Rapport de TP 4MMPOO : Simulation Orientée-Objet de systèmes multiagents

MAHIEU Lucas (SL_PHELMA)
DEVALON Hugues (SLE_PHELMA)

10 novembre 2015

Préambule L'objectif de ce TP est de développer en Java des applications permettant de simuler de manière graphique des systèmes multiagents. Dans un premier temps, nous nous intéresserons à trois systèmes de type automate cellulaire : le jeu de la vie de Conway, un jeu de l'immigration, et le modèle de ségrégation de Schelling. Dans un second temps, nous nous intéresserons à la simulation d'un système de mouvement d'essaims auto-organisés : le modèle de Boids.

1 Automates Cellulaires

1.1 Jeu de la vie de Conway :

Nous présentons en figure 1 page 1 la représentation UML des classes permettant la Simulation de l'automate cellulaire du jeu de la vie de Conway. Nous avons explicité les attributs privées des classes pour que vous compreniez mieux nos choix de conception.

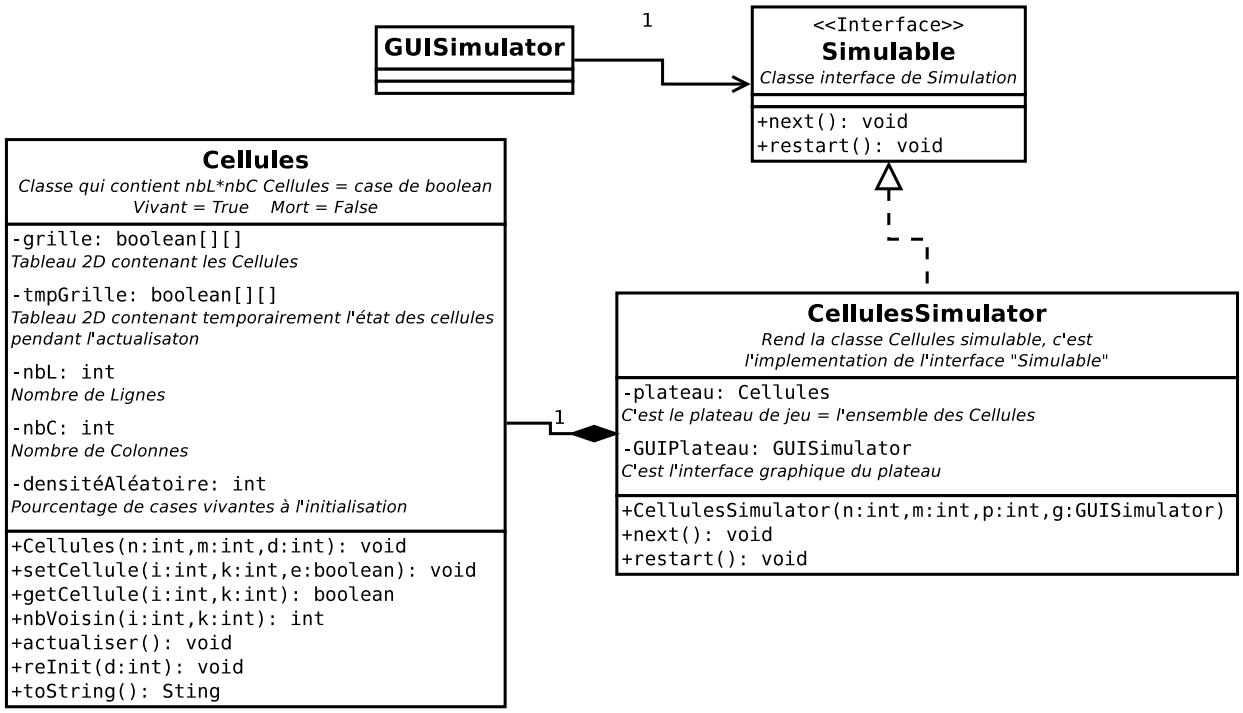

FIGURE 1 – Diagramme des classes du jeu de la vie de Conway

Comme vous pouvez le constater, nous avons utiliser des classes assez 'génériques' pour qu'elle soit facilement réutilisables pour les autres jeux. D'ailleurs on peut remarquer que notre conception est presque identique à celle proposée pour le jeux de balles. En effet, ici nous manipulons un tableau de *boolean* et non plus un tableau de *Point*. Ainsi pour l'adapter à d'autres jeux, il suffirait de modifier les booleans par des entiers, des couleurs, ou autre états plus complexes.

Pour tester ces classes, nous avons utiliser deux classes de tests : l'une pour faire une simulation textuelle, l'autre pour tester la simulation graphiquement.

1.2 Jeu de l'immigration :

Les seules différences avec le jeux de la vie sont
— Qu'une cellule peut prendre un nombre n_e d'états et non plus seulement 2.
— Les règles de changement d'états.

Du fait que notre conception est générique, nous avons réutilisé le jeu de la vie en y ajoutant une seule chose : le nombre d'états possibles pour une cellules. Ainsi nous pouvions supprimer l'attribut de densité utile à l'initialisation du jeu de la vie, et modifier la règle dans la méthode *void actualiser( )*.

Nous avons fait le choix de créer une correspondance entre les *Entiers* qui représente l'état d'une cellule et une *Couleur* qui sera affichée dans sa simulation graphique.

1.3 Le modèle de Schelling :

A TOI DE COMPLETER CETTE PARTIE LA POUR EXPLIQUER CE QUE TU AS UTILISEE ET PQ

2 Un modèle d'essaims : les *boids*

2.1 Troupeaux de boids

2.1.1 Un gestionnaire à évènements discrets :

2.1.2 Modification de votre simulateur :

2.1.3 Simulation de plusieurs groupes de boids :

3 Bilan