

# Projet interfaces graphiques - 2011—2012

## SAC - Simulateur d'Automates Cellulaires

### But

Il s'agit d'implémenter un logiciel simulateur d'automates cellulaires — AC.

De quoi s'agit-il ? Plutôt que de longs discours inutiles nous vous reportons à la page Wikipédia sur le sujet qui est une introduction raisonnable aux ACs. Nous vous conseillons aussi d'autres lectures intéressantes. L'exemple canonique d'automate cellulaire a deux dimensions est le Jeu de la vie de John Horton Conway. Un autre chercheur très connu mais dont les travaux sur la question sont plus récent est Stephen Wolfram, il est le créateur de Mathematica et grand contributeur dans le domaine des automates cellulaires à une dimension (par ici).

Comme les lectures précédentes le laisse entrevoir, les automates cellulaires sont des machines (ordinateurs) dont les exécutions ont une nature « graphique », l'espace-temps de leur évolution produit généralement de jolis motifs que certains chercheurs explorent afin de les expliquer ou tentent de générer. Bref, ces machines sont programmables et constituent un modèle intéressant de machines massivement parallèles ou pour l'étude de certains systèmes complexes.

Ainsi le logiciel produit comprendra une partie visualisation graphique importante (visualisation des exécutions) et une partie de manipulation des automates (programmation) eux-mêmes qui nécessitera de penser raisonnablement l'interface.

On notera qu'il existe de nombreux outils de simulation d'automates cellulaires et que l'on peut s'en faire une idée en cherchant un peu sur le web et qu'on peut les utiliser dans un premier temps pour se familiariser avec les ACs. Mais le projet devra rester original et ne faire qu'éventuellement s'inspirer de l'existant.

De nombreux points ne sont pas spécifiés dans le sujet de façon que certaines libertés puissent être prises; attention car l'aventure peut se révéler délicate et il vaut mieux se reporter aux enseignants en cas de doute ou d'improvisation hasardeuse.

### Simulation/Visualisation

Il est demandé de permettre la simulation d'automates cellulaires unidimensionnels et bidimensionnels. Un automate cellulaire 1D est constitué d'une ligne de cellules et à une exécution correspondra donc un demi-plan (2D) de cellules; pour le cas des AC 2D, l'exécution prend place dans un espace 3D. La représentation d'espace tridimensionnel étant un problème en soi, il n'est pas demandé de le réaliser mais d'utiliser naturellement le temps comme troisième dimension...

D'autre part, la visualisation doit autoriser à réaliser des opérations de zoom avant ou arrière sur l'espace-temps (car certains phénomènes n'apparaissent qu'à grande échelle alors que d'autres ne sont observables qu'à petite échelle). Ces opérations de zoom doivent être compatibles avec l'exécution en cours (i.e. ne perturbent pas la simulation du modèle en cours).

La visualisation doit aussi permettre de « remonter » le temps si possible, attention car cela suppose l'existence d'un historique complet sur le passé d'une exécution en cours ce qui peut se révéler très très gourmand.

D'autre part, l'utilisateur final devra être autorisé à contrôler la vitesse d'exécution voire de choisir un

mode pas à pas, etc. Il devra aussi lui être permis de déplacer la fenêtre de visualisation comme bon lui semble sur l'espace-temps (i.e. scrolling). Bref, toute fonctionnalité qui pourrait permettre de mieux visualiser les exécutions d'automates cellulaires sera la bienvenue.

On notera que pour qu'une simulation puisse être réalisée, l'utilisateur devra avoir la possibilité d'entrer la configuration de départ de l'automate (son entrée).

## Programmation

Les ACs étant des machines programmables, il faut prévoir un « module » de programmation. De façon classique, mais pas forcément efficace, la programmation des AC utilise ce que l'on appelle une table de transition. C'est-à-dire une table qui étant donné l'état d'un voisinage fourni la valeur de la cellule considéré au temps suivant, on liste ainsi tous les états des voisinages possibles et on leur associé un état image. On devra pouvoir programmer, au moins, les automates cellulaires 1D dans ce mode. Une autre façon de programmer est de donner une description du jeu de vie, dans lequel on décrit l'image d'un voisinage et la fonction de son application à la cellule. Les règles de jeu de vie, si au moins 2 voisins.