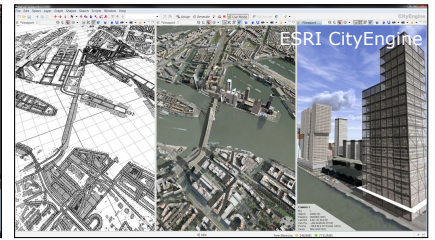


Génération procédurale



Grammaires, L-System, shape grammars

Nicolas Bredèche

Professeur des Universités (ISIR, SU)

nicolas.bredeche@upmc.fr

Module: 2i013

Dernière mise à jour: 2019-02-25

--

Objectif du cours

- Objectif du cours
 - les processus génératifs
 - développement et règles de réécriture
 - génération automatique de structures/formes
- Génération de formes
 - les grammaires génératives
 - les L-Systems
 - les grammaires de forme

Grammaires

Une brève introduction



Noam Chomsky (né en 1928)
grammaire générative et transformationnelle

Langage

- Définition
 - Un langage est un ensemble de phrases construit à partir d'un ensemble de mots
- Éléments:
 - Un lexique de “mots” (ou “symboles”)
 - Des phrases
 - des suites finies de mots

Langage: exemples

$$A = \{a, b\}$$

- Des phrases: $a, ababba, abbba, \dots$

- Quelques langages:

$$L_1 = \{aa, aba, aaa\}$$

$$L_2 = \{a^m b^n; m, n \in \mathbb{N}\}$$

$$L_3 = \{a^n b^n; n \in \mathbb{N}\}$$

Langage: exemples

$$A = \{a, b\}$$

- Des phrases: $a, ababba, abbba, \dots$

- Quelques langages:

$$L_1 = \{aa, aba, aaa\}$$

$$L_2 = \{a^m b^n; m, n \in \mathbb{N}\}$$

$$L_3 = \{a^n b^n; n \in \mathbb{N}\}$$

Un langage qui englobe L_1, L_2 et L_3

$$L_4 = \{a^n b^m a^l; m, n, l \in \mathbb{N}\}$$

extrait de <http://paulegre.free.fr/Teaching/gramm2.pdf>

Grammaire

- Définition
 - Une **grammaire** permet d'engendrer des **phrases** à partir d'un **lexique** (ensemble de mots) et de **règles de transformation**
- Éléments
 - un lexique propre A , de symboles terminaux
 - ▶ y compris « ϵ », mot vide
 - un lexique intermédiaire I
 - ▶ y compris « S », symbole de transition
 - un ensemble de règles R
- Un langage (définition plus précise)
 - Ensemble de toutes les phrases pouvant être générées à partir de S étant donnée A

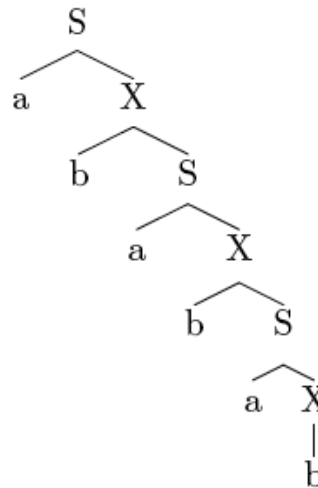
Equivalence de représentation

$$A = \{a, b\}, I = \{S, X\}$$

$S \Rightarrow aX$
$X \Rightarrow b$
$X \Rightarrow bS$

$$S \Rightarrow aX \Rightarrow abS \Rightarrow abaX \Rightarrow ababS \Rightarrow ababaX \Rightarrow ababab$$

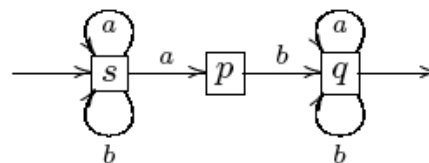
Représentation sous forme d'arbre:



extrait de <http://paulegre.free.fr/Teaching/gramm2.pdf>

Automates Finis et Grammaire

$S \Rightarrow aS$	$P \Rightarrow bQ$	$Q \Rightarrow aQ$
$S \Rightarrow bS$		$Q \Rightarrow bQ$
$S \Rightarrow aP$		$Q \Rightarrow e$



Reformulation en automates finis d'une grammaire régulière

la formulation sous forme d'automates finis permet de générer ou de reconnaître facilement des phrases

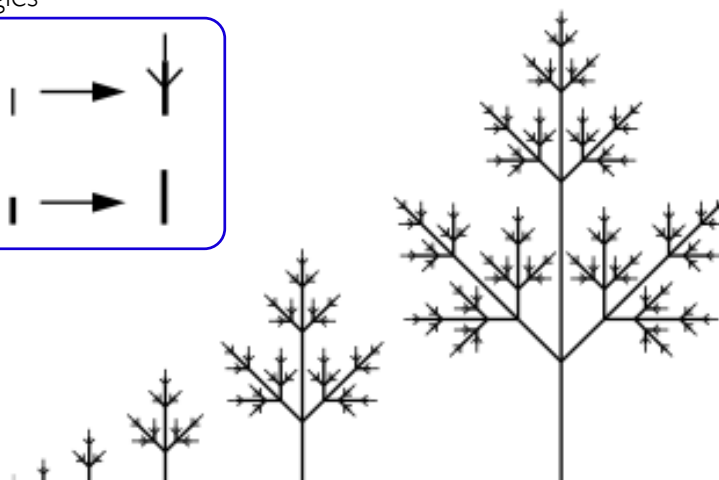
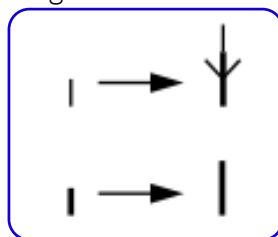
extrait de <http://paulegre.free.fr/Teaching/gramm2.pdf>



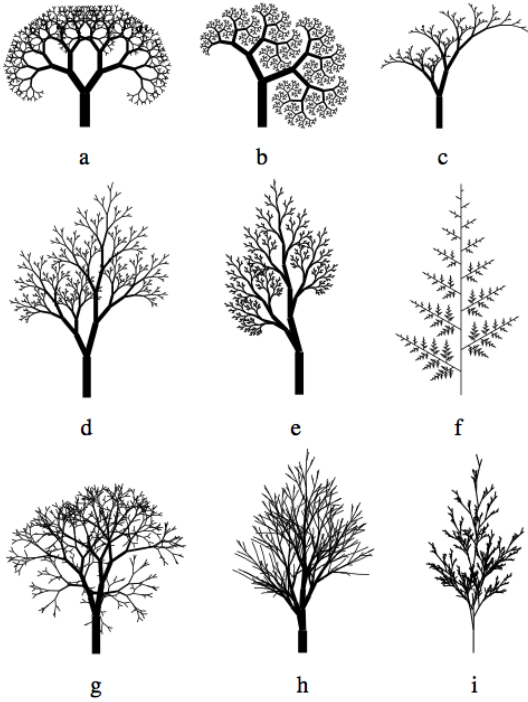
L-Systems

Système de Lindenmayer

règles



L-systems : langage et grammaire pour le développement d'une plante type



Prusinkiewicz et al. 1996, "L-Systems: from the theory to visual models of plants"

image (right): Wikipedia commons

Cas particulier : la règle identité

La règle de production "identité"
si aucune règle ne s'applique, alors recopie

Définition formelle

parametric-0L-System

G = (V, Σ, ω, P)

V Alphabet

Σ Ensemble de paramètres formels

ω Axiome condition initiale du système

P Règles de production

Format d'une règle de production:

prédécesseur : condition => successeur

Alphabet : $V = \{A, B\}$

Constantes : $S = \{\}$

Axiome de départ : $w = A$

Règles : $(A \rightarrow B) \wedge (B \rightarrow AB)$

n=0, A

n=1, B

n=2, AB

n=3, B AB

n=4, AB B AB

n=5, B AB AB B AB

n=6, AB B AB B AB AB B AB

Remarque: la progression du nombre de symboles à chaque itérations suit la suite de Fibonacci

Différentes classes de L-System

- Propriétés
 - Paramétrique ou non
 - Déterministe ou stochastique
 - Sensible au contexte ou non
- Nomenclature
 - exemple: D0L-System
 - ▶ Même forme qu'à peu près tout ce qu'on a vu jusqu'ici

Paramétrique... ou non (1/2)

- non-paramétrique
 - $G = (V, \omega, P)$
 - V Alphabet
 - ω Axiome condition initiale du système
 - P Règles de production
 - prédécesseur \Rightarrow successeur
 - exemple:
 - ▶ $\omega : A$
 - ▶ $P : A \Rightarrow AB$

La classe la plus simple:

D0L-System = non-paramétrique, déterministe, pas sensible au contexte

Paramétrique... ou non (2/2)

- paramétrique [“parametric-L-System”]:

- $G = (V, \Sigma, \omega, P)$

V	Alphabet
Σ	Ensemble de paramètres formels
ω	Axiome condition initiale du système
P	Règles de production

- prédécesseur : conditions \Rightarrow successeur

- exemple [Parametric-DOL-System]:

- ▶ $\omega : A(7)$

- ▶ $P : A(x) : x > 3 \Rightarrow A(x-1)B$

déterministe vs. stochastique

- déterministe [“DL-System”]

- une seule règle possible

- exemple:

- ▶ $\omega : A(7)$

- ▶ $P : A(x) : x > 3 \Rightarrow A(x-1)B$

- stochastique [“SL-System”]

- choix au hasard entre plusieurs règles possibles

- exemple:

- ▶ $\omega : A(7)$

- ▶ $P1 : A(x) : x > 3 \Rightarrow A(x-1)B$; $P2 : A(x) : x > 4 \Rightarrow A(x-1)A(x)$

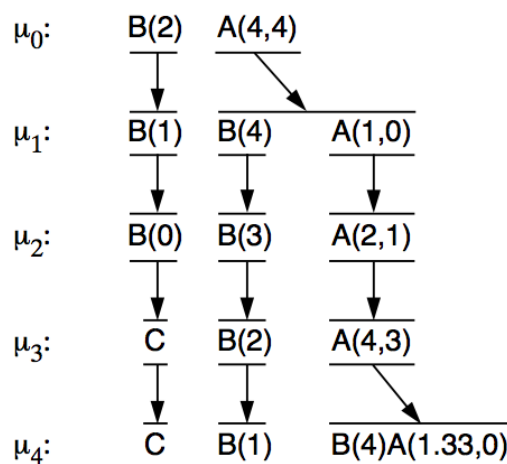
Sensibilité au contexte

- sensible au contexte [“1L-System” ou “2L-System”]
 - Définition formelle:
 - ▶ bloc-gauche<prédécesseur>bloc-droit {; conditions} => successeur
 - le prédécesseur n'est pas seul dans la partie gauche de la règle
 - exemples:
 - ▶ D2L-System :
 - $\omega : A$; $P : B<A>B \Rightarrow AB$
 - ▶ D1L-System :
 - $\omega : A$; $P : <A>C \Rightarrow B$ (ou encore: $P : B<A> \Rightarrow E$)

Remarque: on parle de 0L-System si il n'y a pas de sensibilité au contexte

“Parametric-D0L-System”

$$\begin{aligned}
 \omega &: B(2)A(4,4) \\
 p_1 &: A(x,y) : y < 3 \rightarrow A(x * 2, x + y) \\
 p_2 &: A(x,y) : y \geq 3 \rightarrow B(x)A(x/y, 0) \\
 p_3 &: B(x) : x < 1 \rightarrow C \\
 p_4 &: B(x) : x \geq 1 \rightarrow B(x - 1)
 \end{aligned}$$



“Parametric-D2L-System”

déterministe, s'il n'y a que cette règle

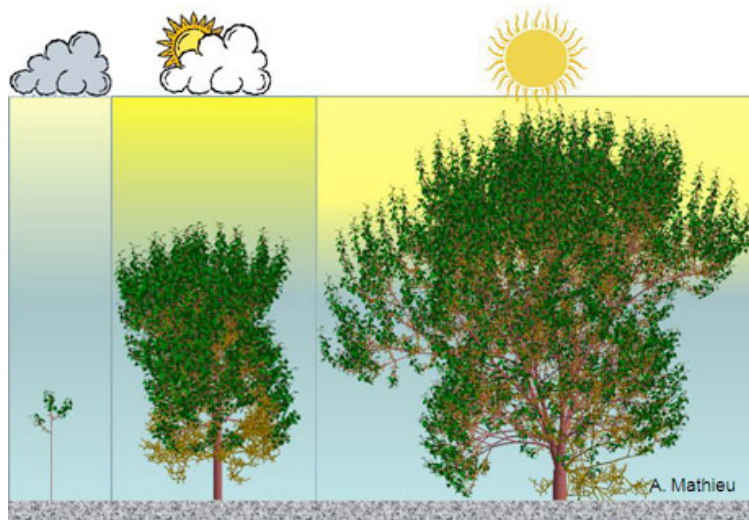
$\omega : \dots A(4)B(5)C(6) \dots$

$P : A(x) < B(y) > C(z) : x + y + z > 10 \rightarrow E((x + y)/2)F((y + z)/2)$

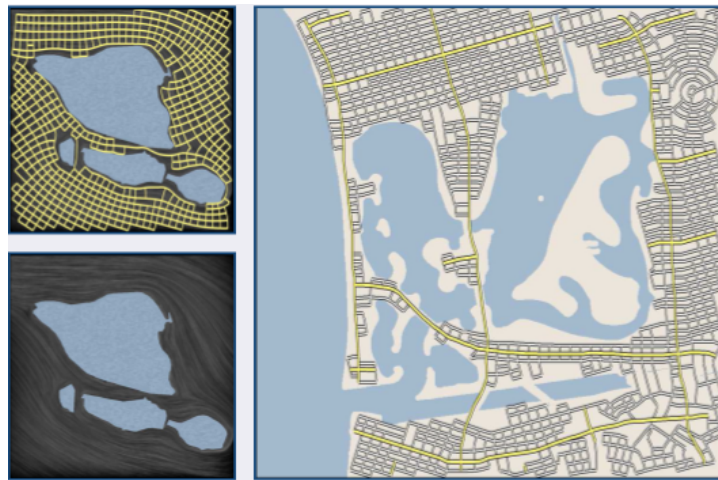
$\dots E(4.5)F(5.5) \dots$

source: Prusinkiewicz et al. L-Systems: from theory to visual models of plants. 1997

Exemple : prise en compte du contexte



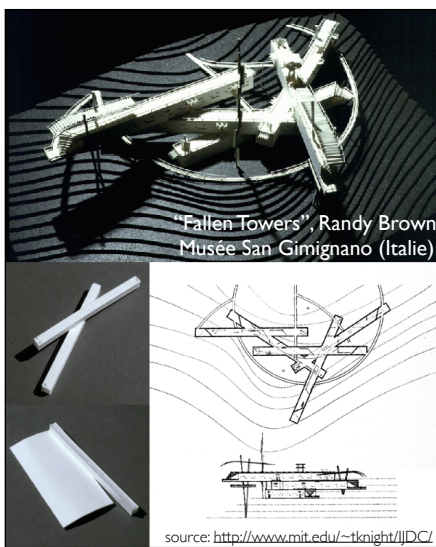
Simulation de l'influence de l'ensoleillement sur la croissance d'un arbre



Croissance de routes

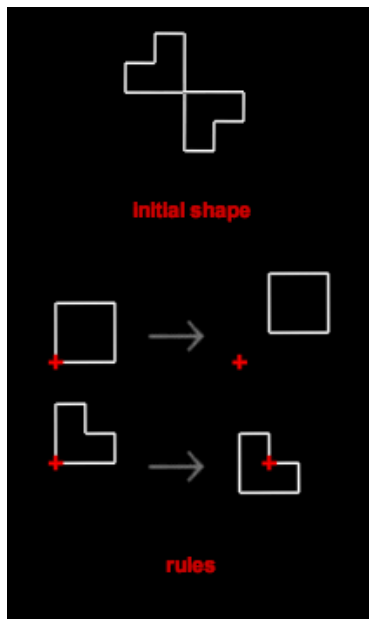
L-System adapté + génération de pattern + tenseurs
 [Parish et al., SIGGRAPH 2001][Esch et al., SIGGRAPH 2008]

source: présentation Eric Galin, GDC 2008



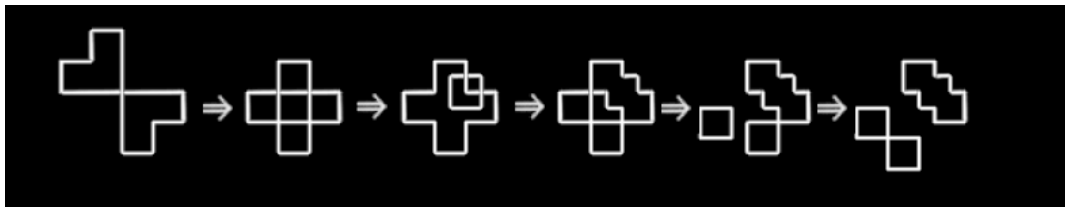
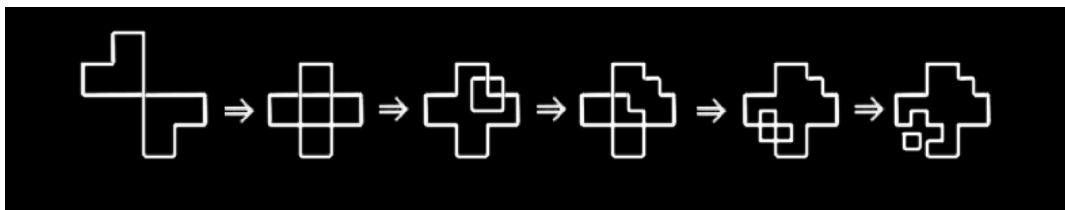
Grammaires de formes

« Shape grammars »



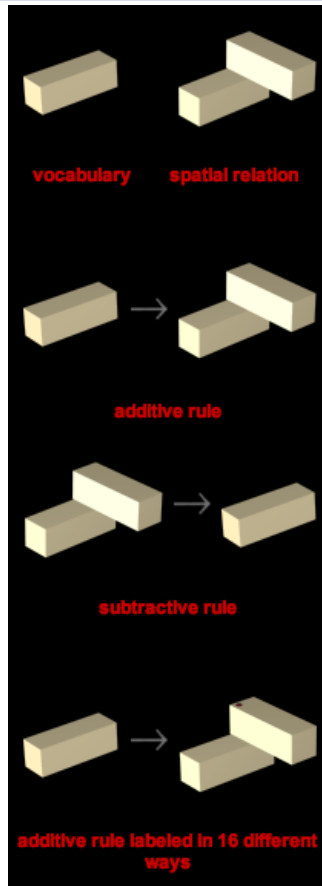
“Shape Grammar”
[Stiny&Gips, 1976]

source: <http://www.mit.edu/~tknight/JJDC/>



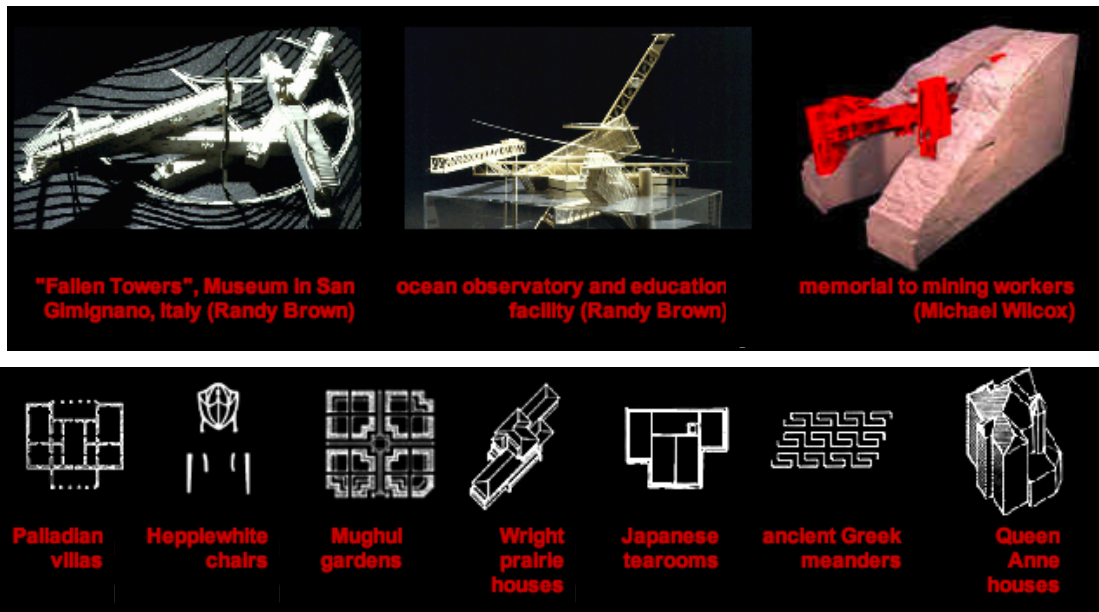
Deux exemples d'application possible des deux règles précédentes

source: <http://www.mit.edu/~tknight/JJDC/>



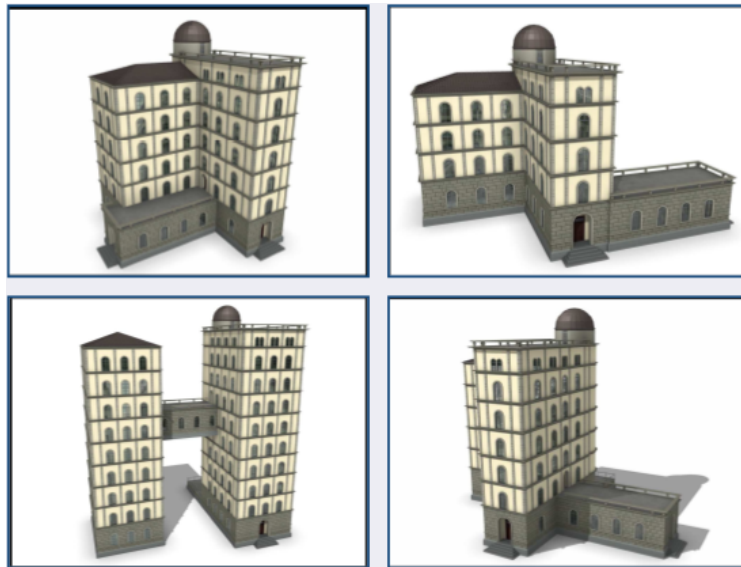
Règles en 3D

source: <http://www.mit.edu/~tknight/IJDC/>



Quelques exemples de projets architecturaux

source: <http://www.mit.edu/~tknight/IJDC/>

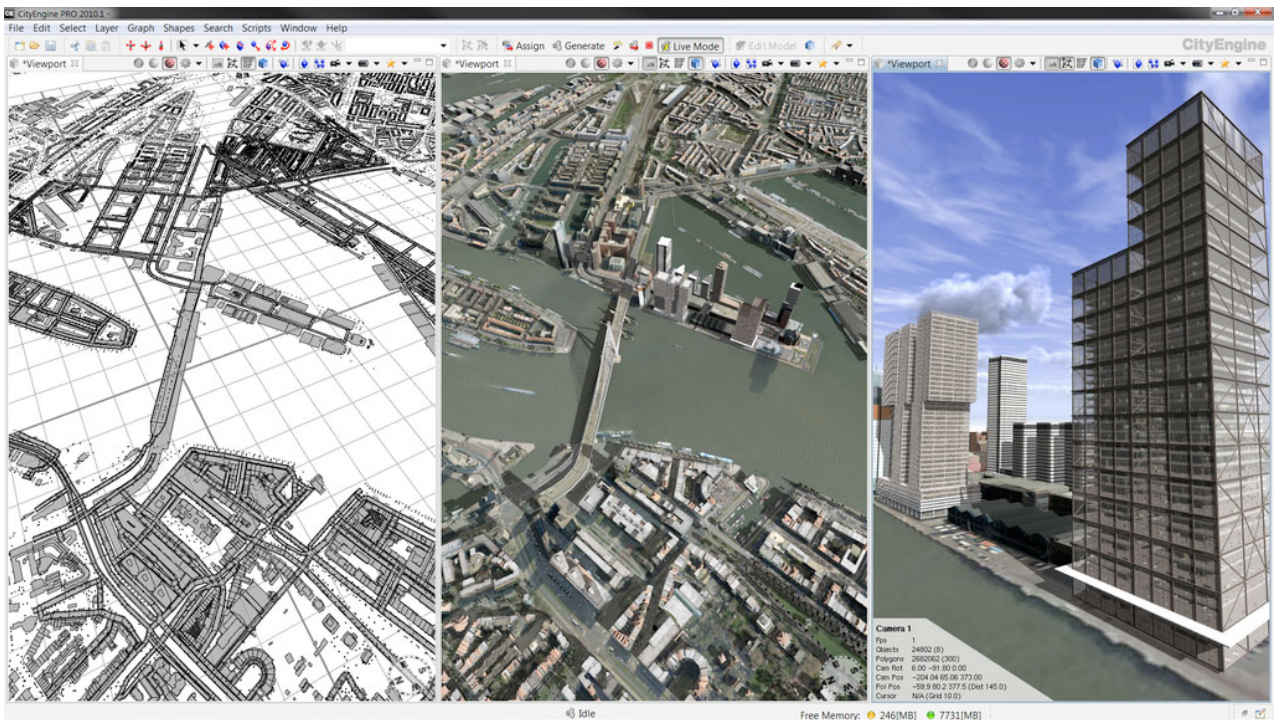


Génération d'immeubles

shape grammar
[Müller et al., SIGGRAPH 2006]

source: présentation Eric Galin, GDC 2008

32



ESRI CityEngine
(Esri R&D Center Zurich, 2008-...)

<http://www.esri.com/cityengine>

Synthèse

- Ce que l'on a vu
 - Grammaire
 - construire/dérouler un langage+grammaire,
 - créer/reconnaître une phrase...
 - L-Systems
 - construire/dérouler un L-System, une grammaire
 - savoir identifier le type d'un L-System

Bibliographie

- <http://www.mit.edu/~tknight/IJDC/>
- <http://algorithmicbotany.org/papers/#abop>
- <http://paulegre.free.fr/Teaching/gramm2.pdf>
- nombreuses ressources sur le web

