Simulation de systèmes électroniques grâce aux automates cellulaires

Alexandre Gordien

Institut Universitaire de Technologie de Créteil-Vitry
Département Génie Électrique et Informatique Industrielle

16 février 2011

Champs d'application

Un **automate cellulaire** est un algorithme qui, pratiquement, permet de modéliser un grand nombre de phénomènes physiques, biologiques ou sociaux :

- simulation du développement urbain
- simulation de la propagation de feux de forêt
- simulation d'avalanches
- simulation d'épidémie
- simulation de trafic autoroutier
- simulation des processus de cristallisation
- simulation des processus de percolation
- simulation de la croissance de plantes
- simulation de **systèmes électroniques**

Premier automate cellulaire

Dans les années 1940, **John Von Neuman** utilise les travaux de Stanislaw Ulam sur la croissance des cristaux pour mettre au point son modèle de système auto-réplicatif.

Il crée ainsi *the universal copier and constructor* : il démontre qu'un motif particulier peut produire une infinité de copies de lui-même.

Caractérisation

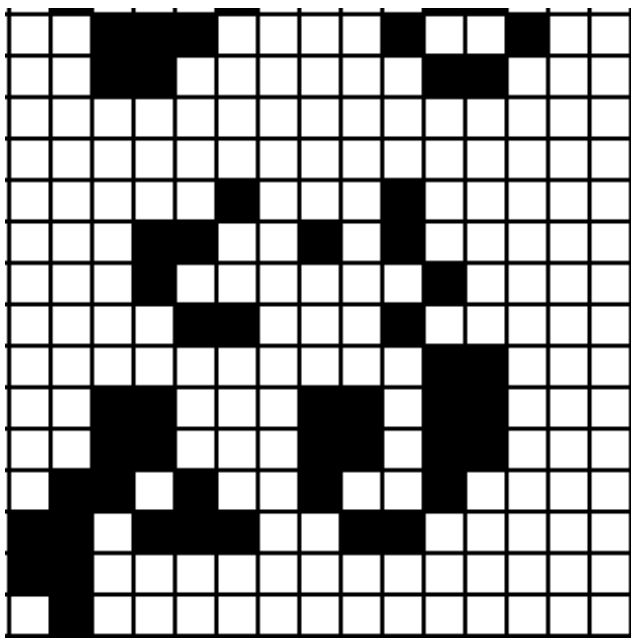
Chaque automate cellulaire est défini en fonction des trois points suivants :

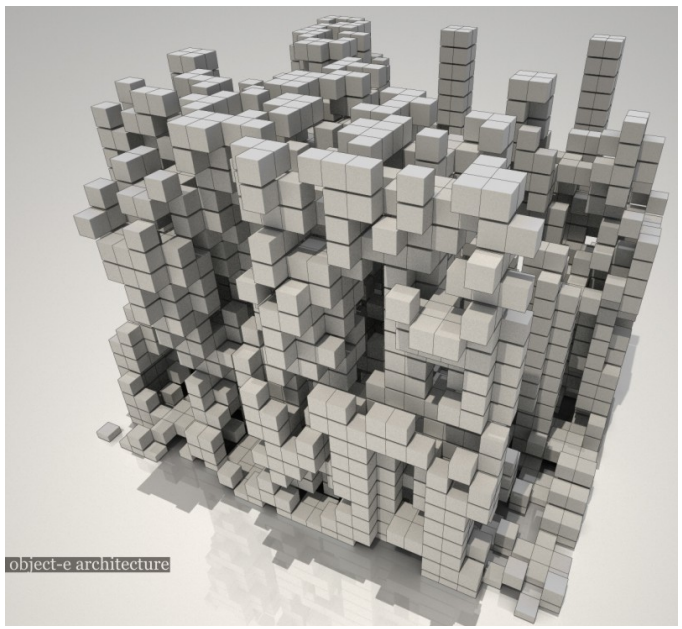
1. un espace
2. un voisinage
3. des états et des règles de transition

Un espace

L'espace d'un automate cellulaire est son **cadre d'évolution**. Il est divisé en **dimensions** puis en **cellules** :

- 2D le plus courant, cellules carrées ou hexagonales
- 3D cellules cubiques





Un voisinage

Le **voisinage** d'une cellule donnée est l'ensemble des cellules proches dont son état dépend. On trouve courament :

- l'environnement de **Neumann** ou les "premiers voisins"
- l'environnement de **Moore** ou les "premiers et seconds voisins"

Neumann

		N		
	O	C	E	
		S		

Moore

	NO	N	NE	
	O	C	E	
	SO	S	SE	

Des états et des règles de transition

Chaque cellule possède un **état**, calculé grâce à des **règles** en fonction des états des cellules voisines.

La variable d'état d'une cellule peut être :

- **booléenne** (cas des automates totalistiques)
- **multiple**

Les règles permettent de calculer l'état futur d'une cellule en fonction de son état actuel et des états de ses voisines.

Caractérisation

Nous définirons donc nos automates par ces trois points :

Caractérisation

Nous définirons donc nos automates par ces trois points :

Un espace

- 2D le plus courant, cellules carrées ou hexagonales
- 3D cellules cubiques

Caractérisation

Nous définirons donc nos automates par ces trois points :

Un espace

- **2D** le plus courant, cellules carrées ou hexagonales
- **3D** cellules cubiques

Un voisinage

- **Neumann** premiers voisins
- **Moore** premiers et seconds voisins

Caractérisation

Nous définirons donc nos automates par ces trois points :

Un espace

2D	le plus courant, cellules carrées ou hexagonales
3D	cellules cubiques

Un voisinage

Neumann	premiers voisins
Moore	premiers et seconds voisins

Des états et des règles de transition

États	variables booléennes ou multiples
Règles	calculent l'état futur

Le WireWorld

Le **WireWorld** est un automate cellulaire inventé en 1987 par **Brian Silverman**.

Il simule le fonctionnement de composants électroniques et de portes logiques.

Programmation d'un WireWorld

Deux points doivent être réfléchis :

Programmation d'un WireWorld

Deux points doivent être réfléchis :

1. le choix du langage de programmation :
   - Langage simple, rapididité de programmation et d'exécution

Programmation d'un WireWorld

Deux points doivent être réfléchis :

1. le choix du langage de programmation :
   - Langage simple, rapididité de programmation et d'exécution
2. la méthode de construction :
   - Programmation d'automates cellulaires simples avant le Wireworld

Le choix du langage de programmation

De nombreux langages possibles, parmi eux :

- Python : sa syntaxe est très simple et il peut être orienté objet mais c'est un langage interprété
- C : sa syntaxe est plus complexe et il n'est pas orienté objet mais c'est un langage compilé

Après tests, le langage C est le plus adapté : le calcul des images est beaucoup plus rapide qu'avec Python.

Méthodologie

La programmation *ex nihilo* d'un WireWorld est complexe. Son développement a donc été réparti en trois phases :

Méthodologie

La programmation *ex nihilo* d'un WireWorld est complexe. Son développement a donc été réparti en trois phases :

Phase 1 : Mise en place de la structure globale du programme

Méthodologie

La programmation *ex nihilo* d'un WireWorld est complexe. Son développement a donc été réparti en trois phases :

Phase 1 : Mise en place de la structure globale du programme

Phase 2 : Complexification des règles

Méthodologie

La programmation *ex nihilo* d'un WireWorld est complexe. Son développement a donc été réparti en trois phases :

- Phase 1 : Mise en place de la structure globale du programme
- Phase 2 : Complexification des règles
- Phase 3 : Chargement de l'espace depuis un fichier image

Croissance d'une structure fractale

Le premier automate cellulaire développe une figure à partir d'un point unique, "façon fractale".

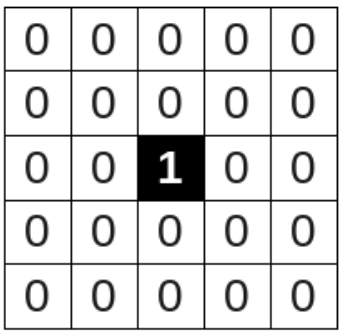
Croissance d'une structure fractale

Espace	2D, cellules carrées
Voisinage	Moore
États et règles	État 0 ou 1. Une cellule passe de l'état 0 à l'état 1 si, et seulement si, une seule de ses voisines est à 1. Sinon elle reste inchangée.

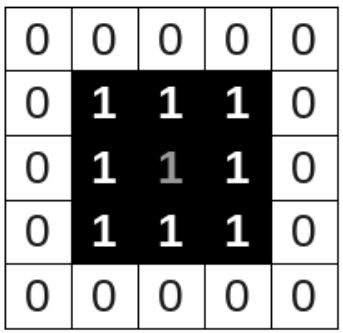
- Mise en place de la boucle principale
- Gestion des grilles (*double buffering*)
- Sauvegarde de la simulation - Conversion video

Croissance d'une structure fractale

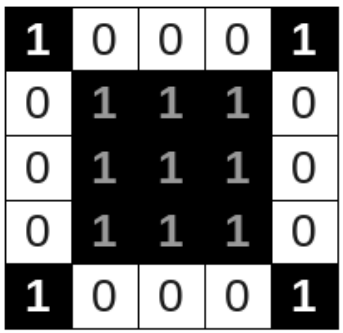
Fonctionnement :



Initialisation



Pas 1



Pas 2

Résultats :

`CroissanceStructureFractale.avi`

"Jeu de la vie" de John Conway

Le **Jeu de la vie** est certainement l'automate le plus célèbre.
Son concepteur, le mathématicien **John Conway**, en s'intéressant aux travaux de Von Neumann, met au point un nouvel algorithme décrivant une machine capable de s'auto-reproduire.

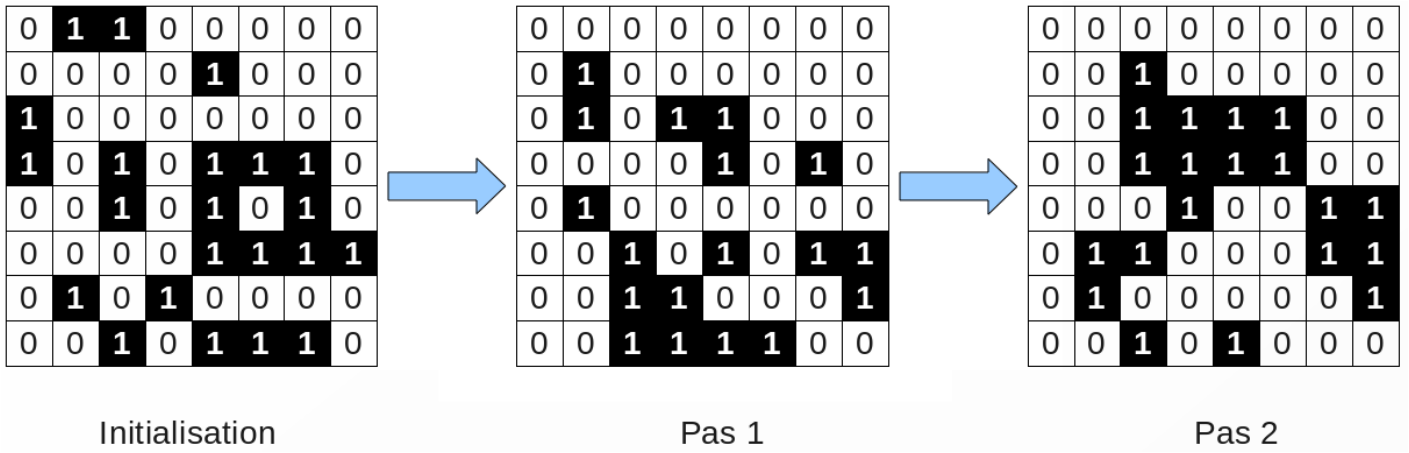
"Jeu de la vie" de John Conway

Espace	2D, cellules carrées
Voisinage	Moore
États et règles	Cellules Vivantes ou Mortes.
Naissance	La cellule (re)naît si exactement 3 de ses voisines sont vivantes
Survie	La cellule reste vivante si 2 ou 3 de ses voisines sont vivantes
Mort	La cellule meurt dans tous les autres cas

- Complexification des règles
- Graphe indicateur de la population de cellules vivantes
- Amélioration de l'IHM pour la préparation de la simulation

"Jeu de la vie" de John Conway

Fonctionnement :

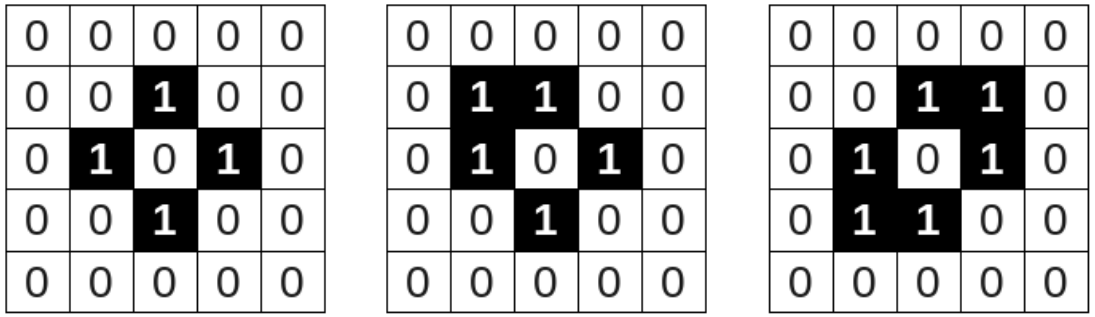


Résultats :

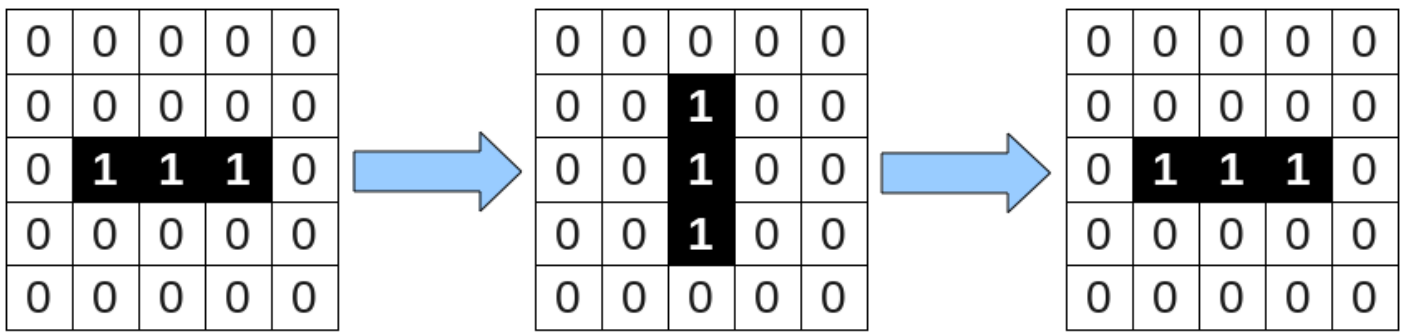
`GameOfLife5000-200.avi`

Figures remarquables

Après plusieurs itérations, on obtient des figures dites **stables** :



D'autres répètent le même motif à intervalles réguliers, les **oscillateurs** :



Résultats :

`GameOfLifeMotifs.avi` - `GameOfLife2000-80.avi`

Le WireWorld

Le WireWorld n'est pas un automate *totalistique*.

Le WireWorld

Espace 2D, cellules carrées

Voisinage Moore

États et règles États : 'vide', 'cuivre', 'tête d'électron', 'queue d'électron'

- Une cellule *vide* reste toujours *vide*
- Une *tête d'électron* devient toujours une *queue d'électron*
- Une *queue d'électron* devient toujours une cellule *cuivre*
- Une cellule *cuivre* reste toujours une cellule *cuivre* sauf si une ou deux de ses voisines sont des *têtes d'électron*. Elle devient alors une *tête d'électron*.

- Complexification des règles
- Réorganisation modulaire de la source
- Allocation dynamique de mémoire pour les grilles
- Chargement des images par les paramètres du *main*
- Réécriture de la fonction *cvGet2D*

Le WireWorld

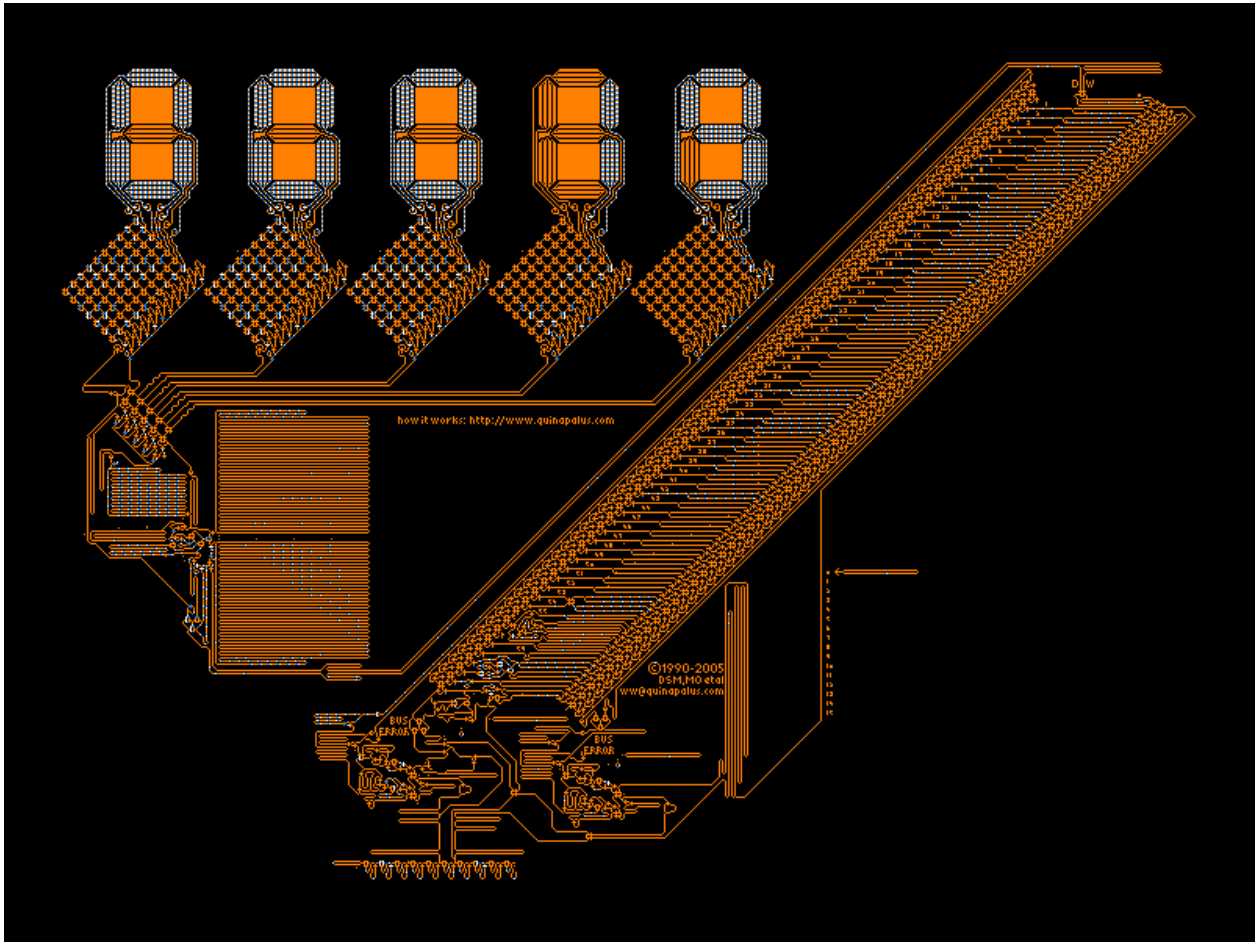
Le WireWorld développé simule l'action de divers composant électroniques parmis lesquels des diodes, une porte *OU EXCLUSIF* ou encore une bascule RS.

`WireWorldDiodes.avi` - `WireWorldXOR.avi` - `WireWorldRS.avi`

Ce sytème peut modéliser n'importe quel type de WireWorld, pouvu qu'on lui passe un fichier de base respectant certains codes.

Le WireWorld

On peut complexifier le WireWorld pour simuler des systèmes possédant, des entrées de données, des horloges, des mémoires et des afficheurs. Par exemple, le travail du groupe *Quinapalus* :



Conclusion

On peut simuler des systèmes logiques et électroniques complexes de façon simple grâce aux automates cellulaires.

Le WireWorld est plus un outil pédagogique que scientifique.

Après le WireWorld

Le modèle d'automate cellulaire que présente le WireWorld peut être adapté dans un tout autre domaine :

la résolution de labyrinthes

L'application des automates cellulaires au *pathfinding* serait une nouvelle méthode de résolution graphique de labyrinthes.

Sources

- Thèse de Pierre Guillon - *Automates cellulaires : dynamiques, simulations, traces*
- Jean-Philippe Rennard - *Automates cellulaires*
- Nazim FATES - *Les automates cellulaires, vers une nouvelle épistémologie ?* - Mémoire DEA Histoire et Philosophie des sciences
- Livre Groupe - *Automates Cellulaire : Jeu de la vie, Vers de Parterson, Vaisseau, Constructeur Universel, Boucle de Langton, Planeur, Fourmi de Langton*
- Laurent Signac - Université de Poitier TP : *Automates Cellulaire : Jeu de la vie, Simulation de feux de forêt, Propagation d'une épidémie*
- Banque de composants pour les WireWorld - `http://karl.kiwi.gen.nz/CA-Wireworld.html`
- ConwayLife (Wiki) - *List of common oscillators*
- John Von Neumann - *Théorie Générale et logique des automates*