs (même les programmeurs très expérimentés font des erreurs de codage),

modifié (le comportement d'un programme devrait pouvoir être ajusté pour de nombreuses raisons n'ayant rien à voir avec le programmeur),

validé (la relecture critique de code reste une méthode très efficace d'assurance qualité), etc.

Ainsi, un programmeur utile écrit son code avec deux cibles en tête : la machine et l'humain. Parmi les alternatives, on doit favoriser l'usage d'idiomes communs et faciles à comprendre (par un programmeur compétent) et constamment s'efforcer à produire du code limpide. La marque d'un bon programmeur est la beauté de son code.

Correction négative

Le code non fonctionnel n'est pas recevable et la correction sera très sévère en ce qui concerne les remises partiellement fonctionnelles. La remise de TP ne réalisant pas les spécifications pourrait vous coûter jusqu'à la totalité des points.

Vous perdrez des points si vous utilisez un utilitaire produisant une archive qui n'est pas au format demandé ou si la structure de votre répertoire de remise n'est pas conforme aux demandes.

Correction positive

fonctionnalité: 3.33 pts, robustesse : 3.33, pts beauté : 3.33 pts.

Retard

20% par jour (tranches de 24h)

Soupçon de plagiat

zéro, 0, nil, zip, quedalle.

Date de remise

Mardi le 9 février 23h55.

Bon travail et bonne chance