

# Simmulation du jeu de la vie - conception logiciel 2022-2023

mis en page par  
DIARE Youssouf 22008756  
OLANGASSICKA Franck 22112035  
KONE saybou 21911516  
YATTOURA Mohamed 22012262  
L2 info,  
Groupe 2

25 avril 2023



**UNIVERSITÉ  
CAEN  
NORMANDIE**

# Sommaire

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Présentation du projet . . . . .	2
1.2	Objectif du projet . . . . .	2
1.3	organisation et répartition des tâches . . . . .	2
<b>2</b>	<b>Architecture du programme</b>	<b>2</b>
2.1	les classes . . . . .	3
2.2	les packages . . . . .	3
<b>3</b>	<b>Éléments techniques</b>	<b>3</b>
3.1	Classe Cellule . . . . .	4
3.2	Grille . . . . .	4
3.3	L'interface graphique . . . . .	4
3.4	Manuel d'utilisation . . . . .	5
3.4.1	Jeu graphique . . . . .	5
3.5	Difficulté rencontrée : Algorithme HashLife . . . . .	6
<b>4</b>	<b>conclusion</b>	<b>6</b>
4.1	Améliorations possibles . . . . .	7

# 1 Introduction

L'unité d'enseignement Projet 1 a pour objectif de nous faire comprendre le concept de la programmation orientée objet, nous initiés aux implémentations des algorithmes et parfaire notre maîtrise de java.

Pour cela des groupes de 4 d'étudiants ont été formés et plusieurs projets ont été proposés, Éditeur de livres dont vous êtes le héros, Interpréteur de systèmes de Lindenmeyer, Interpréteur de programmes chimiques, etc. Notre groupe a choisi le projet Jeu de la vie.

## 1.1 Présentation du projet

Le jeu de la vie est un projet informatique qui simule l'évolution des cellules d'une grille bidimensionnelle .Il a été crée par le mathématicien Jhon Horton Conway en 1970. Ce jeu se déroule en plusieurs étape ou génération , au départ l'utilisateur définit une grille de cellules qui peuvent être vivante ou morte et à chaque étape les cellules évoluent en fonction des règles suivantes :

- Une cellule vivante qui a moins de deux voisins vivants meurt, par solitude.
- Une cellule vivante qui a plus de trois voisins vivants meurt, par surpopulation.
- Une cellule morte qui a exactement trois voisins vivants devient vivante, par reproduction.
- Les cellules qui ont deux ou trois voisins vivants restent en vie.

## 1.2 Objectif du projet

Le but de ce projet est de développer un programme permettant la simulation de l'évolution de cellule sur une surface. Dans un premier temps, il s'agit de développer le moteur du jeu (cf. règles) puis le doté d'une interface graphique permettant la visualisation des générations. Dans un second temps, il s'agira alors d'implémenter l'algorithme Hashlife pour accélérer la resolution puis étendre ce système aux automates cellulaires neuraux.

## 1.3 organisation et répartition des taches

Dans un premier temps, nous avons décidé de travailler ensemble pour essayer de bien comprendre le projet, identifier les objectifs et élaborer un plan de travail. A la suite de cette première phase de travail, on a identifié et classifié dans l'ordre de priorité les différents objectifs :

Dans un premier temps, nous avons cherché à modéliser les digrammes de classe sur lesquels se basera le jeu, et développer le moteur du jeu basé sur les règles du jeu. Puis une interface graphique permettant à l'utilisateur de jouer.

Dans un second temps, nous avons essayé d'implémenter l'algorithme hashlife.

Pour atteindre nos objectifs à temps et permettre à chacun d'entre nous de bien comprendre le projet,nous nous sommes séparé en binôme .KONE Saybou et DIARE Youssouf se sont concentrés sur le moteur de jeu et l'algorithme hashlife tandis que YATTOURA et OLAN-GASICKA Franck se sont chargé de la réalisation de l'interface graphique .Pour faciliter ce travail de groupe nous avons eu à se trouver souvent pour travailler en collaboration.

# 2 Architecture du programme

Pour ce projet nous avons opté pour une architecture MVC (Modèle Vue Contrôleur). Nous avons donc décomposé le projet en 6 packages :

**GameOfLife** : Il contient toutes les classes utilisées comme model de jeu à savoir la classe Cellule, Grid, Regle, MyIterator, etc.

**GameOfLife.Ihm** : Dans ce package se trouve toutes les classes de la Vue. Images : Qui contient toutes les images utiliser dans ce projet.

**hashlife** : Ce package contient les classes qui devait nous permettre l'implémentation de l'algorithme hashlife.

**GameOfLife.execution** : Ce package contient les classes d'execution du jeu.

Avec cette architecture nous avons obtenus les diagrammes de classes suivants :

## 2.1 les classes

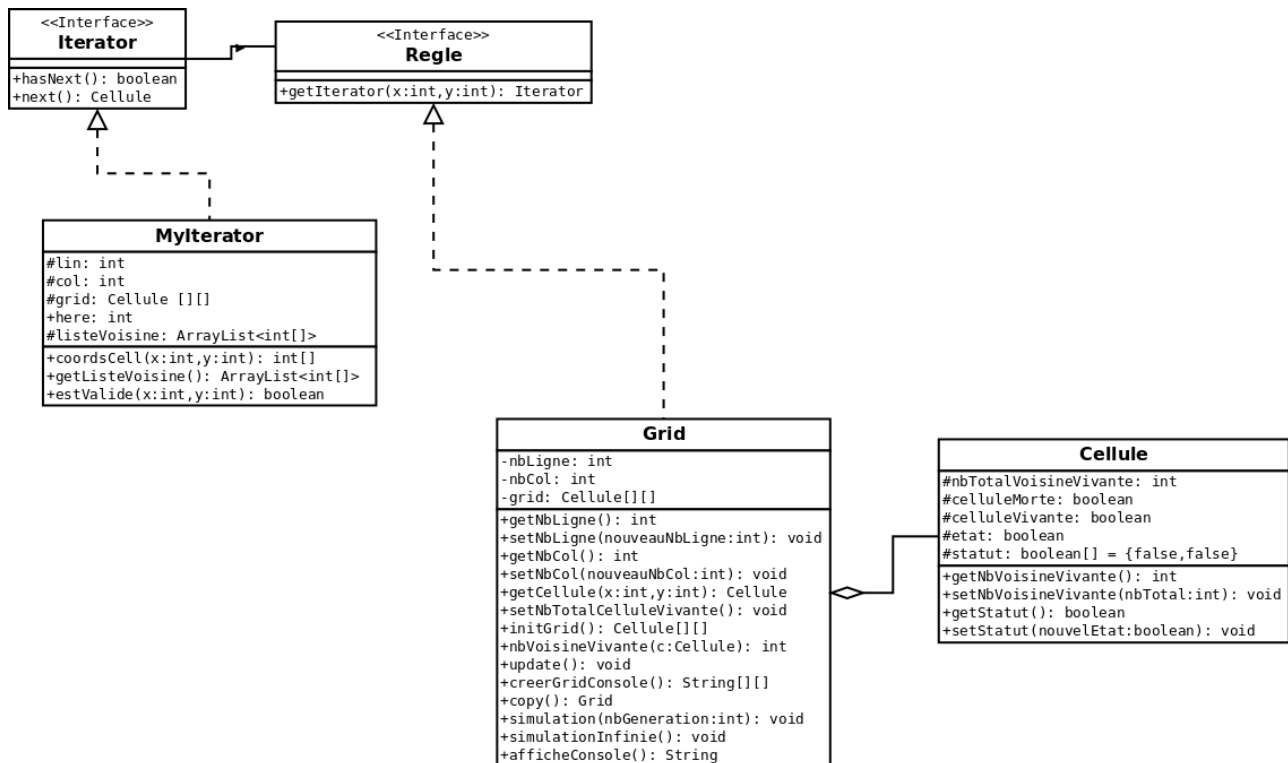


Diagramme de classe du moteur du jeu

## 2.2 les packages

faire un diagramme des classes

## 3 Éléments techniques

Comme on l'avait dit plus haut la simulation du jeu de la vie utilise une grille composé de cellules ,dans cette partie nous allons vous expliquer les points technique du projet.Avant de coder la classe Grille nous avons défini un type cellule qui aura les arguments et les méthodes qui nous permettrons d'initialiser et de faire évoluer notre Grille future .

Notre moteur de jeu est basé sur une grille à deux dimensions qui représente la surface sur laquelle les cellules évolueront en fonction du nombre de leur cellules voisines vivante ,et pour connaître les cellules voisines nous avons utiliser le pattern iterator de java qui nous donne le moyen de les trouver

### 3.1 Classe Cellule

la classe cellule est un element essentielle dans notre moteur du jeu .c'est le principale constituant de la grille et nous avons repr  senter une cellule par son nombre de voisines totales ,son   tat (true :cellule vivante, false : cellule morte ).

### 3.2 Grille

La classe Gille est celle qui utilise toutes ces classes pr  alablement d  finies, elle contient   galement des m  thodes comme celle qui permettent de construire le grille du jeu.

La grille est la classe principale du projet,son importance r  side dans le fait qu'elle permet de repr  senter de mani  re discr  te l'espace dans lequel les cellules   voluent. La grille est un ensemble de cellules qui sont dispos  es en lignes et en colonnes, et chaque cellule peut   tre vivante ou morte.

Les r  gles du jeu de la vie sont appliqu  es    chaque cellule de la grille, en fonction de l'  tat de ses voisines imm  diates.

### 3.3 L'interface graphique

Pour r  aliser l'interface graphique, nous avons utiliser un JFrame pour avoir une fen  tre dans laquelle on a plac   au sud un JPanel qui contient les boutons (JButton) permettant d'iteragir avec la grille. En suite au nord nous avons plac   un JComponent (GridGraphique) contenant la grille.

GridGraphique est une classe qui h  rite de JComponent    partir de laquelle nous importons les images du package Images repr  sant certains objets du jeu comme les cellules vivantes et les cellules mortes    la m  thode chargementImages(). Chaque image est dessin  e en fonction de la cellule correspondant.

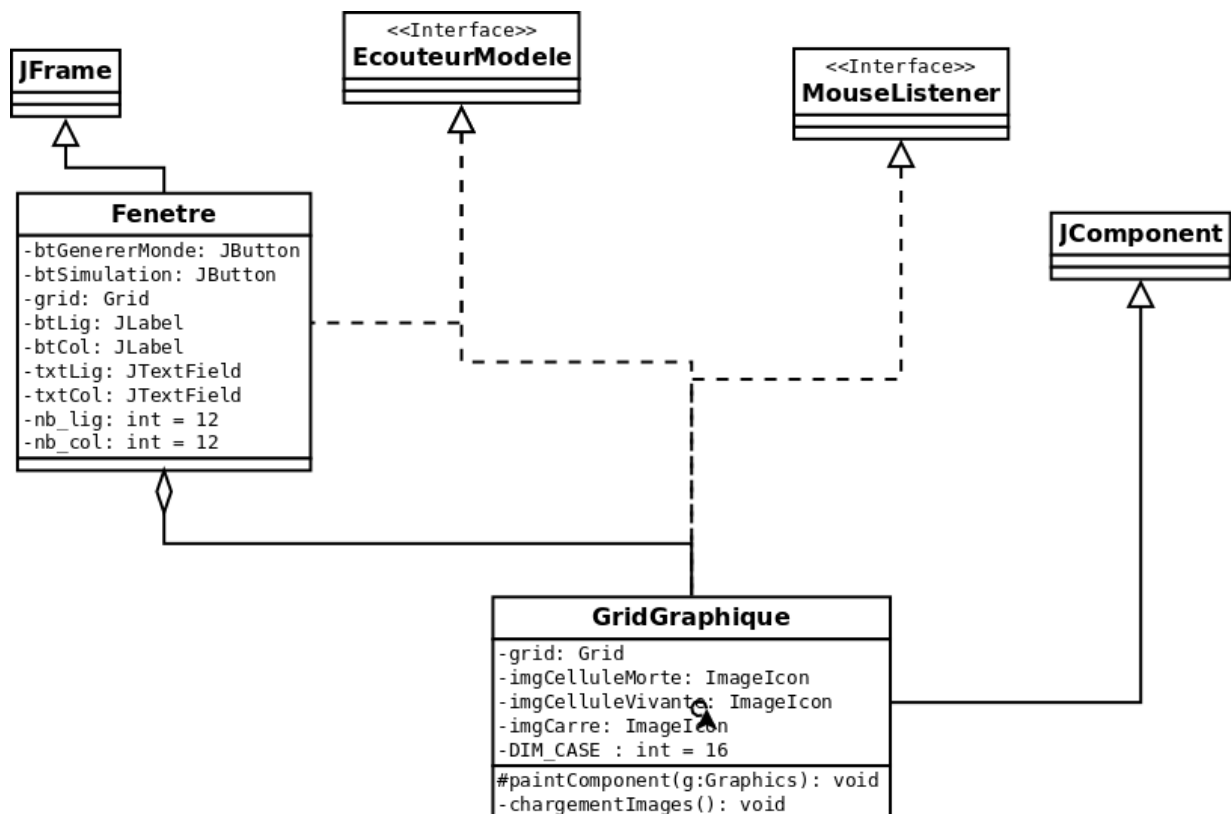


Diagramme de classe Graphique

### 3.4 Manuel d'utilisation

Le jeu de la vie a pour objectif principal de faire évoluer les cellules d'une génération à une autre.

#### 3.4.1 Jeu graphique

Pour lancer le jeu en version graphique rendez-vous dans la racine du projet, ouvrez un terminale et taper les commandes suivantes :

- ant compile
- ant run

Une fois le jeu lancé vous tomberez sur la fenêtre principale (voir figure 1)

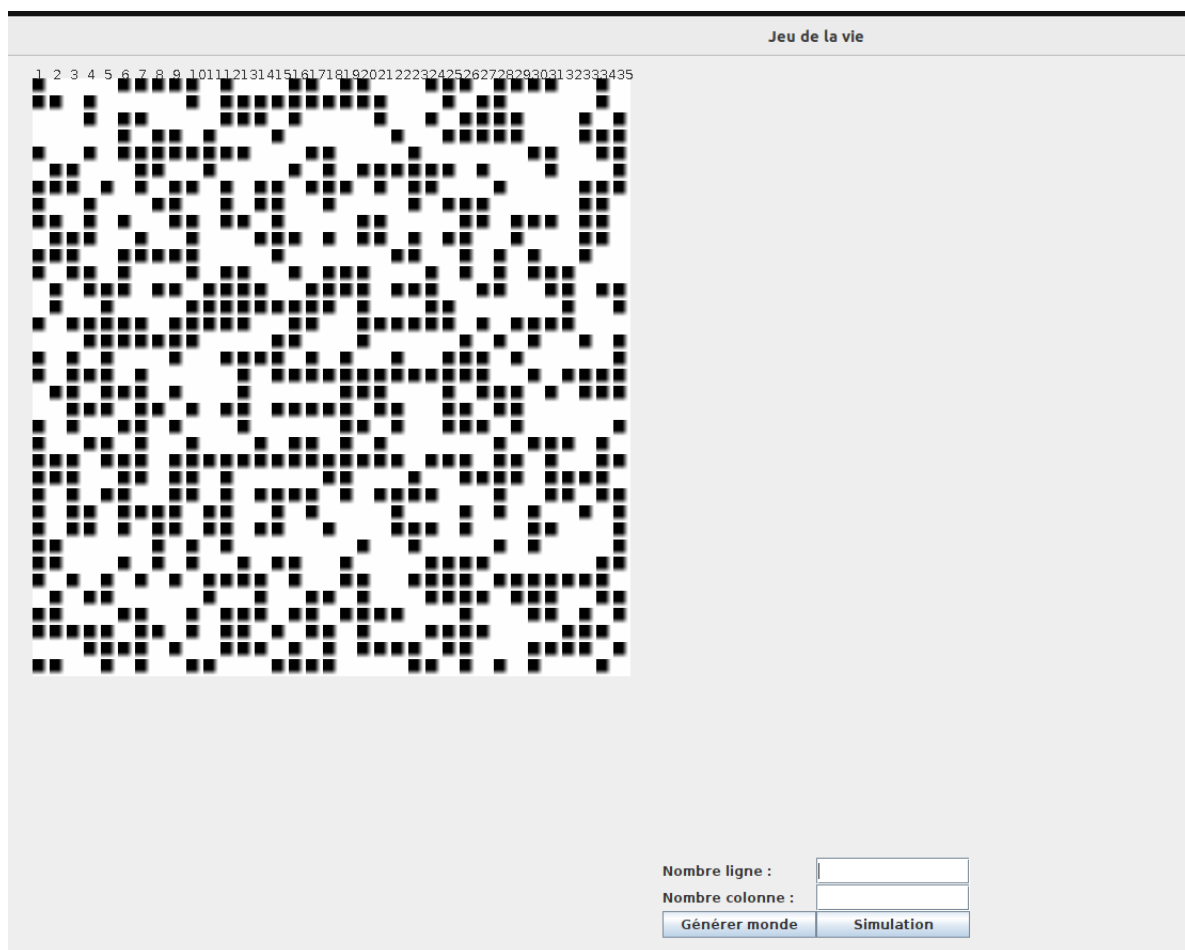


Figure 1 :Fenêtre principale du jeu

Sur la page principale vous aurez à saisir la taille de votre grille, après avoir saisi la taille de la grille vous avez deux boutons (voir figure 2) :

- **Générer monde** : permet de générer un monde aléatoire .
- **Simulation** : Permet de faire la simulation de la grille .

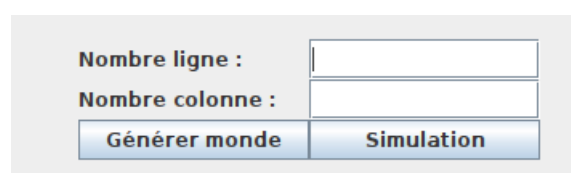


Figure 2 :Les touches d'interaction

durant la simulation graphique, en arrière plan vous pouvez observer une simulation dans le terminale (voir figure : 3).

[illegible]

Figure 3 :Affichage de la grille en console

### 3.5 Difficulté rencontrée : Algorithme HashLife

L'algorithme hashlife introduit la notion de macro-cellule. Une macro-cellule est une portion de la grille de taille  $2^n.2^n$

Cette portion de la grille peut-être représenter par une structure de donnée quadtree. Cet algorithme permet d'accélérer la résolution une fois implémenté dans le jeu de la vie. Après plusieurs recherche sur cet algorithme notamment sur : [site de drdobbs](#), [Un bloc expliquant hashlife](#), Nous avons essayé de l'implémenter dans notre jeu de la vie naïve mais en vain. Dans le package hashlife se trouve la classe Noeud permettant de représenter les macro-cellules de la grille.

## 4 conclusion

La réalisation de ce projet a été une occasion pour nous membre de ce groupe d'apprendre encore plus le langage de programmation Java. Pendant ce projet, nous avons pu appliquer ce que nous avons appris dans les CM, les TP de ce semestre et ceux du semestre passé à savoir : la programmation orientée objet, la conception d'applications et d'autre savoir-faire.

d'un point de vue humain, nous avons appris à mieux travailler en équipe, à communiquer, à bien organiser un travail d'équipe et le plus important à combler nos lacunes. Nous avons également appris à travailler à distance en utilisant les outils de travail de groupe comme SVN.

Dans ce projet, nous avons pour objectif premier de réaliser le jeu de la vie naïve en implémentant toutes les règles du jeu dans une application JAVA. Nous avons donc pu coder

le moteur du jeu respectant les règles du jeu, nous avons aussi réussi par faire une interface graphique afin que le jeu soit jouable graphiquement.

#### 4.1 Améliorations possibles

Nous pouvons améliorer cette application à plusieurs niveaux ; tout d'abord, implémenter l'algorithme hashlife pour accélérer la résolution puis étendre ce système aux automates cellulaires neuraux.

ensuite, donner la possibilité de sauvegarder une grille de génération.

L'interface graphique peut être améliorée en affichant par section pour une très grande grille de génération.