

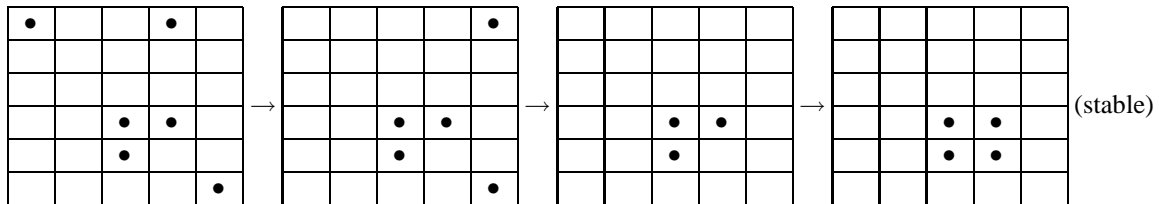
TP Jeu de la Vie

(d'après le sujet de DS de janvier 2002)

Le “jeu de la vie” consiste à faire évoluer un ensemble de cellules réparties sur une grille rectangulaire torique. À chaque cycle l'état de toutes les cellules est évalué **simultanément** et elles évoluent comme suit :

- Une cellule inactive s'active si elle est entourée d'exactly 3 cellules actives.
- Une cellule active ne survit que si elle est entourée de 2 ou 3 cellules actives.

Voici un exemple d'évolution sur 2 cycles (la situation d'arrivée est stable) :



Pour programmer ce problème, nous allons modéliser les cellules par des objets d'une classe `Cell`, la grille de cellules sera représentée par un objet de la classe `Environment` et enfin une classe `GameOfLife` qui gèrera l'évolution de l'environnement et son affichage. Ces classes appartiendront à un paquetage `jeudelavie`.

Pour cette réalisation, nous allons nous appuyer sur le paquetage `grid`. Vous trouverez les interfaces et classes de ce paquetage dans le fichier `jeu.de.la.vie.zip` sur le portail. Il est impératif de l'étudier avant de commencer. L'interface `grid.GridDisplayer` permet un affichage d'objet du type `grid.Grid`. Deux réalisations de cette interface sont fournies, l'une permet un affichage en mode texte (`grid.textGridDisplayer`) et l'autre en mode graphique (`grid.GraphicalGridDisplayer`).

Réfléchissez à l'ensemble de la modélisation avant de commencer à coder !

Les cellules Une cellule est définie par son état actif ou inactif. Afin de permettre l'affichage d'une cellule, en mode texte ou graphique, on décide qu'une cellule active sera caractérisée par le caractère '*' et la couleur `java.awt.Color.blue` et une cellule inactive par le caractère espace (' ') et la couleur blanche (utilisez des attributs de classe tels que `ACTIVE_COLOR`, `INACTIVE_COLOR`, `ACTIVE_CHAR`, `INACTIVE_CHAR`).

L'environnement L'environnement est une grille torique, afin de pouvoir facilement **réutiliser** les classes d'affichage en mode texte et graphique du paquetage `grid`, un objet qui permet de représenter l'environnement sera du type `grid.Grid`.

Un objet de cette classe sera défini par sa hauteur h , sa largeur l et un tableau de $h \times l$ instances de `Cell`. Il faut définir les méthodes qui permettent de modifier ou de récupérer la cellule à une position (cf `grid.Position`) donnée. Afin de gérer l'évolution de l'environnement, il faut pouvoir déterminer le nombre de cellules voisines actives d'une position donnée.

Le jeu de la vie Pour jouer au “jeu de la vie” il faut gérer l'évolution **simultanée** de toutes les cellules de l'environnement, et répéter cette évolution pendant un nombre donné de cycles. On veut un affichage du nouvel état de l'environnement après chaque étape de l'évolution (en fonction du jeu de la vie créé cet affichage pourra être graphique ou texte). Vous pouvez choisir de partir d'un état de l'environnement aléatoire (éventuellement avec un pourcentage de cellules actives à fixer) ou d'une configuration prédéfinie.

La classe correspondant à cet objet devrait ressembler (ce n'est pas “contractuel” !) à cela :

```
...
public class GameOfLife {

    private Environment environment; // l'environnement torique
    private GridDisplayer displayer; // pour l'affichage

    public GameOfLife (int width, int height) {
        environment = new Environment(width, height);
    }
}
```

```

/** initialise l'environnement avec des cellules dont
 * l'état initial est tiré aléatoirement
 */
public void randomInit() {
    ...
}

/** fixe le "gridDisplay"
 */
public void setGridDisplay(GridDisplay display) {
    ...
}

/** execute le jeu de la vie pendant nbSteps cycles
 * et affiche l'environnement après chaque cycle
 */
public void execute(int nbSteps) {
    ...
}

/** execute un seul cycle du jeu de la vie. Toutes les
 * cellules évoluent simultanément.
 */
private void executeOneCycle() {
    ...
}
} // GameOfLife

```

On veut... pouvoir, pour une initialisation de l'environnement aléatoire par exemple, exécuter un jeu de la vie par une ligne de commande de la forme :

```
java -jar jeudelavie.jar largeur_grille hauteur_grille %_actives_depart nb_cycles
```