# GENERATION PROCEDURALE DE PLANTES

## HMIN317 – Moteur de jeux


### Résumé

Rapport de l'exposé sur la génération procédurale de plantes.



Maxime Demaille - David Lonni

Master 2 IMAGINA

# Table des matières

Introduction....2
1) Fractale....2
2) Iterated Function System (IFS)....3
3) L-System....5
4) Application....7
5) LOD....8
5.1) Statique....9
5.2) Continu....9
5.3) Simplification à niveaux de détails continus....9
6) Billboard....10
Conclusion....12
Table des figures....12
Bibliographie....12

# Introduction

La génération procédurale désigne le fait d'utiliser des méthodes mathématiques pour créer des objets ou des environnements de manière dynamique, autrement dit créer des données au moyen d'algorithmes.

Dans ce document, nous allons nous intéresser à la génération procédurale de végétaux et présenter les algorithmes qui sont utilisés, ainsi que les domaines d'application de cette génération.

## 1) Fractale

Une fractale est un objet géométrique «infiniment morcelé» dont des détails sont observables à une échelle arbitrairement choisie.

Le terme « fractale » est un néologisme créé par Benoît Mandelbrot en 1974. En zoomant sur une partie de la figure, on peut retrouver toute la figure, on dit qu'elle est auto similaire.

Une fractale désigne des objets dont la structure est invariante par changement d'échelle. Certains végétaux comme la fougère ou le chou possèdent de splendides fractales.

*Figure 1: Image de Jon Sullivan - Chou Romanesco.*

*Figure 2 : Image de Kimbar – Fougère*

## 2) Iterated Function System (IFS)

C'est une théorie mathématique développée par John Hutchinson en 1981, utilisée dans le cadre de la géométrie fractale. Cette théorie est entièrement fondée sur les invariances par changement d'échelle.

Barnsley a démontré, avec le Théorème du collage, que tout ensemble de points peut être approximé par un IFS.

En termes simples, le théorème du collage prouve qu'on peut recouvrir toute forme de l'espace par des copies d'elle-même.

**Remarque :**

- La plupart des fonctions des IFS sont des fonctions affines. On appelle flame IFS des fractales obtenues par des fonctions non linéaires.

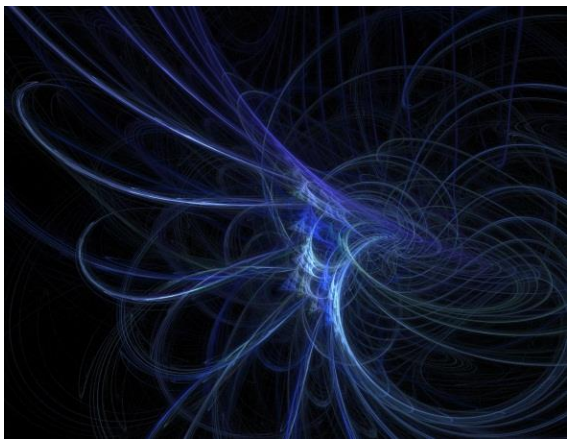

*Figure 3 : Image de Jonathan Zander - Flame IFS.*

**Exemples :**

L'ensemble de Cantor :

Ensemble de fonctions définissant des transformations à appliquer sur les objets par rapport à un point que l'on répète X fois pour obtenir une fractale.

*Figure 4 : Image de Sarang - Ensemble de Cantor en 7 itérations.*

Le tapis de Sierpinski (1916), est défini par 8 similitudes de rapports 1/3. C'est une fractale obtenue à partir d'un carré.

Le tapis se fabrique en découpant le carré en neuf carrés égaux avec une grille de trois par trois, et en supprimant la pièce centrale, et en appliquant cette procédure indéfiniment aux huit carrés restants.

Si on prolonge la fractale à l'infini, la surface du carré est intégralement « vidée ».

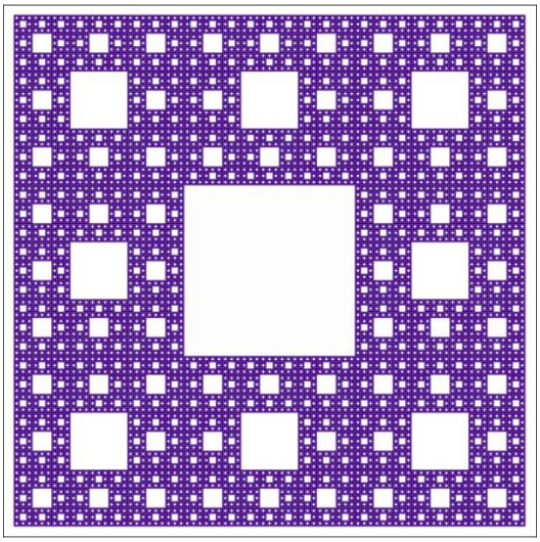

*Figure 5 : Image de KarocksOrkav - Tapis de Sierpiński animé.*

Le triangle de Sierpinski peut s'obtenir à partir d'un triangle « plein », par une infinité de répétitions consistant à diviser par deux la taille du triangle puis à les accoler en trois exemplaires par leurs sommets pour former un nouveau triangle. À chaque répétition le triangle est donc de même taille, mais « de moins en moins plein ».

*Figure 6 : Image de Wereon - Evolution du triangle de Sierpinski en 5 iterations.*

La fougère de Barnsley est ici construite à partir de quatre contractions affines (rouge, bleu, cyan et vert sur l'illustration).

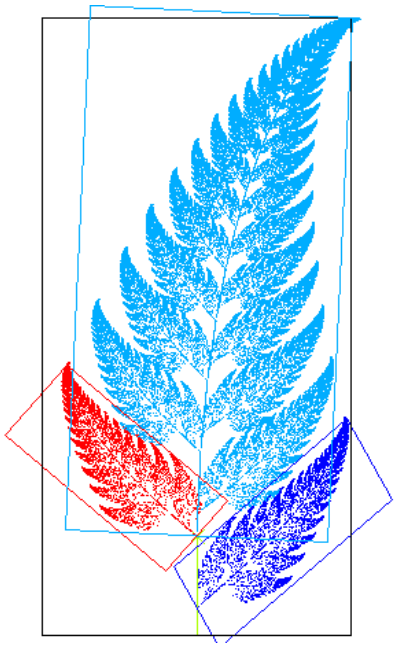

*Figure 7 : Image d'António Miguel de Campos - La fougère de Barnsley, élaborée par un système de quatre fonctions affines.*

## 3) L-System

Un L-Système ou système de Lindenmayer est une grammaire formelle, inventée en 1968 par le biologiste hongrois Aristid Lindenmayer, qui consiste à modéliser le processus de développement et de prolifération de plantes ou de bactéries.

Un L-système est un ensemble de règles et de symboles qui modélisent un processus de croissance d'êtres vivants comme des plantes ou des cellules. Le concept central des L-systèmes est la notion de réécriture. La réécriture est une méthode utilisée pour construire des objets complexes en remplaçant des parties d'un objet initial simple, ceci en utilisant des règles de réécriture.

Par exemple, on considère l'alphabet ( a,d,g ), le mot initial u(0) = a et les règles de dérivation suivantes :

- a devient agaddaga
- g devient g
- d devient d

On obtient alors :

u(o) = a
u(1) = agaddaga
u(2) = agaddagagagaddagaddagaddagagagaddaga

On peut appliquer ce raisonnement sur des figures géométriques. Nous allons détailler un exemple :

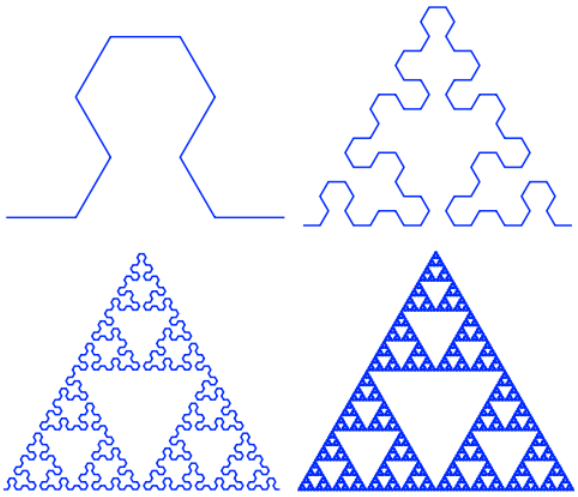

*Figure 8 : Image d'António Miguel de Campos - Triangles de Sierpinski  sur n itérations pour n =2, n = 4, n = 6 et n = 8.*

Par exemple, le triangle de Sierpinski est représenté par la grammaire suivante :

- Alphabet : A,B,+,-
- Départ : A
- Règles : A devient +B-A-B+, B devient -A+B+A-
- angle : 60°

A et B signifient “dessiner vers l’avant”, + signifie “Tourner à gauche selon l’angle” et – signifie “tourner à droite selon l’angle”.
Une figure peut correspondre à plusieurs grammaires.

## 4) Application

La génération procédurale est utilisée dans les jeux vidéo, dans les effets spéciaux et dans l'animation. En effet, il est nécessaire, dans ces domaines, d'afficher une grande variété de végétation, et la génération procédurale permet d'automatiser ce processus.

L'outil SpeedTree permet de générer procéduralement et d'éditer des plantes. Cet outil est notamment utilisé pour générer les plantes dans les jeux The Witcher 3, Shadow of Mordor, Far Cry 4, mais aussi dans les films Avatar, Star Trek : Into Darkness ou encore Pirates of the Caribbean : On Stranger Tides.


*Figure 9 : Image du jeu The Witcher 3.*


*Figure 10 : Image du film Star Trek : Into Darkness.*

## 5) LOD

Le Level Of Detail, généralement abrégé LOD est une technique utilisée dans la modélisation 3D temps réel (principalement dans le jeu vidéo), qui définit un niveau de détail d'un objet, parmi plusieurs prédéfinis, suivant la taille qu'il aura à l'écran.

Aujourd'hui, la modélisation 3D en temps réel possède un potentiel technique important limité par les contraintes de calculs des processeurs graphiques.

Le nombre de calculs nécessaires augmente avec le nombre de faces et les effets de texture affectés à ces faces.

Lorsqu'un objet ne fait plus que quelques pixels à l'écran, lorsque l'objet rétrécit, soit par éloignement, soit pour autre raison, l'utilisateur ne pourra plus distinguer l'ensemble des détails.

Une version simplifiée du modèle permet de conserver un aspect similaire tout en réduisant la somme des calculs nécessaires à sa représentation.

Il existe 2 types de LOD par simplification de maillage :

- Statique

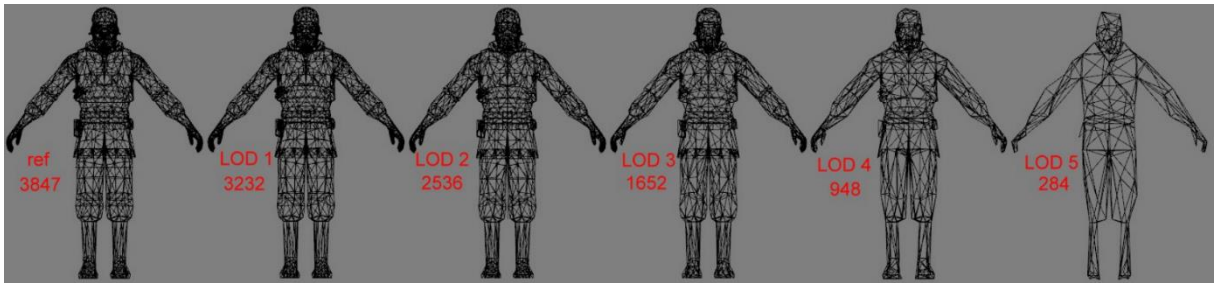


*Figure 11 : Image de LOD statique.*

- Continu

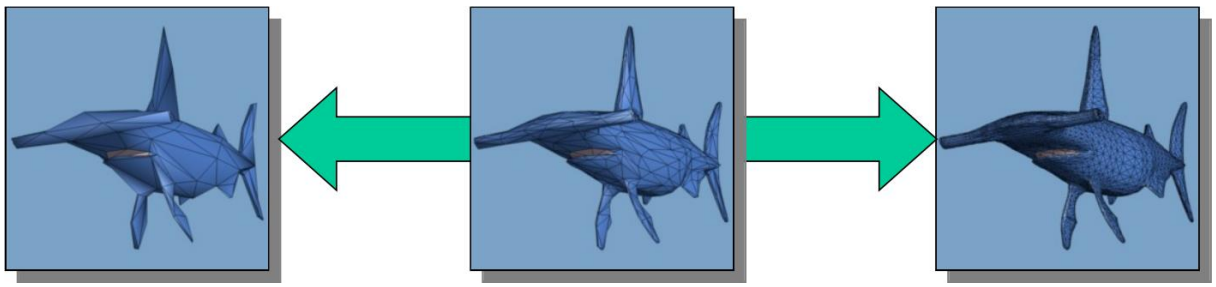

*Figure 12 : Image de LOD continu.*

### 5.1) Statique

Le LOD statique consiste à définir plusieurs versions du modèle avec différents niveaux de détails, et donc avec plus ou moins de polygones, afin de pouvoir afficher le bon modèle en fonction de l'éloignement de la caméra.

L'avantage de cette méthode est qu'elle est très rapide à mettre en œuvre. Cependant elle a pour inconvénients quelques problèmes de transitions entre les modèles ce qui a pour effet de faire des "effets de saut". En effet, le passage brutal d'un modèle détaillé à un modèle plus simplifié peut se voir à l'écran et engendrer cet effet. Le second inconvénient de cette méthode est qu'il est nécessaire de stocker en mémoire plusieurs objets d'un même modèle à différentes résolutions.

### 5.2) Continu

Le LOD continu consiste à décimer un maillage à la volée pendant l'exécution du programme.

Cette méthode à l'avantage d'éviter le problème de transitions cité précédemment et ainsi empêcher "l'effet de saut". Grâce à cela on obtient une décimation dynamique de l'objet et on génère donc autant de modèle de l'objet que nécessaire en fonction de la distance avec la caméra.

Il permet donc également de contrer le second inconvénient du LOD statique à savoir le problème de mémoire. Ici, il n'est pas nécessaire de stocker en mémoire les différents maillages car ils sont générés au vol directement. Le seul inconvénient de cette méthode est le temps de calcul nécessaire à la simplification dynamique de ces maillages.

### 5.3) Simplification à niveaux de détails continus

Ici, nous nous intéresserons à la simplification à niveaux de détails continus (CLOD, « Continuous Level Of Detail ») qui nous permettra donc à la volée de changer la résolution de nos plantes générées procéduralement.

Il existe de nombreux algorithmes permettant cette simplification à la volée, comme l'algorithme de Catmull-Clark, les schémas de subdivision, l'algorithme de progressive meshes.

Lorsqu'on arrive à un niveau de simplification trop important, on peut remplacer notre modèle 3D par un imposteur ("billboard") pour alléger encore le temps de calcul.

## 6) Billboard

Un objet 3D peut être simplifié en un ensemble de polygones sur lesquels sont plaqués des textures semi-transparentes, les billboards.

Dans les jeux vidéo, certains billboards sont toujours orientés face à la caméra dans le but de donner l'illusion d'un modèle 3D complexe.

C'est le cas par exemple le plus souvent de l'herbe sur les terrains. Les billboards sont toujours orientés vers la caméra afin que l'on ne remarque pas que c'est uniquement un plan texturé en tournant autour de celui-ci.

*Figure 13 : Exemple de billboard toujours orienté face à la caméra.*

Mais les billboards peuvent également être fixe comme on peut le voir pour les arbres qui sont parfois générés avec deux billboards perpendiculaires afin de donner l'impression d'un arbre 3D.

*Figure 14 : Exemple d'arbre affiché avec deux billboards orientés perpendiculairement.*

Le feuillage des arbres peut être représenté par un nuage de billboards qui, là encore, ne sera pas orienté face à la caméra.

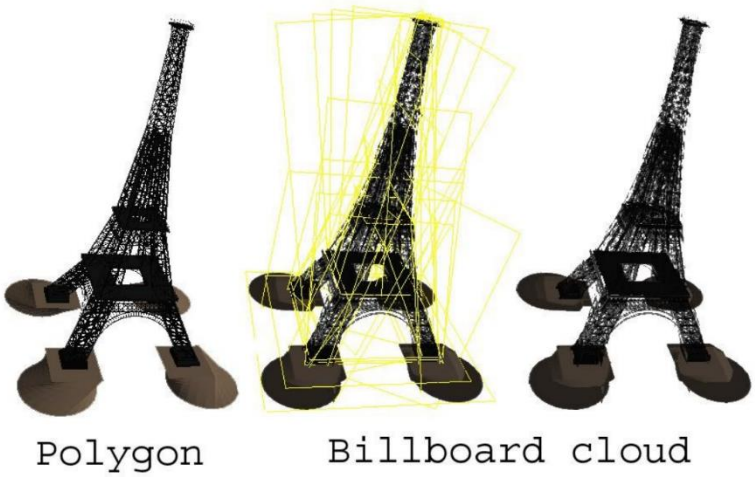


*Figure 15 : Image de nuage de billboard.*

Comme on peut le voir avec l'exemple ci-dessus, un objet 3D complexe peut être imité avec un ensemble simple de billboards. C'est le cas avec les feuillages des arbres générés de manière procédurale.

Exemple :

Ces arbres sont représentés avec plus ou moins de facettes et de quadrilatères texturés en fonction de la distance au point de vue.

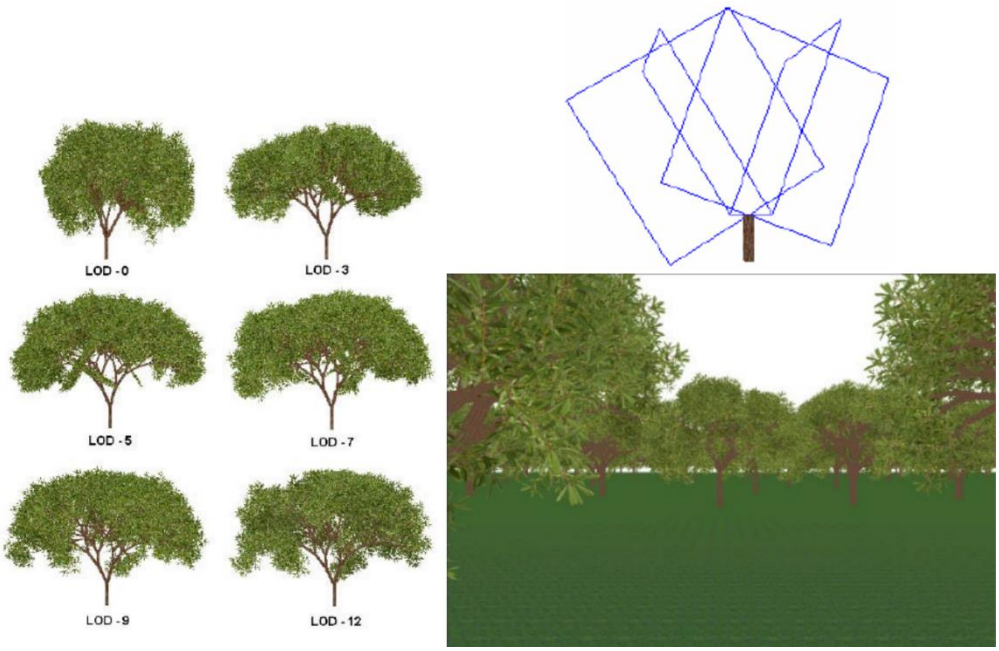


*Figure 16 : Exemple de LOD avec des billboards.*

# Conclusion

La génération procédurale de végétation est de plus en plus utilisée dans les jeux vidéo, d'autant qu'avec la puissance des ordinateurs modernes, il est désormais possible de créer de vastes environnements. Nous avons vu qu'il existait plusieurs méthodes permettant la génération procédurale et que celles-ci étaient largement utilisées dans les grandes productions. En effet, ces méthodes permettent de générer des environnements réalistes, et les techniques de LOD et de subdivision associées permettent l'utilisation de la génération procédurale dans les jeux vidéo, où il est constamment nécessaire de réaliser des compromis entre graphismes et performances.

# Table des figures

Figure 1: Image de Jon Sullivan - Chou Romanesco. ..... 2
Figure 2 : Image de Kimbar – Fougère ..... 2
Figure 3 : Image de Jonathan Zander - Flame IFS. ..... 3
Figure 4 : Image de Sarang - Ensemble de Cantor en 7 itérations. ..... 3
Figure 5 : Image de KarocksOrkav - Tapis de Sierpiński animé. ..... 4
Figure 6 : Image de Wereon - Evolution du triangle de Sierpinski en 5 iterations. ..... 4
Figure 7 : Image d'António Miguel de Campos - La fougère de Barnsley, élaborée par un système de quatre fonctions affines. ..... 5
Figure 8 : Image d'António Miguel de Campos - Triangles de Sierpinski sur n itérations pour n =2, n = 4, n = 6 et n = 8. ..... 6
Figure 9 : Image du jeu The Witcher 3. ..... 7
Figure 10 : Image du film Star Trek : Into Darkness. ..... 7
Figure 11 : Image de LOD statique. ..... 8
Figure 12 : Image de LOD continu. ..... 8
Figure 13 : Exemple de billboard toujours orienté face à la caméra. ..... 10
Figure 14 : Exemple d'arbre affiché avec deux billboards orientés perpendiculairement. ..... 10
Figure 15 : Image de nuage de billboard. ..... 11
Figure 16 : Exemple de LOD avec des billboards. ..... 11

# Bibliographie

Fractales :

- https://commons.wikimedia.org/wiki/File:Fractal_Broccoli.jpg#/media/File:Romanesco_broccoli_fractals.jpg
- https://commons.wikimedia.org/wiki/File:Bransleys_fern.png

IFS :

- https://commons.wikimedia.org/wiki/File:Star_Apophysis_Fractal_Flame.jpg#/media/File:Lines_Apophysis_Fractal_Flame.jpg
- https://commons.wikimedia.org/wiki/File:Animated_Sierpinski_carpet.gif

- https://fr.wikipedia.org/wiki/Triangle_de_Sierpi%C5%84ski#/media/File:Sierpinski_triangle_evolution.svg
- https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_fonctions_it%C3%A9r%C3%A9es#/media/File:Fractal_fern_explained.png

L-System :

- https://en.wikipedia.org/wiki/L-system#/media/File:Serpinski_Lsystem.svg