

1 Introduction

Le « jeu de la vie » (CONWAY, 1970) est un modèle simpliste de l'évolution d'une population de cellules qui naissent, vivent et meurent en fonction de leur voisinage. Les cellules sont disposées sur une grille (matrice carrée), dont chaque case peut contenir une cellule ou être vide à une étape donnée. La figure 1 ci-dessous présente un exemple de grille de dimension 5×5 .

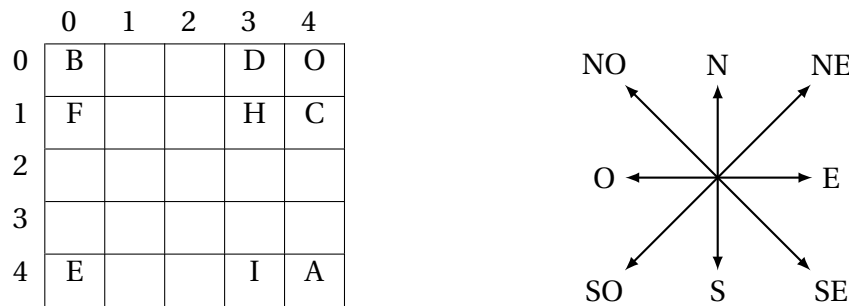


FIGURE 1 – Grille torique de dimension 5×5

La grille est dite torique, c'est-à-dire que chaque ligne (et chaque colonne) reboucle sur elle-même. Chaque case dispose donc de 8 cases voisines. Par exemple dans le cas de la case de coordonnées (0, 4) contenant la lettre O sur la figure 1, on a les 8 cases voisines suivantes :

- la case A située au nord en (4, 4) ;
- la case E située au nord-est en (4, 0) ;
- la case B située à l'est en (0, 0) ;
- la case F située au sud-est en (1, 0) ;
- la case C située au sud en (1, 4) ;
- la case H située au sud-ouest en (1, 3) ;
- la case D située à l'ouest en (0, 3) ;
- la case I située au nord-ouest en (4, 3).

L'évolution de la grille se fait au tour par tour en appliquant un ensemble de règles locales à chaque case. Ces règles prennent en compte l'état et le voisinage de la case et sont appliquées simultanément sur toutes les cases de la grille. Voici les règles que vous utiliserez :

Survie Chaque cellule entourée de 2 ou 3 cellules continue de vivre.

Mort Chaque cellule entourée de 4 cellules ou plus meurt de surpopulation. Chaque cellule entourée d'au plus 1 cellule meurt d'isolement. Si une cellule meurt, la case qui la contient devient vide.

Naissance Une cellule apparaît dans chaque case vide entourée d'exactly 3 cellules.

La figure 2a présente une grille initiale dans laquelle les cellules sont indiquées par la lettre V. La figure 2b présente la grille obtenue au tour suivant après application des trois règles ci-dessus.

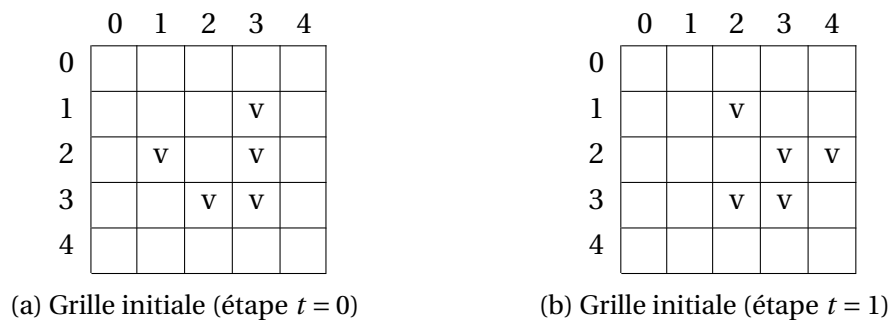


FIGURE 2 – Évolution de la population

2 Implémentation

La grille étant une matrice (carrée ou non), nous allons représenter la grille sous forme d'un tableau dynamique à deux dimensions, les cases de cette matrice ayant deux valeurs possibles :

- la valeur 0 si la case est vide;
- la valeur 1 si la case contient une cellule.

Vous manipulerez ces deux valeurs à l'aide de deux constantes symboliques : VIDE valant 0 et PLEINE valant 1.

- ① Implémenter la méthode `creerGrille` qui prend en paramètres une grille et ses dimensions. Cette méthode initialise la grille manuellement, c'est-à-dire que l'utilisateur peut définir si une cellule est vivante ou non pour toute la grille. □

Remarque : Vous avez deux fichiers d'exemple sur AREL. La saisie peut être automatisée par la redirection de flux. Par exemple, le programme pourra être appelé de la façon suivante :

```
cat jeuVie-Exemple-5x5.txt | ./progJeuDeLaVie
```

- 2 Implémenter la méthode `afficherGrille` qui prend en paramètres une grille et ses dimensions et l’affiche sur la sortie standard en symbolisant les cases vides par un point et les cellules par la lettre V. □

- 3 Pour parcourir le voisinage d’une case donnée, on implémente 8 fonctions renvoyant les valeurs des 8 cases voisines (cf. figure 1). Chacune de ces fonctions prend en argument une grille, ses dimensions et deux entiers positifs ou nuls x et y correspondant aux coordonnées de la case dont on veut connaître le voisinage :
 - `voisinNord` renvoie la valeur de la case de la grille située au nord de la case (x, y) ;
 - `voisinNordEst` renvoie la valeur de la case de la grille située au nord-est de la case (x, y) ;
 - `voisinEst` renvoie la valeur de la case de la grille située à l’est de la case (x, y) ;
 - `voisinSudEst` renvoie la valeur de la case de la grille située au sud-est de la case (x, y) ;
 - `voisinSud` renvoie la valeur de la case de la grille située au sud de la case (x, y) ;
 - `voisinSudOuest` renvoie la valeur de la case de la grille située au sud-ouest de la case (x, y) ;
 - `voisinOuest` renvoie la valeur de la case de la grille située à l’ouest de la case (x, y) ;
 - `voisinNordOuest` renvoie la valeur de la case de la grille située au nord-ouest de la case (x, y) .□

- 4 Implémenter la fonction `compteCelluleVoisinage` qui prend en argument une grille, ses dimensions et deux entiers positifs ou nuls x et y . Cette fonction renvoie le nombre de cellules présentes dans le voisinage de la case de coordonnées (x, y) dans la grille. □

- 5 Implémenter la fonction `calculerNouvelleValeur` qui prend en argument une grille, ses dimensions et deux entiers positifs ou nuls x et y . Cette fonction renvoie la valeur de la case (x, y) de la grille au tour suivant après application des règles. □

- ⑥ Implémenter la méthode `calculerNouvelleGrille` qui prend en argument une grille, ses dimensions et calcule la nouvelle grille en appliquant les règles simultanément à toutes les cases. Une fois la nouvelle grille calculée, elle sera stockée dans la grille passée en paramètres. □

- ⑦ Implémenter la méthode `jeuDeLaVie` qui prend en argument une grille, ses dimensions et le nombre d'étapes n à faire, et qui fait évoluer la grille n fois. Seule la grille à l'étape $t = n$ sera affichée. □

- ⑧ Modifier votre fonction `main` afin que les dimensions et le nombre d'étapes soient passés en argument au programme. Votre programme devra demander la saisie de la grille puis afficher la grille après avoir évolué n fois. □