

## Sujet 2

# Le jeu de la vie de Conway (\*)

### 1 Introduction

Le but de ce projet est d'étudier l'évolution d'une population d'individus dans ce qu'on appelle *le jeu de la vie*. Le jeu de la vie a été très populaire dans les années 1970-1980, en tant qu'exemple ludique d'*automate cellulaire* et d'application pratique de la micro-informatique alors en plein essor. Il a été proposé par John Horton Conway, célèbre mathématicien britannique, en 1970.

Ce système dynamique obéit à des règles très simples. Il est donc extrêmement facile à programmer :

- Il est défini sur un réseau à deux dimensions composé de mailles appelées *cellules*.
- Pour simplifier, les bords de ce réseau seront supposés périodiques.
- Il obéit à une dynamique à temps discret et l'état à un instant donné ne dépend que de l'état à l'instant précédent (pour la culture, on dit que la dynamique est *markovienne*).

Malgré cette simplicité, ce type de systèmes présente une très grande richesse de comportements qui a fait son succès.

### 2 Règles d'évolution

À un instant donné  $t_n$ , chaque case peut prendre deux états possibles, 0 (*morte*) ou 1 (*vivante*) (d'où la dénomination de jeu *de la vie*). L'état de chaque case à l'instant  $t_{n+1}$  ne dépend que de celui de ses 8 plus proches voisines et d'elle-même à l'instant  $t_n$ . Plus précisément, l'évolution d'une cellule dépend du nombre de cellules voisines qui sont vivantes :

- une cellule morte (état 0) à l'instant  $t_n$  ne change d'état (devient vivante) à l'instant  $t_{n+1}$  que si elle possède exactement 3 voisines déjà vivantes ;
- une cellule vivante à l'instant  $t_n$  ne change d'état (meurt) à l'instant  $t_{n+1}$  que si elle possède strictement moins de 2 voisines vivantes (elle meurt alors d'isolement) ou si elle possède strictement plus de 3 voisines vivantes (elle meurt alors de surpopulation).

Les mises à jour des cellules sont simultanées (et non pas séquentielles), c'est-à-dire qu'on ne remplace pas l'état d'une cellule par son état suivant avant d'avoir calculé les états suivants de toutes les cellules. Un exemple est donné sur la figure.

Nous allons dans la suite explorer la richesse des comportements observés en fonction de l'état initial à l'instant  $t_0$ .

### 3 Travaux

- La figure 1 présente un exemple d'évolution sur un pas de temps. Vérifier à la main cet exemple et continuer à le faire évoluer sur plusieurs pas de temps. Dans ce cas précis, la population finit par complètement disparaître. Combien de pas de temps faut-il pour atteindre cet état ?
- Écrire un programme C qui permet de suivre l'évolution d'une population de départ au cours du temps.

## Sujet 2. Le jeu de la vie de Conway (\*)

---

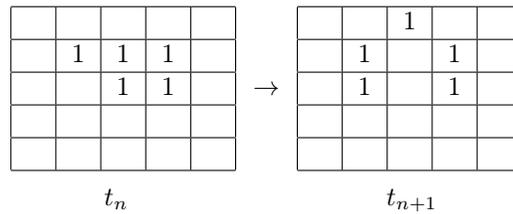


FIGURE 1 – Exemple d'évolution entre deux instants successifs du jeu de la vie. Les 1 représentent les cases vivantes, les cases mortes sont laissées vides.

- La visualisation des résultats pourra dans un premier temps se faire en affichant les états successifs dans un terminal `xterm` avec deux symboles différents pour les cellules mortes ou vivantes (par exemple un espace pour les cellules mortes et une étoile `*` pour les cellules vivantes). Au besoin, on pourra utiliser la fonction `delay(t)` donnée en annexe pour ralentir l'affichage. Une représentation graphique plus élaborée pourra éventuellement être envisagée dans un second temps (par exemple en écrivant les valeurs dans un fichier et en utilisant la commande `plot "fichier.dat" matrix with image de gunplot`).
- Le programme étant relativement simple à implémenter, on attachera une attention particulière à l'écriture de façon structurée, en faisant au maximum appel à des fonctions simples, par exemple une fonction `voisins(i,j)` qui calcule le nombre de cellules vivantes parmi les 8 voisins (attention aux conditions aux bords périodiques), ou encore une fonction `soupe()` qui crée une "soupe" de cellules aléatoirement vivantes ou mortes dans l'état initial.
- Tester l'évolution pour des conditions initiales différentes. La page suivante : [http://en.wikipedia.org/wiki/Conway's\\_Game\\_of\\_Life](http://en.wikipedia.org/wiki/Conway's_Game_of_Life) donne un certain nombre d'exemple simples. Les programmer. Commencer par le *planeur* (ou *glider* en Anglais). Qu'observez-vous ?
- Pour ces schéma simples, les raisons pour lesquelles ils sont stationnaires ou périodiques dans le temps peuvent être démontrées rigoureusement. Trouver quelques exemples ou cela est faisable.

Au-delà, Internet fourmille d'exemples plus ou moins sophistiqués ou ludiques, faites appel à votre imagination !

## 4 Annexe : Fonction delay

```
void delay(int milliseconds)
{
    long pause;
    clock_t now, then;

    pause = milliseconds*1000; // ajuster ce paramètre au besoin
    now = then = clock();
    while( (now-then) < pause )
        now = clock();
}
```