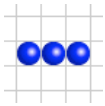
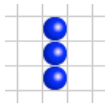
Devoir de programmation

I. Le jeu de la vie

Le jeu de la vie[1] se déroule sur un échiquier de taille infinie sans bord dont les cases, appelées cellules, peuvent prendre deux états : vivante ou morte. A chaque étape, l'évolution de chaque cellule est déterminée par l'état de ses huit voisines :

- Une cellule morte possédant exactement trois voisines vivantes devient vivante (elle naît).
- Une cellule vivante possédant deux ou trois voisines le reste, sinon elle meurt.

Ainsi, la configuration  donne au tour suivant la configuration  qui redonne ensuite la première.

Dans le cadre du devoir, nous nous intéressons à détecter quatre types d'évolution asymptotique :

- Mort : le jeu atteint la configuration sans cellule vivante
- Stable : le jeu atteint une configuration qui n'évolue plus
- Oscillateur : le jeu atteint une configuration que l'on retrouve à intervalle régulier
- Vaisseau : le jeu atteint une configuration que l'on retrouve à intervalle régulier mais déplacé d'un certain nombre de lignes et de colonnes.

Remarquer qu'un comportement du type mort est aussi du type stable, qu'un du type stable est du type oscillateur, qu'un du type oscillateur est du type vaisseau. Quand le comportement asymptotique d'un jeu n'est pas du type vaisseau, nous dirons qu'il est de type inconnu.

Etant donné un comportement I_0, I_1, ... , I_n, ... du type vaisseau, nous appelons taille de la queue "p", période "T"et déplacement "(A,B)", les valeurs minimales vérifiants $I_p + (A,B) = I_{p+T}$ où $I_p + (A,B)$ est la configuration I_p déplacée de A lignes et B colonnes.

II. Le sujet

Réaliser un programme du jeu de la vie en Java permettant de

- lire un jeu à partir de fichiers au format lif[2],
- simuler l'évolution d'un jeu,
- calculer le type d'évolution asymptotique avec ses caractéristiques.

---

[1] voir http://fr.wikipedia.org/wiki/Jeu_de_la_vie pour connaître les règles du jeu

[2] voir http://t0m.free.fr/jdlv/jdlv_lifexp.htm pour connaître le format lif

III. Que doit faire votre programme ?

Le programme java sera composé d'un unique archive jar **JeuDeLaVie.jar**. Ce programme devra s'exécuter dans un terminal avec les options suivantes :

- `java -jar JeuDeLaVie.jar -name` affiche vos noms et prénoms
- `java -jar JeuDeLaVie.jar -h` rappelle la liste des options du programme
- `java -jar JeuDeLaVie.jar -s d fichier.lif` exécute une simulation du jeu d'une durée d affichant les configurations du jeu avec le numéro de génération.
- `java -jar JeuDeLaVie.jar -c max fichier.lif` calcule le type d'évolution du jeu avec ses caractéristiques (taille de la queue, période et déplacement); max représente la durée maximale de simulation pour déduire les résultats du calcul.
- `java -jar JeuDeLaVie.jar -w max dossier` calcule le type d'évolution de tous les jeux contenus dans le dossier passé en paramètre et affiche les résultats sous la forme d'un fichier html.

Pour candidater à une note supérieure à 16/20, vous devez adapter votre programme pour aussi gérer des échiquiers de taille finie pour deux types de mondes :
a) Mondes circulaires : une cellule sur un bord est voisine avec la cellule du bord opposé
b) Mondes avec frontières : une cellule sur un bord n'a pas de voisine au delà du bord.

IV. Une structure de données imposée

Afin de manipuler efficacement un échiquier de taille infinie, vous stockerez l'ensemble des positions des cellules vivantes dans une liste simplement chaînée sans utilisée l'API Java (LinkedList ou ArrayList). Cette liste sera triée par numéro de ligne, puis par numéro de colonnes. Toute autre structure de données est à bannir.

V. Quoi et quand rendre ?

1. Quand ? ***le vendredi 4 mai 2018 à 12h00***
2. Envoyer par email à votre chargé de TD, votre mini-rapport au format pdf, votre archive jar et un dossier compressé contenant vos fichiers de jeux ".lif" utilisés pour votre expérimentation.
3. Donner à votre chargé de TD en version papier :
   a) votre mini-rapport de projet
   b) vos résultats d'expérimentation

VI. Quelles références disponibles sur la toile et à la bibliothèque des sciences

1. http://fr.wikipedia.org/wiki/Jeu_de_la_vie pour connaître les règles du jeu
2. http://t0m.free.fr/jdlv/jdlv_lifexp.htm pour connaître le format lif
3. http://www.termsys.demon.co.uk/vtansi.htm contient des commandes de contrôle du terminal
4. https://docs.oracle.com/javase/7/docs/api/ API java
5. http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html donne un ensemble d'informations suffisantes pour l'écriture de votre rapport de projet sous la forme de commentaires utilisables par l'outil javadoc.

6. La programmation en pratique, Brian W. Kernigan et Rob Pike. Editeur Vuibert informatique. Lire le premier chapitre sur la manière de commenter un programme et choisir des noms de variables

VII. Un guide pour l'élaboration de votre projet

Etape 1 : Se familiariser avec le jeu de la vie (voir réf. 1), le format lif (voir réf. 2) et récupérer sur la toile un ensemble de jeux au format lif.

Etape 2 : Ecrire un constructeur initialisant un jeu à partir d'un fichier au format lif et une méthode toString(). Tester vos méthodes sur votre ensemble de jeux.

Etape 3 : Ecrire une méthode calculant la génération suivante d'un jeu. Tester votre méthode sur votre ensemble de jeux. Cette étape est sûrement la plus délicate.

Etape 4 : Etudier et exécuter le Programme 2 sur un terminal. Des compléments d'informations peuvent être obtenus dans les références 3, 4 et 8. Ecrire et tester votre programme d'animation.

```
import java.io.*;
import javax.swing.Timer;
import java.awt.event.*;
import java.awt.Event;

public class Programme2 {

      static public void main(String[] args) {
            final String message = args[0] + "          ";

            System.out.print((char)Event.ESCAPE + "7");

            Timer t = new Timer(200, new ActionListener() {
                  int i=0;
                  public void actionPerformed(ActionEvent e) {
                        System.out.print((char)Event.ESCAPE + "8");
                        System.out.println(message.substring(i%message.length())
                              + message.substring(0,i%message.length()));
                        System.out.println("t = " + i);
                        i++;
                  }
            });
            t.start();

            try {
                  System.in.read();
            }
            catch (IOException e){}

            t.stop();
      }
}
```

Programme 2 : Une animation sur un terminal

Etape 5 : Reprendre complètement votre programme et le rendre lisible (voir réf. 7). Votre programme doit respecter les normes de codage décrite dans la réf. 5. Commenter toutes les classes, méthodes et attributs de façon à permettre l'extraction de ces commentaires par l'outil javadoc (voir réf. 5, 6, 8). A la fin de cette étape, la commande javadoc doit rendre une première version de votre rapport.

Etape 6 : Ecrire votre programme calculant le type de comportements asymptotiques d'un jeu et donnant la valeur de ses caractéristiques (taille de la queue, période et déplacement). Les principes de ce calcul sont les suivants :

a) Calculer une génération du jeu contenu dans la période :

- prendre deux générations du jeu : une dans la position initiale et une autre dans la position suivante,
- pendant que la première génération avance d'un pas, la seconde avance de deux pas.
- l'algorithme s'arrête quand les deux générations atteignent la même configuration à un déplacement près,
- la génération obtenue est la génération recherchée
- si l'algorithme échoue, le jeu est de type inconnu

b) Calculer la taille de la période,

c) Calculer la taille de la queue.

Etape 7 : Refaire l'étape 5

Etape 8 : Ecrire votre programme calculant le type de comportements d'un ensemble de jeux contenu dans un dossier. Votre programme doit produire les résultats aux formats HTML.

Etape 9 : Refaire l'étape 5

Etape 10 : Ecrire le programme principal et refaire l'étape 5

Etape 11 : Adapter votre programme pour traiter des échiquiers de taille finie.

Etape 12 : Refaire l'étape 5