Programmation Orienté Objet

Rapport du Projet

«JEU DE LA VIE»

JACQUETTE Pierrick & STASYSZYN Romain
21305551 & 21305734

Table des matières

Introduction

Ce projet a été réalisée au sein de l'enseignement Complément de programmation orientée objets.

Notre projet est de réaliser un jeu de la vie (ou Game of Life) en Java. Ce jeu permettra de jouer dans les règles basiques mais aussi des extensions. Le jeu de la vie est un automate cellulaire crée par Conway en 1970.

Le but de ce projet est de mettre en application les principes fondamentaux de la programmation objet , c'est à dire l'héritage entre les classes, la gestion des interfaces ou des classes abstraites.

Pour bien répondre au problème, nous nous inspirerons du cycle général de la conception objet que nous avons vu en cours depuis le début de la licence. Cette méthode de conception nous permettra d'avoir une présentation claire et structurée de notre projet.

Dans une première partie, nous rappellerons brièvement les règles de base du jeu de la vie ainsi que les variantes possibles.

Ensuite, nous vous montrerons comment exécuter le programme

Enfin, pour terminer nous parlerons des différents problèmes que nous avons rencontré lors de la réalisation du projet et comment nous y avons remédié.

Règles du Jeu

Ce projet réalise le concept du « jeu de la vie » mais cela n'est pas un jeu comme on pourrait l'entendre, il n'y a aucun joueur a proprement parlé, c'est un automate cellulaire, chaque cellule à un état qui varie en fonction de ses voisins sur la grille.

Le nombre de voisins peut varier en fonction des extensions : 8 ou 6, cela dépend du type de la grille, soit rectangulaire soit hexagonal. C'est donc une grille à deux dimensions qui sera torique pour simuler une grille infinie.

Le nombre d'états d'une cellule dépend de la règle choisie par l'utilisateur : plusieurs possibilités d'états pour chaque cellule et plusieurs possibilités de règles. Chaque règle définissant l'évolution d'une cellule à chaque étape.

1. Conway

Une cellule qui est morte ayant exactement trois voisines vivantes devient vivante.

Une cellule qui est vivante ayant deux ou trois voisines vivantes survit, sinon elle meurt.

2. Day & Night

Une cellule qui est morte, vit à l'étape suivante si elle est entourée de trois, six, sept ou huit cellules voisines vivantes.

Une cellule qui est vivante survit à l'étape suivante si elle est entourée de trois, quatre, six, sept ou huit cellules voisines vivantes.

3. HighLife

Une cellule qui est morte y naît à l'étape suivante si elle est entourée de trois ou six voisines vivantes.

Une cellule qui est vivante survit à l'étape suivante si elle est entourée de deux ou trois cellules vivantes.

4. Immigration

Une cellule qui est morte, naît à l'étape suivante si elle est entourée de trois voisines.

Une cellule vivante survit à l'étape suivante si elle est entourée de deux ou trois cellules vivantes.

Lorsqu'une cellule naît, elle prend l'état de la majorité des trois cellules vivantes qui lui ont donné naissance.

Exécution

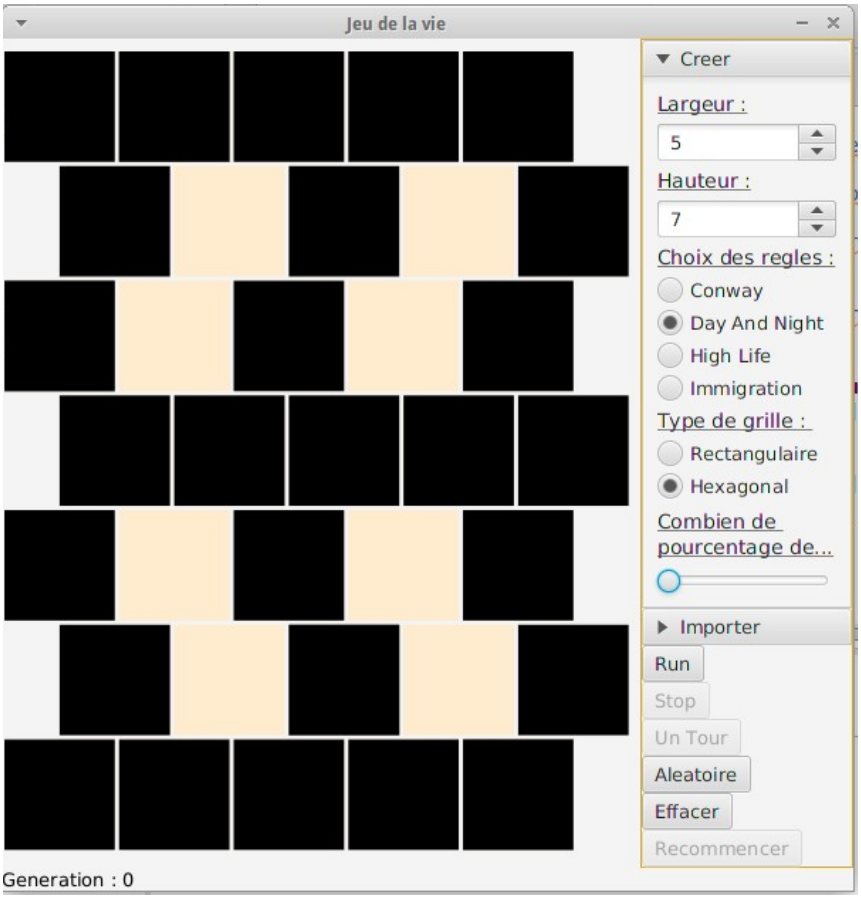
Pour les deux versions il faut télécharger le dossier, l'extraire, aller dans le dossier console ou graphique en fonction de si vous voulez un affichage en console ou dans une interface graphique. L'interface graphique utilise javafx, le code utilise les nouveautés de java 8, il faut donc java 8 sur votre machine sinon cela ne compilera pas correctement.

1. Interface graphique

```
$ javac controleur/*.java model/*.java vue/*.java
```

```
$ java controleur.Main
```

Par la suite vous pourrez dans la fenêtre graphique choisir vos options, vous pouvez cliquer sur les cellules pour les rendre vivantes ou mortes.



2. Console

```
$ javac controleur/*.java model/*.java
```

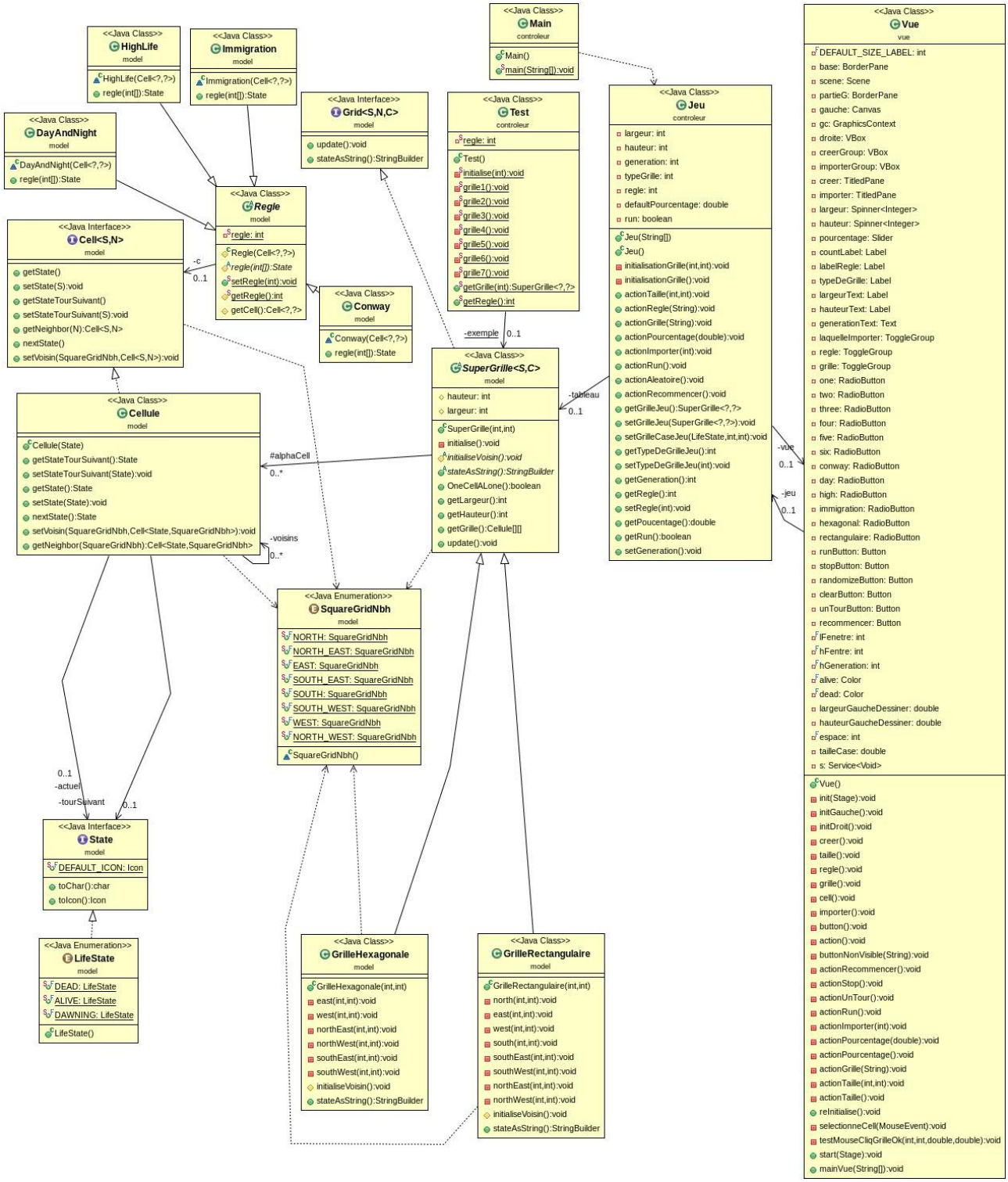
```
$ java controleur.Main
```

Puis des configurations initiales sont déjà crées, il faut choisir l'option Importer, taper 2

```
Voulez-vous créer votre configuration ou importer ?
Creer(1) ou Importer(2)
```

Si vous taper 2, vous pourrez alors choisir une des configurations déjà existantes

Diagramme de classes



Conclusion

Ce jeu de la vie en langage Java fut très intéressant à concevoir et à programmer. Ce projet nous a permis de bien comprendre les qualités de la programmation orientée objets.

Nous avons essayé d’utiliser au mieux ses caractéristiques tels que l’héritage, les classes abstraites ou bien la généricité.

Les propriétés de ce mode de programmation sont très pratiques et permettent une bien meilleure conception et réalisation d'un projet à plusieurs.

Au début du projet, nous avons trouvé les différentes classes et méthodes à créer grâce à l’utilisation de diagrammes de classes, même si la réflexion était déjà commencée, comme certaines classes étaient fournis. Cette démarche permet d’avoir une réflexion structurée tout au long de l'avancement.