



Projet JAVA Semestre 4

JEU DE LA VIE

Mannivanin Cécile
Farah Amina
Capdevila Martin
Bazire Fabrice
Groupe I1



TABLE DES MATIÈRES	2
LE PROJET	3
MISE EN PLACE	3
CHOIX DES ALGORITHMES	3
DIFFICULTÉS	4
EXPÉRIMENTATIONS	5
RÉPARTITION DU TRAVAIL	9
DIAGRAMME DE CLASSE	10
LIEN GITHUB	11

Le projet

Mise en place

Lors de la découverte du sujet, aucun des membres du groupe ne connaissait le jeu de la vie. Nous avons donc effectué des recherches pour bien en comprendre les règles avant de commencer à coder. Cette phase nous a permis d'élaborer un plan des différentes étapes à suivre lors de la conception de l'algorithme. Nous avons essayé de respecter ce plan et le planning associé du mieux que nous pouvions malgré certaines difficultés bloquant notre avancé.

La prise en main d'IntelliJ a été très rapide étant donné sa ressemblance avec Eclipse. Cependant, nous avons eu beaucoup plus de mal à utiliser Git convenablement.

Choix des algorithmes

Comme imposé dans le sujet, nous avons codé le jeu de la vie en utilisant des listes que nous avons codées nous-même. Nous avons fait le choix de faire une première classe contenant les méthodes liées aux listes et une autre classe contenant les méthodes propres aux respect des règles et au fonctionnement du jeu lui-même.

Nous avons essayé de coder le plus simplement possible mais nous sommes conscients que certaines méthodes ne sont pas optimisées.

L'exemple des méthodes parcourant les listes semble le plus pertinent. Ces méthodes sont coûteuses en ressources, comme la méthode qui permet aux cellules mortes de vivre. En effet celle-ci doit vérifier la présence d'éléments dans plusieurs listes et a de nombreuses reprises. Cependant toutes ces interventions nous paraissaient nécessaires.

Difficultés

Tout au long du projet, nous avons fait face à des difficultés. Nous avons surmonté un grand nombre d'entre elle en échangeant au sein du groupe mais également en discutant de nos problèmes mutuels avec nos camarades des autres groupes.

Nous avons notamment pris du retard au moment du codage de la généricité des listes.

Malgré notre investissement, il reste des points du code non fonctionnel :

- Le changement des règles du jeu, on peut changer les règles du jeu dans un fichier lif, mais les nouvelles ne fonctionnent pas très bien
- La gestion du max passé en paramètre avec les options `-c` et `-w` à l'exécution du projet
- La période ainsi que la taille de la queue ne fonctionnent pas tout le temps, on pense cependant que la taille de la queue est bonne mais pas la période, et comme la taille de la queue est calculée par rapport à la période, le résultat est faussé.
- Un affichage de « null » non désirés et intempestif
- La gestion des types vaisseaux

Expérimentations

Voici les résultats des différentes options d'exécutions possibles :

- -h : affichage de toutes les options possibles

```
macbook-pro-de-fabrice:src fabrice$ java Principale -h
options possibles :
-name : affiche le nom des créateurs de cette appli
-s : effectue une simulation du jeu (nécessite une durée et un fichier.lif en paramètre derrière -s)
-c : calcule le type d'évolution du jeu (nécessite une durée max et un fichier.lif en paramètre derrière -c)
-w : calcule le type d'évolution de plusieurs jeux dans un dossier (nécessite une durée max et le nom du dossier en paramètre derrière -w)
macbook-pro-de-fabrice:src fabrice$
```

- -name : affiche les noms des membres du groupes

```
macbook-pro-de-fabrice:src fabrice$ java Principale -name
Cécile, Amina, Martin et Fabrice
```

- -s temps fichier.lif : affiche pendant un temps donné les générations générés par le fichier.lif donné

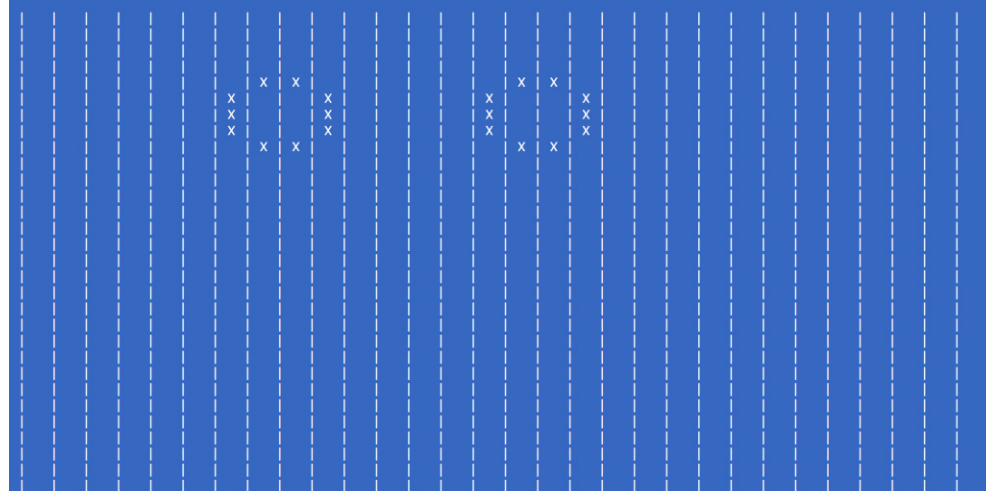


```
a.lif
1 #Life 1.06
2
3 #D regle de base
4
5 #N
6
7 #P 2 2
8 ...**.....**.....
9 ..*..*.....*..*.....
10 ..*..*.....*..*.....
11 ..*..*.....*..*.....
12 ...**.....**.....
13 |
```

```

macbook-pro-de-fabrice:~ fabrice$ cd documents/github/projet-info-s4/src
macbook-pro-de-fabrice:src fabrice$ java Principale -s 3 a.lif
La durée est de : 3
Le nom du fichier est : a.lif
nullnullnullnullnullnullnullnull13
nullnullnullnullnull

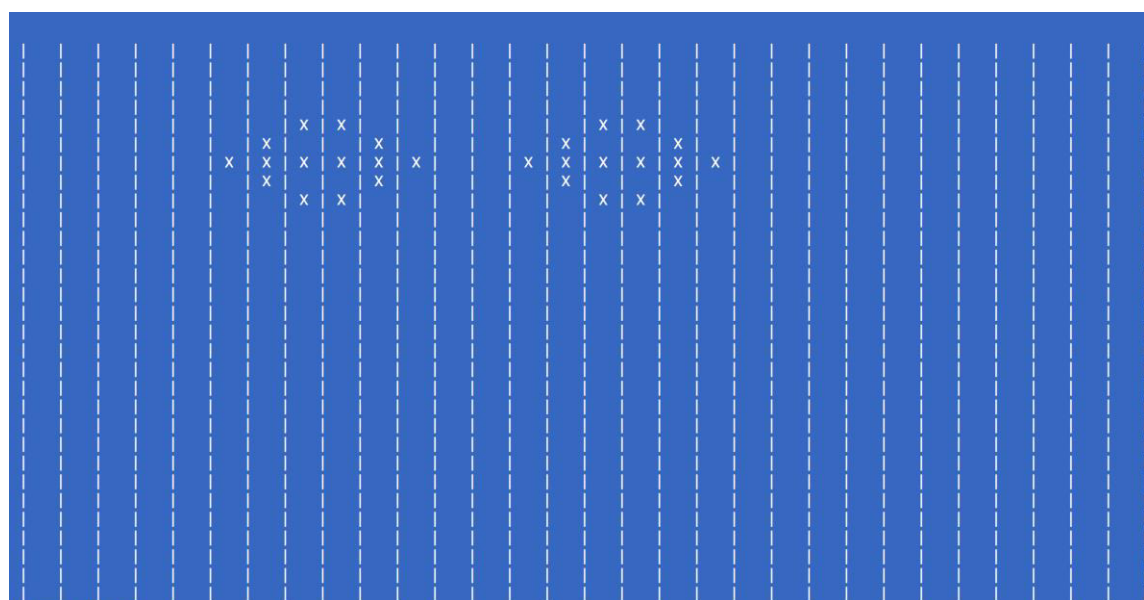
```



```

generation n°0
nullnullnull13
nullnullnullnullnullnull

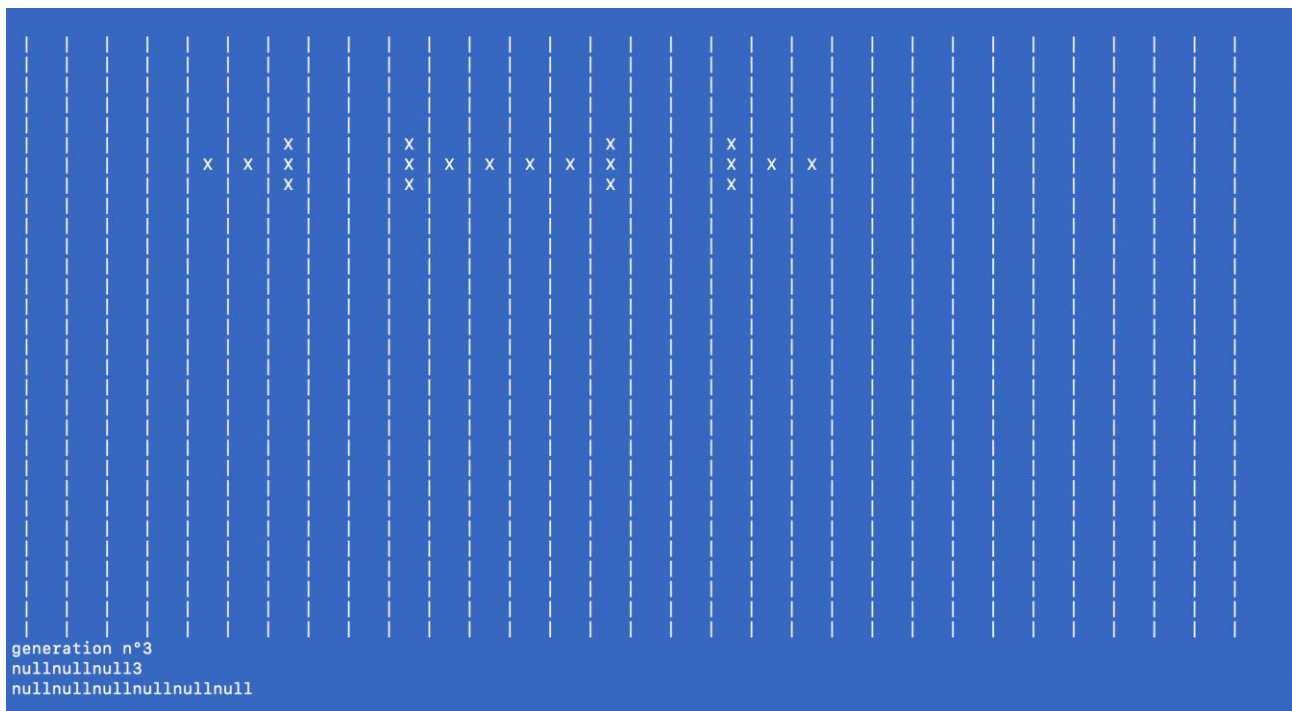
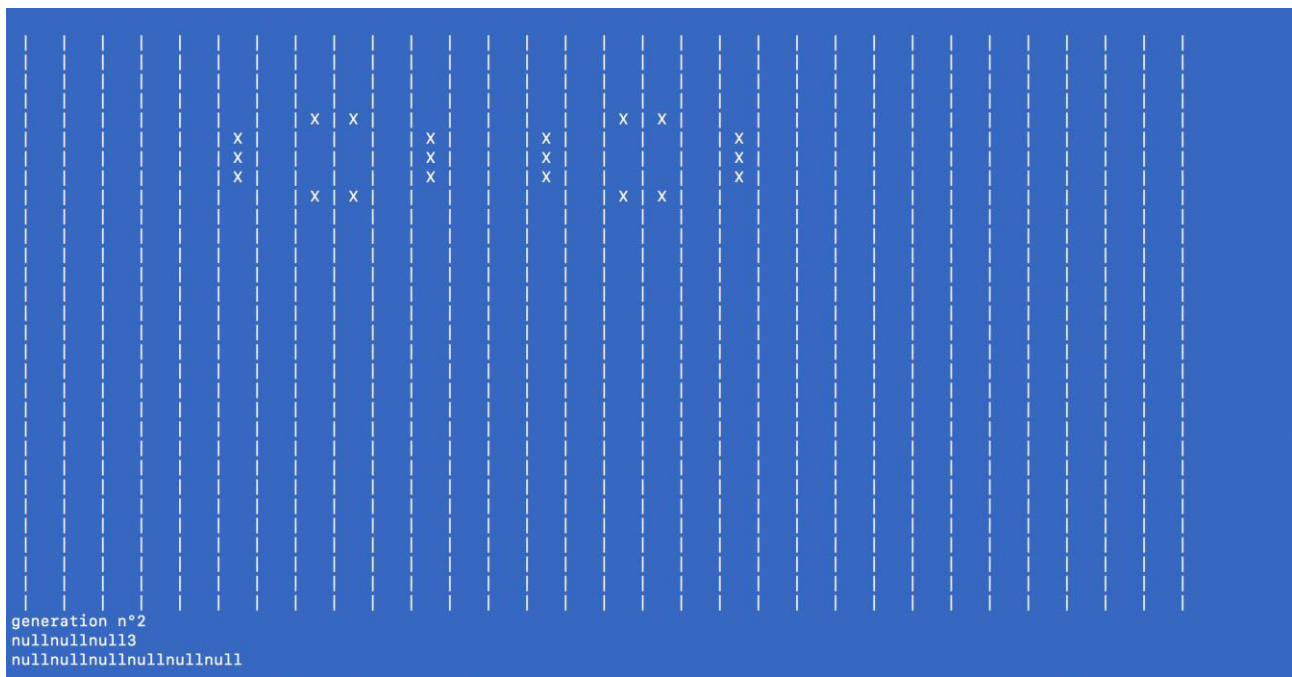
```



```

generation n°1
nullnullnull13
nullnullnullnullnullnull

```



```

generation n°4
macbook-pro-de-fabrice:src fabrice$

```

- -c max fichier.lif : affiche le comportement de la configuration donné dans le fichier.lif si le temps max est suffisant pour la trouver

```

a.lif
1 #Life 1.06
2
3 #D regle de base
4
5 #N
6
7 #P 2 2
8 .....**.....**.....
9 ..*.*.*.....*.*.....
10 ..*.*.*.....*.*.....
11 ..*.*.*.....*.*.....
12 .....**.....**.....
13

```

```

macbook-pro-de-fabrice:src fabrice$ java Principale -c 15 a.lif
nullnullnullnullnullnullnullnull13
nullnullnullnullnullnullnull13
nullnullnullnullnullnullnull13
nullnullnullnullnullnullnull13
nullnullnullnullnullnullnull13
nullnullnullnullnullnullnull13
nullnullnullnullnullnullnull13
nullnullnullnullnullnullnull13
nullnullnullnullnullnullnull13
nullnullnull
Le comportement est : oscillateur
La période est de : 15
La taille de la queue est : 1
macbook-pro-de-fabrice:src fabrice$

```


Répartition du travail

Le travail a été naturellement réparti à mesure que le projet avançait. Nous avons favorisé un travail collectif pour élaborer la majorité des algorithmes. La réflexion commune nous a permis de relever et corriger les erreurs de chacun pendant l'élaboration des algorithmes. Tous les morceaux de codes écrits en dehors de ces réunions ont été commentés (nous avons retiré ces commentaires pour la version finale car ils surchargeaient le code et étaient écrits de manière informelle) ou expliqué de vive voix par l'auteur. Cependant, chaque membre du groupe a traité plus particulièrement une partie du sujet.

Amina a beaucoup travaillé sur la classe Liste. Elle a fourni les principales méthodes de cette classe, notamment pour la généricité des listes.

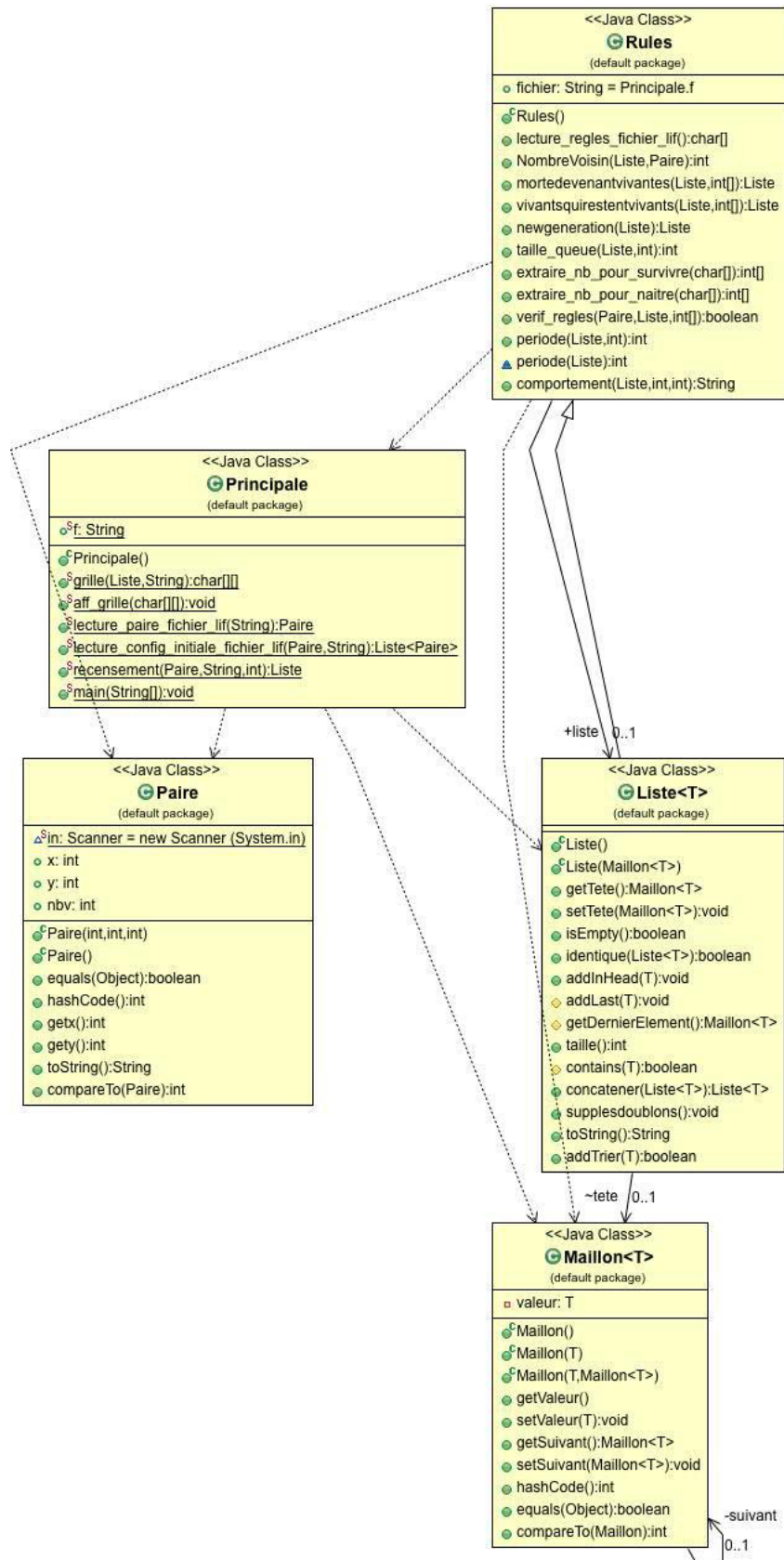
Martin s'est concentré sur l'architecture des algorithmes en travaillant principalement sur papier en langage algorithmique.

C'est principalement Fabrice qui a traduit et arrangé ces algorithmes en Java. Il a également traité la partie de lecture des fichiers lif et enfin il a mis en place la gestion de versions avec GitHub.

La java doc a été partagée entre Cécile et Martin.

La majorité des modifications et débogages ont été réalisées avec l'ensemble du groupe, lors de concertations.

Diagramme de Classe



Lien GitHub

Voici le lien du projet avec toutes ces versions : <https://github.com/fabrice-bazire/projet-info-s4>