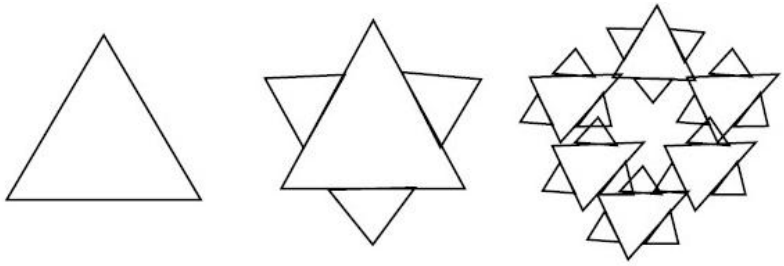
TP Génération de contenu - Géométrie
Modélisation procédurale

Modélisation procédurale :

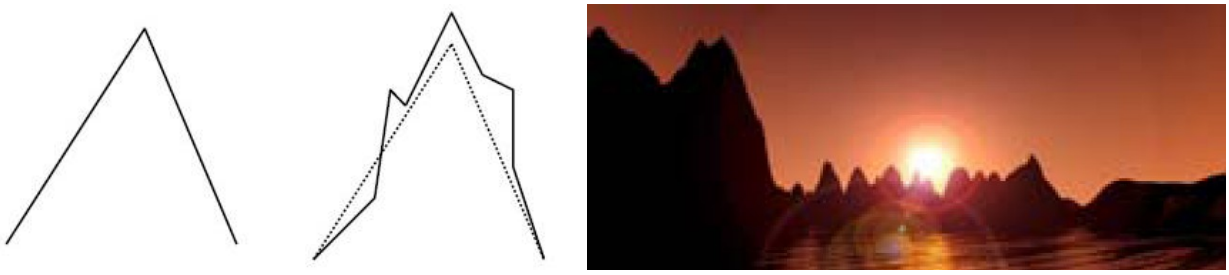
La modélisation procédurale permet de créer de la géométrie à la volée. En effet, les primitives géométriques sont créées par une procédure. Ce type de modélisation est surtout utilisé pour modéliser des objets complexes et répétitifs comme les plantes, paysages, villes ou tout objet pouvant être décrit par des règles de construction.

Fractales :

L'exemple le plus connu de modélisation procédurale est la modélisation par fractal, où les détails sont ajoutés récursivement.

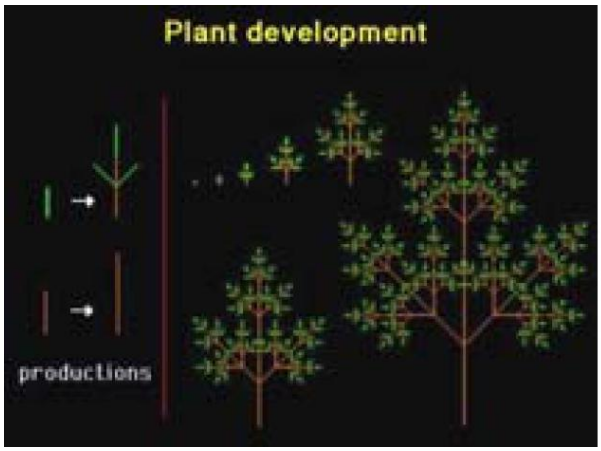


Les terrains peuvent être créés par modélisation fractale en déplaçant aléatoirement le terrain à chaque étape :

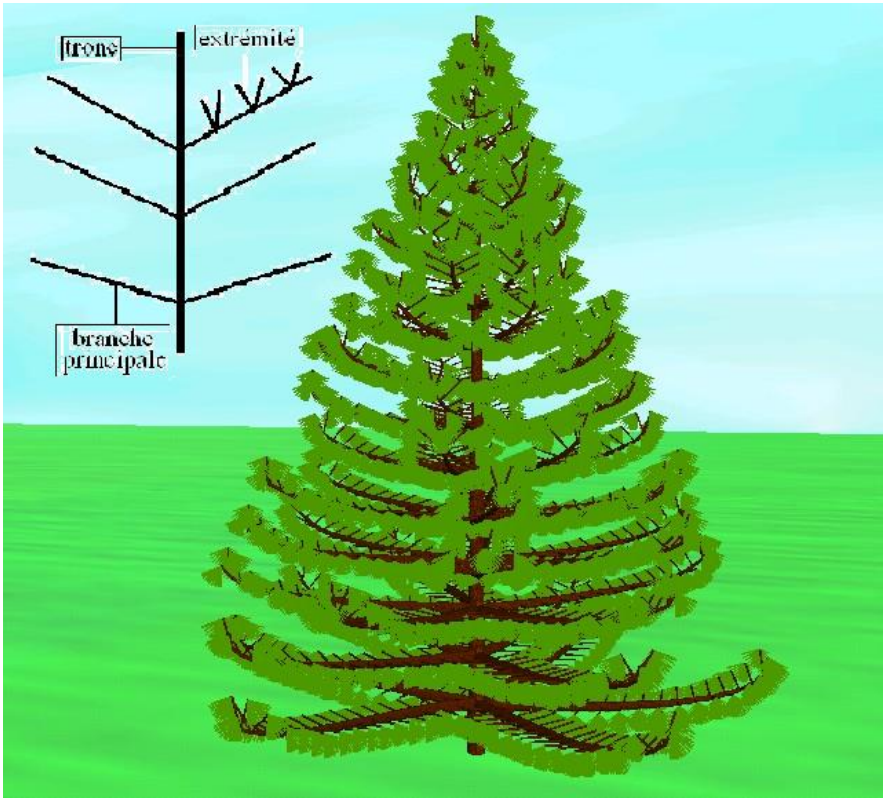


L-systèmes :

Les plantes peuvent en général être modélisées par des grammaires qui régissent leur croissance.



1. Lisez la page wikipedia sur les L-systèmes *(http://fr.wikipedia.org/wiki/L-System)*.
2. Nous allons utiliser un tel système pour modéliser un sapin.



Un tel arbre est composé de trois types d'objets :

- des morceaux de tronc,
- des morceaux de branches principales,
- des extrémités.

Chacun de ces objets a :

- un âge en fonction de sa date de croissance a,
- une longueur l,
- un angle dans le plan parallèle au sol appelé angle horizontal α_h,
- un angle dans le plan perpendiculaire au sol appelé angle vertical α_v.

A partir de ces définition, un sapin peut-être construit avec une grammaire simple :

(a) Tree(a, l) → T(a, l, 0, 90)

(b) T(a, l, α_h, α_v) → T(a, l, 0, 90) [P(a-1, l/2, rnd(0,360), rnd(0,30))]*

(c) P(a, l, α_h, α_v) → P(a-1, 0.98l, $\alpha_h \pm \epsilon$, rnd(0,30)) B(a-1, rnd(0.3,0.8), α_h+90, rnd(0,30)) B(a-1, rnd(0.3,0.8), α_h-90, rnd(0,30))

(d) B(a, l, α_h, α_v) → ∅

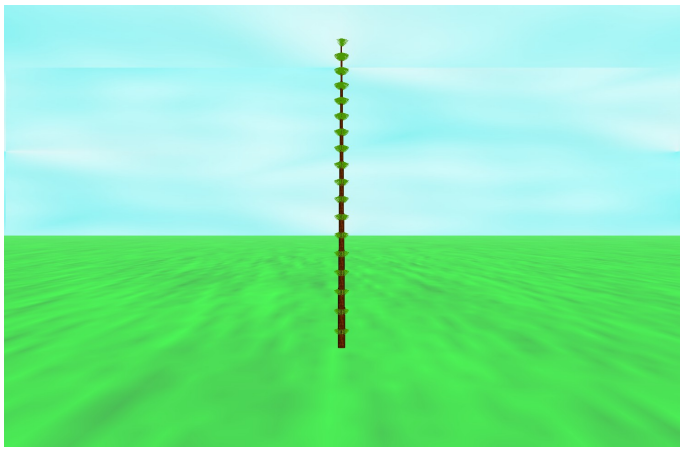
où

- rnd(a,b) renvoie un nombre aléatoire entre a et b,
- la notation [X]* signifie que l'élément X apparaît un nombre aléatoire de fois.

3. Récupérez le code fourni *(arbres.zip)* et compilez-le. Etudiez sa structure :

   - `arbres.cpp` s'occupe de la partie OpenGL (affichage ; contrôles : caméra avec Up, Down, Left, Right, PageUp, PageDown ; définition des objets). En particulier, dans `initScene()`, un abre d'âge 18 et de longueur 1 est créé : `arbre = new Tree(18.0,1.0);`.
   - Dans tree.h, deux types d'objets sont définis. La classe `Tree` définit un arbre décrit par un pointeur vers un objet de type `Branch`. La structure `Branch` définit les objets de base d'un arbre. En plus des attributs vus précédemment, la structure `Branch` définit grâce à deux bouléens (`trunc` et `ppal_branc`) le type de l'objet et un ensemble de branches accrochées à cet objet (`twigs`).
   - La classe `Tree` possède deux méthodes qui permettent de dériver les règles de la grammaire ci-dessus. La méthode `new_trunk()` crée un objet T et dérive la règle b et la méhode `new_branch()` crée un objet P ou B et dérive les règles c et d. La règle a est dérivée dans le créateur de l'objet.

4. Complétez la méthode `new_trunk()`. Vous devriez obtenir le tronc :



5. Complétez la methode `new_branch()`. Vous devriez obtenir un sapin complet.