TP - Jeu de la vie de Conway

Le « jeu de la vie » est une modélisation simplifiée à l'extrême d'évolution d'une population de cellules biologiques. Le jeu se déroule sur une grille à 2 dimensions. Chaque cellule de la grille est *vivante* ou *morte*. L'état des cellules évolue au cours du temps. L'état d'une cellule à l'instant $t + 1$ dépend de l'état de la cellule et de ses 8 voisins à l'instant t.

— Une cellule entourée de 3 cellules vivantes reste ou devient vivante,
— Une cellule entourée de 2 cellules vivantes conserve son état,
— Dans tous les autres cas elle meurt (ou reste morte).

A chaque instant, toutes les cellules évoluent en même temps.

Le but du TP est de programmer le jeu de la vie en Java.

La grille, en théorie infinie, est ici finie. On supposera que les bords opposés sont recollés, ainsi chaque cellule a 8 voisins (les 4 coins sont par exemple voisins les uns des autres).

Un objet de la classe `grille` représente l'état des cellules sur la grille, au moyen du tableau de booléens `etat`. Un autre tableau, `etatSuivant` est utilisé pour faire évoluer toutes les cellules en même temps.

Exercice 1 : Constructeur.

Ecrire le constructeur `public Grille(int width, int height)` qui initialise les attributs (notamment construit les attributs tableaux).

Exercice 2 : Initialisation.

Ecrire la méthode `public void vider()` qui initialise la grille. Chaque cellule devient morte.

Exercice 3 : Initialisation aléatoire.

Ecrire la méthode `public void randomInit()` qui initialise la grille. Chaque cellule devient vivante avec probabilité 0.5, indépendamment les unes des autres (utiliser `Math.random()` qui renvoie un double entre 0 et 1).

Exercice 4 : Nouvel état d'une cellule.

Ecrire la méthode `private boolean nouvelEtat(int i, int j)` qui renvoie le nouvel état de la cellule de coordonnées `i` et `j`, calculé en fonction de l'état de la cellule et de ses voisins.

Exercice 5 : Mise à jour de toutes les cellules.

Ecrire la méthode `public void metAJour()` qui fait évoluer toutes les cellules. Attention, si une cellule change d'état, c'est son ancien état qui est utilisé pour déterminer le nouvel état de ses voisins. C'est ici que l'attribut `etatSuivant` est utile.

Exécution.

Compiler et exécuter `Conway.java`. L'interaction avec le programme se fait comme suit :

— touche Entrée : lancer ou arrêter l'évolution
— touche Espace : vider la grille
— touche r : initialiser aléatoirement la grille
— touche + : accélérer l'évolution
— touche - : ralentir l'évolution
— clic sur la grille : change l'état d'une cellule.

Exemples de configurations intéressantes, à essayer :