

HOTZ-GARBER, ARIEL

**SYSTÈMES DE LINDENMAYER EN C++
AVEC ALGORITHMES GÉNÉTIQUES**

Travail présenté à
Jean-Cristophe Demers
Dans le cadre du cours
420.C61, Projet synthèse

Techniques de l'informatique
Cégep du Vieux Montréal
Mardi 5 octobre 2021

Présentation générale

L'objectif principal du présent projet consiste en la réalisation d'une application permettant d'afficher des fractales de formes et de types variables, selon certaines contraintes choisies par un utilisateur via une interface graphique.

Présentation détaillée

Fonctionnalités

Un L-système ou système de Lindenmayer est un système de réécriture consistant en un alphabet de symboles permettant de constituer des chaînes de caractères. À un ou plusieurs caractères de départ (appelé « axiome ») est appliquée une ou plusieurs règles de transformation successives. Cette chaîne est ensuite interprétée graphiquement, chaque symbole correspondant à une transformation particulière (par exemple une ligne droite, un angle à quarante-cinq degrés). Ce système formel a été proposé afin de décrire le comportement de cellules végétales et le la croissance développementale des plantes.

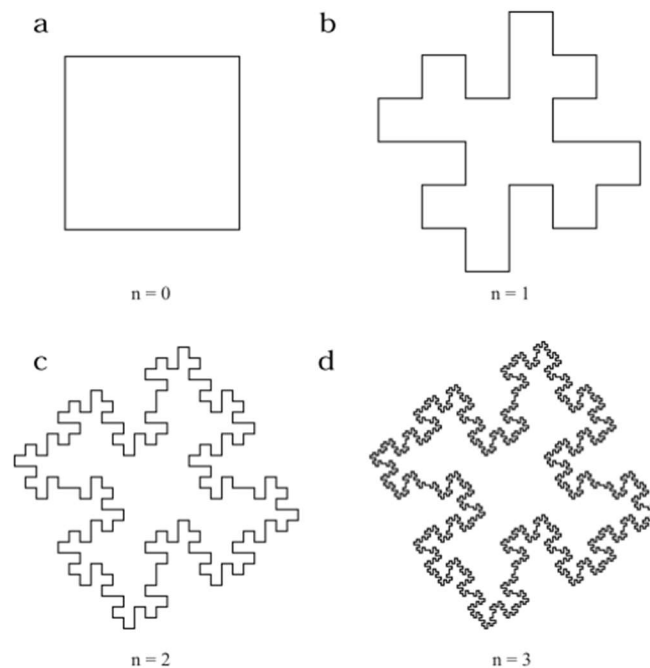


Figure 1. Génération d'une île de Koch à l'aide de l'axiome F-F-F-F et de la règle de transformation $F \rightarrow F-F+F+FF-F-F+F$. La règle est appliquée récursivement chaque itération. L'angle δ équivaut à 90 degrés. Le symbole F représente un segment de

droite. Les symboles plus et moins représentent une rotation d'angle *delta* à gauche et à droite respectivement¹.

Un mot composé de cet alphabet est interprété par une tortue graphique, chaque symbole modifiant la direction de la tortue. La position de la tortue est représentée par les coordonnées cartésiennes (x, y) et un angle *alpha* déterminant sa direction.

L'alphabet de base du logiciel comprendra les caractères « F, +, - » et le paramètre *delta* qui seront interprétés des manières suivantes : « F » représentera un segment de droite (dont la longueur ne pourra être modifiée par l'utilisateur, mais sera dépendante du nombre d'itérations), « + » et « - » modifieront l'angle *alpha* (c'est-à-dire l'angle représentant la direction à laquelle la tortue fait face) vers la gauche et vers la droite respectivement (cet angle sera déterminé à l'aide du paramètre *delta*). Par exemple, si l'angle *delta* équivaut à 33 degrés, la nouvelle position de la tortue sera (x, y, *alpha* + 45).

Il sera également possible de faire varier cet angle *delta* à l'aide des symboles « (» et «) » qui représenteront respectivement une incrémentation et une décrémentation. À cela se rajoutera le symbole « f » qui permettra de dessiner une courbe paramétrable.

Un symbole « * » dessinera un polygone régulier du choix de l'utilisateur. Le caractère « & » ignorera les *x* prochains caractères ou *x* sera compris entre 1 et 9, et déterminé au hasard. Finalement « { » et « } » augmenteront et diminueront une composante RGB. Le tableau suivant présente les différents symboles de l'alphabet et leur signification.

Symbole	Signification	Paramétrable
F	Dessine un segment de droite	Non
f	Dessine une courbe	Oui
+	Modifie la direction de la tortue vers la gauche	Oui
-	Modifie la direction de la tortue vers la droite	Oui
*	Dessine un polygone régulier	Oui (choix du polygone)
(et)	Modifient angle <i>delta</i>	Oui

&	Ignore les x prochains caractères	Non
{ et }	Modifient une composante RGB	Oui (composante)

Le logiciel offrira la possibilité de choisir plusieurs paramètres qui permettront à l'utilisateur de déterminer la forme de la fractale. L'utilisateur pourra ainsi choisir le nombre d'itérations c'est-à-dire le nombre de fois que la règle de transformation sera appliquée, qui affecteront la complexité – entre autres- de la forme. Il pourra déterminer la règle de départ (l'axiome), soit à partir d'une liste préexistante provenant d'une base de données, soit en générant son propre axiome qu'il entrera au clavier. Il sera également possible de choisir la ou les règles de transformation de transformation (pour un minimum d'une règle et un maximum de deux) à l'aide de champs.

Un bouton de départ permettra de démarrer le dessin à partir de l'axiome et de la règle. Un bouton de pause sera également présent. Il sera donc possible d'interrompre le dessin à tout moment. Lors d'une pause, l'utilisateur pourra redémarrer le programme ou choisir de faire défiler les itérations successives à l'aide d'un bouton « suivant ».

Les systèmes ne seront pas exclusivement déterministes. Ainsi, l'utilisateur pourra choisir jusqu'à deux règles de transformations dont une au minimum. À chacune de ses règles s'appliquera une probabilité (pour un total de 1) que l'utilisateur pourra également faire varier. Cette probabilité représentera la probabilité qu'aura la règle correspondante d'être appliquée chaque itération, l'autre n'étant pas appliquée.

Il serait également intéressant d'ajouter un tropisme c'est-à-dire une croissance orientée en fonction d'une variable g qui modifierait l'angle δ vers l'axe des y négatifs afin de simuler l'effet de la gravité. Cette fonctionnalité pourrait également tenir compte du poids cumulatif de la fractale ce qui accentuerait l'angle de croissance².

Les dessins de base seront en noir sur fond blanc, mais il sera possible de modifier à la fois la couleur du fond et la couleur de la forme en entrant une couleur en hexadécimal dans un champ prévu à cette fin. Comme susmentionné, les symboles parenthèse modifieront une composante RGB (la valeur de cette augmentation ou diminution sera fixe, mais l'utilisateur pourra choisir la composante RGB à modifier lors de la simulation).

Lorsque le dessin d'une forme se complétera, le logiciel permettra à l'utilisateur de sauver le mot (c'est-à-dire l'ensemble des symboles interprétés correspondant à la forme affichée) ou non. Si l'utilisateur confirme la sauvegarde, le mot sera conservé dans une base de donnée. L'utilisateur aura par la suite accès à ses formes sauvegardées et pourra ainsi les redessiner à sa guise. Un certain nombre de formes préenregistrées seront disponibles au lancement du logiciel. Lorsqu'un utilisateur sélectionnera une de ces formes, les champs correspondant aux paramètres de cette image seront remplis. L'utilisateur pourra ainsi lancer le dessin pour afficher cette image préenregistrée.

Une autre section de l'interface graphique sera réservée (et donc clairement séparée de la section précédente) aux paramètres de l'algorithme génétique. Celui-ci favorisera certaines formes selon une *fitness* paramétrable en interprétant l'image produite.

La *fitness* est le « trait » qui est favorisé chaque génération et qui détermine la chance qu'à un géniteur de transmettre son génotype à la génération suivante. Cette *fitness* sera évaluée soit en fonction de la symétrie de l'image soit en fonction de la plus grande aire totale soit en favorisant les formes ayant une forme plus « organique », c'est-à-dire ceux comportant plus de courbes, les plus symétriques et aux angles moins aigus.

La population de départ représente le bassin de gènes de départ qui sera modifié (par mutation et par croisement) chaque génération successive. Cette population de départ sera composée de différents mots (c'est-à-dire différentes chaînes composées de caractères de l'alphabet de départ. Suite au dessin d'une forme, l'utilisateur pourra choisir d'utiliser celle-ci pour générer une population de départ et lancer l'algorithme génétique. L'utilisateur pourra aussi faire varier la taille de cette population de départ.

Le croisement s'effectuera en déterminant un pivot qui scindera le chromosome (c'est-à-dire la chaîne de caractères) des deux géniteurs. Leur progéniture sera constituée d'une section de chacun de ses parents. Le croisement s'opérera entre deux parents. Ils seront choisis en fonction d'un poids relatifs qui leur sera accordé en fonction de leur *fitness*.

Cette nouvelle génération sera ensuite exposée à une mutation, déterminée par un degré de probabilité déterminé par l'utilisateur. Si cette mutation a lieu, chaque gène pourra être substitué par un gène différent généré au hasard.

Le nombre de générations, le pourcentage d'élitisme et de mutation seront également modifiables à la guise de l'utilisateur.

Contraintes applicatives

Étant donné que l'utilisateur pourra choisir à la fois le nombre d'itérations pour les systèmes Lindenmayer et le nombre de générations pour les algorithmes génétiques, les nombre d'opérations pourraient facilement devenir trop important ce qui rendrait les délais trop importants. Il conviendrait donc d'instaurer une minuterie qui couperait l'exécution de l'algorithme génétique ou de la génération de systèmes.

Il sera important de rendre le logiciel modulaire afin de pouvoir rajouter des fonctionnalités dans le futur, par exemple rajouter des caractères dans l'alphabet, ou rajouter des paramètres à l'algorithme génétique ou même introduire des nouveaux symboles permettant de créer des effets de branchement qui simuleraient mieux des formes s'apparentant à des plantes ou des arbres.

Plateforme ciblée

Le logiciel sera conçu pour être lancé et exécuté sur un ordinateur.

¹ Tiré de *The Algorithmic Beauty of Plants*

² Cette fonctionnalité simulant un effet de gravité est inspirée d'une fonctionnalité similaire retrouvée au Tiré de <https://www.sidefx.com/docs/houdini/nodes/sop/lssystem.html>