

Université de Cergy-Pontoise

RAPPORT

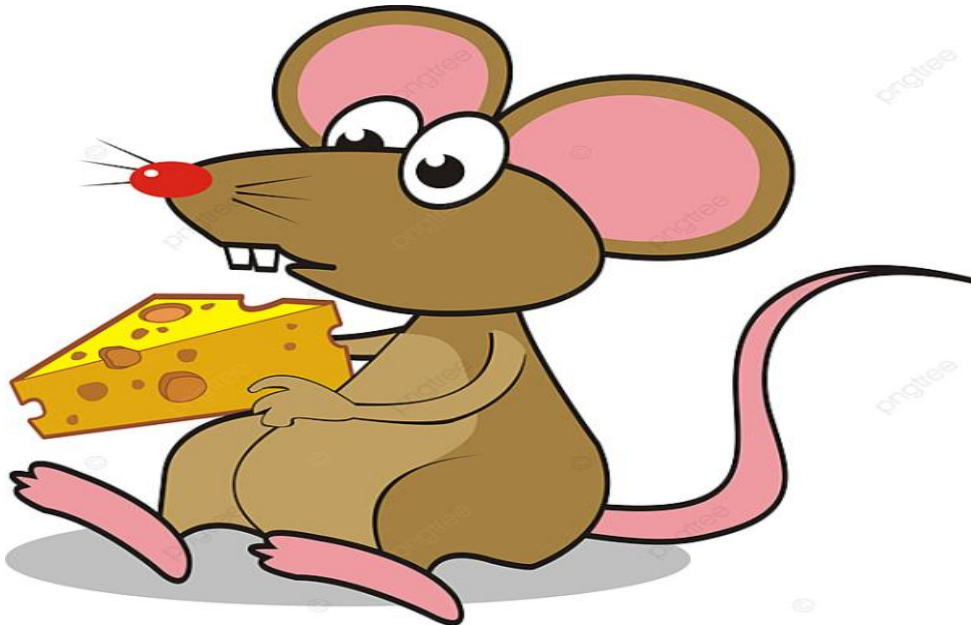
pour le projet Génie Logiciel
Licence d'Informatique deuxième année

sur le sujet

Souris

rédigé par

MALEK Feriel / ROUAS Lila



Avril 2022

Table des matières

1	Introduction	3
1.1	Contexte du projet	3
1.2	Objectif du projet	3
1.3	Organisation du rapport	3
2	Spécification du projet	4
2.1	Notions de base et contraintes du projet	4
2.1.1	Fonctionnement général du logiciel	4
2.1.2	Les sources de nourriture	4
2.1.3	Les obstacles	4
2.2	Caractéristique et comportements des souris	4
2.2.1	Contraintes :	4
2.3	Fonctionnalités attendues du projet	4
3	Conception et réalisation du projet	6
3.1	Architecture globale du logiciel	6
3.2	Conception des classes de données	7
3.3	Conception des traitements (processus)	7
3.3.1	La Grille	7
3.3.2	Les Souris	7
3.3.3	Moteur	8
3.3.4	Statistique	9
3.4	Conception de l'IHM graphique	10
3.5	Les aspects techniques	11
4	Manuel utilisateur	12
4.1	Introduction :	12
4.2	La fenêtre principale :	13
5	Déroulement du projet	14
5.1	Réalisation du projet par étapes	14
5.2	Répartition des tâches entre membres de l'équipe	15
5.3	Synchronisation du travail	15
6	Conclusion et perspectives	16
6.1	Résumé du travail réalisé	16
6.2	Améliorations possibles du projet	17

Table des figures

1	l'architecture globale	6
2	Diagramme UML des classes de donnée	7
3	Moteur	8
4	Statistiques	9
5	Conception de l'IHM	10
6	Aspects techniques	11
7	Accueil et paramétrages de la simulation	12
8	La fenêtre principale	13
9	Encore de la simulation	13
10	Les tâches entre les membres	15
11	Resumé du travail réalisé	16
12	Améliorations possibles	17

Remerciements

Nos remerciements s'adressent particulièrement à Monsieur LIU Tianxiao, pour son encadrement de qualité, sa motivation professionnelle, ses conseils et critiques constructives, ses corrections, sa gentillesse et sa patience ainsi pour le temps qu'il a consacré à la réalisation de ce projet.

<https://www.overleaf.com/project/62484dcc30eccf2493b753b6>

1 Introduction

Ce projet est de créer un environnement de grille en IHM graphiques pour évaluer le comportement des souris précisément leur mouvement, nourriture, caractères. Ce projet nous motive car on trouve ça amusant et très intéressant de créer une simulation et de mettre en œuvre des solutions en intelligence artificielle et que l'utilisateur peut contrôler le jeu comme il veut et passer de temps en dessous.

1.1 Contexte du projet

Dans le cadre de notre deuxième année de licence informatique nous devons réaliser un projet de programmation en Java pour le module de génie logiciel. Le projet des souris est intéressant pour faire un premier pas dans l'intelligence artificielle, gérez-les communications des souris et leurs comportements.

1.2 Objectif du projet

Nous devons créer une simulation où des souris s'évaluent sur une grille qui se communique et se nourrissent. Pour faire ça nous allons utiliser l'environnement de développement Eclipse et la plateforme java.

1.3 Organisation du rapport

Notre rapport présentera tout d'abord les spécifications du projet ; nous parlerons ensuite de la conception et de la réalisation du projet de manière plus détaillée. Nous abordera le déroulement du projet ainsi que la répartition des tâches et ensuite le manuel d'utilisateur. Finalement nous donnerons le bilan de notre projet. Ce rapport a été rédigé avec Latex sur le service web Overleaf.

2 Spécification du projet

2.1 Notions de base et contraintes du projet

2.1.1 Fonctionnement général du logiciel

La grille où les souris s'évaluent comporté des sources de nourriture et des obstacles ; l'utilisateur peut avoir un contrôle sur ces éléments

2.1.2 Les sources de nourriture

Les pôles de nourritures apparaissent aléatoirement et spontanément au cours du temps dans l'environnement des souris et on peut choisir le nombre avant de commencer la partie. Chaque source est limitée, ils disparaissent quand ils sont entièrement consommé, et une souris mange un fromage quand elle se retrouve sur la même case que la nourriture et peut consommer plusieurs.

2.1.3 Les obstacles

Ils seront positionnés aléatoirement sur la grille en début de partie et on peut choisir le nombre aussi, dès qu'une souris coince dans l'obstacle à un moment elle meurt.

2.2 Caractéristique et comportements des souris

Caractéristique d'une souris : Une souris a des unités de vie, un certain nombre par souris ; une souris perd une unité de vie si elle fait un tour sans se nourrir. (Un tour est par exemple tant de déplacements successifs pour les souris) ou si elle rentre dans l'obstacle, elle peut donner naissance à une autre souris quand elle se croise avec une autre souris. Elles peuvent se déplacer dans quatre directions (avant, gauche, droite, arrière).

Déplacement : Une souris ne peut pas être bloquer sur la grille, elle pourra s'orienter dans plusieurs direction et donc aller en arrière. Les souris peuvent se déplacer d'une case ou rester sur place, selon la case elle pourra manger ou alors communiquer. Si deux souris se retrouve dans la même case elles pourront alors communiquer et donner naissance a une nouvelle souris.

Communications : Si deux souris se rencontrent face à face ou sur la même case elles peuvent se communiquer et échanger des informations concernant l'emplacement connu de nourriture.

Comportement : on distingue deux comportements : les souris mutuelles et celles égoïstes. Pour la réception des informations on distinguera les souris réceptives et celles nihilistes.

Accouplement : lorsque deux souris se rencontrent ils peuvent donner naissance à une nouvelle souris.

2.2.1 Contraintes :

Ce projet est sous la forme d'une simulation, en effet l'utilisateur doit pouvoir intervenir lors du déroulement de ce jeu et ses actions modifieront l'aventure de nos souris. En effet le but n'est pas que l'application se déroule seule ou se termine instantanément en affichant les statistiques de la partie sans que nous puissions interagir, chaque partie sera différente et les paramètres de la partie pourront être réglés par l'utilisateur afin que la simulation ne soit pas répétitive.

2.3 Fonctionnalités attendues du projet

Fonctionnalités : L'utilisateur du logiciel doit pouvoir :

- Initialiser les paramètres du jeu (nombre de souris, d'obstacles, de sources de nourriture, dimension de la grille)
- Démarrer la simulation, Faire une pause à tout moment et reprendre le mouvement de souris.

- Avoir un bouton "revenir à l'accueil" qui permet à l'utilisateur de retourner sur la page d'accueil pour choisir de nouveaux paramètres.
- Avoir un bouton "help" qui peut aider l'utilisateur en cas de non-compréhension de la simulation.
- Consultez les statistiques de la partie (nombre de souris en vie, nombre de souris mortes).

3 Conception et réalisation du projet

3.1 Architecture globale du logiciel

Dans la figure ci-dessus, on peut voir l'architecture globale de notre logiciel.

Pour commencer la simulation vous devez choisir la dimension de la grille que vous souhaitez, ainsi que d'autres paramètres comme le nombre de souris, nombre de nourriture et nombre d'obstacles.

- Ensuite vous cliquez sur le bouton Start qui va vous mener vers le jeu là où y aura la grille qui contient des souris qui circulent et se communiquent entre eux, et les autres éléments (nourritures et obstacles).

- Une fois qu'une souris rencontre une autre souris sur la même position ils donnent naissance à une nouvelle souris qui sera générée sur la grille.

- Et que quand une souris est dans la même position qu'un fromage elle le mange et quand elle sera dans la même position qu'un obstacle elle rentre dans le piège et elle meurt.

- Vous pouvez également voir les statistiques de la partie lors de la simulation pour consulter le nombre de souris encore vivantes.

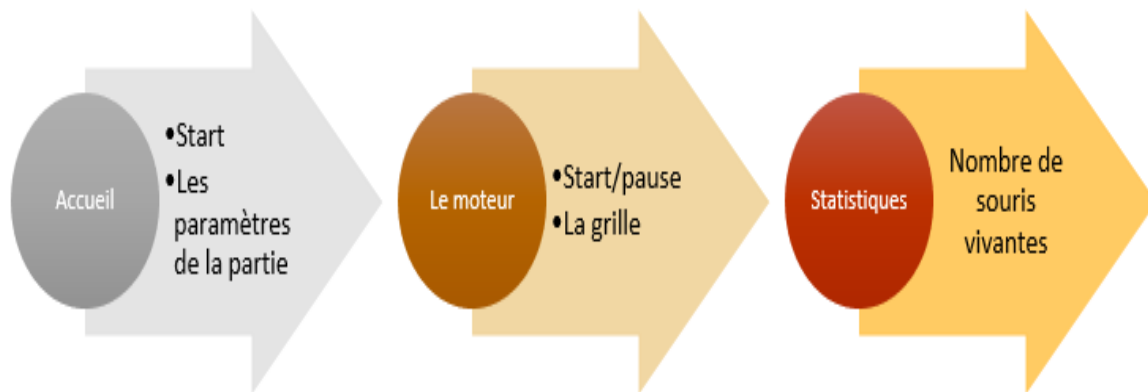


FIGURE 1 – l'architecture globale

3.2 Conception des classes de données

ici on a le diagramme UML de classes des éléments essentiels de notre projet

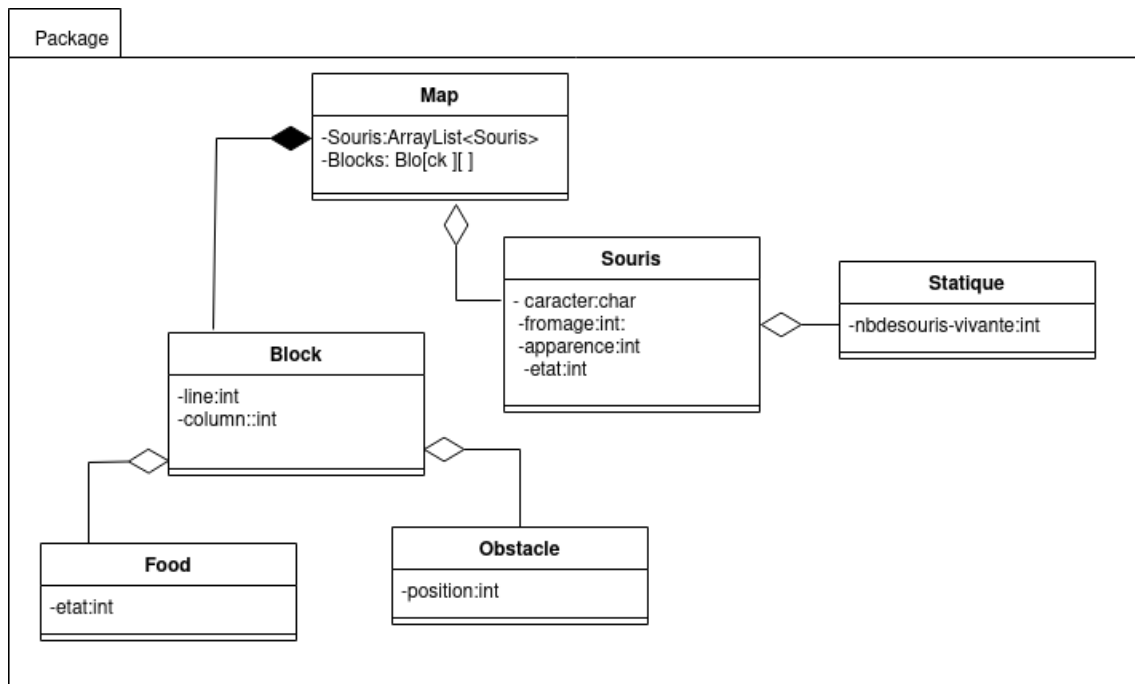


FIGURE 2 – Diagramme UML des classes de donnée

3.3 Conception des traitements (processus)

3.3.1 La Grille

- Nous avons d'abord fabriquer les éléments de la grille, les obstacles ainsi que les sources de nourriture et l'herbe ou les souris peuvent se déplacer librement.
- Pour créer la grille nous nous sommes servi de la classe bloc et map dans lequel nous avons placé aléatoirement les obstacles et la nourriture.

3.3.2 Les Souris

- Nous avons créé d'abord une classe de souri simple comprenant son caractère. Nous avons ensuite prévu d'avoir prévu leurs déplacements, leur vision et leur comportement.
- Pour leur comportement nous avons géré ça dans la classe "MobileElementManager" ainsi que leurs déplacements.
- Pour les couleurs des souris on a utilisée la classe PaintStrategy afin de reconnaitre chaque caractère, et que quand une souris mange un fromage elle grossis.
- Pour l'accouplement ça se passe qu'entre (A et B) ou entre(C et D).
- Les souris elles se déplacent dans 4 directions (en haut, en bas, a gauche et a droite).
- On a spécifié le sexe des souris par exemple les femelles elles ont un point blanc dans le dos.
- Ainsi que les souris qu'on elles mange un fromage elle prend du poids.

3.3.3 Moteur

La classe "MobileElementManager" gère le moteur de jeu, il consiste à parcourir les ArrayList et associe à chaque objet une action. Elle fonctionne sous forme de boucle, voire schéma ci-dessous.

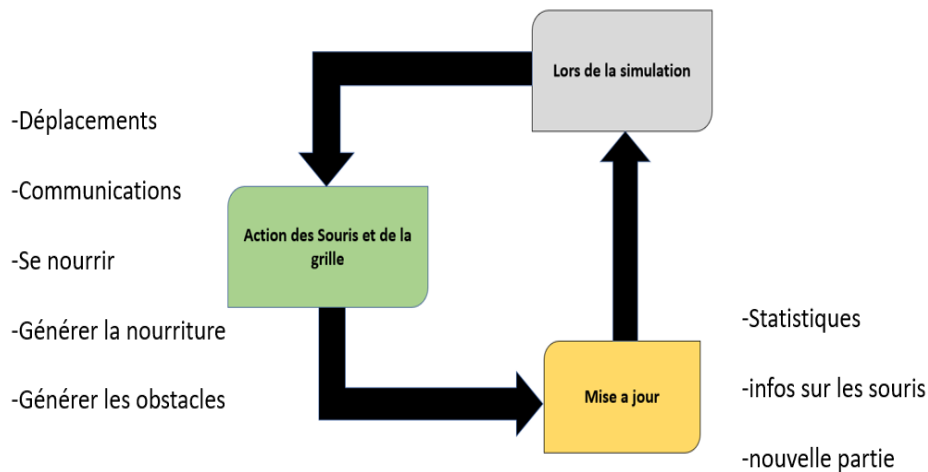


FIGURE 3 – Moteur

- Un tour de jeu représente, l'action des souris (déplacements, accouplement), l'action de la grille et la mise à jour des informations des souris et de la grille.
- Au début d'un tour de simulation l'utilisateur peut effectuer des actions comme spécifier la quantité de nourriture, la quantité des obstacles et les nombres de souris.
- Les sources de nourriture apparaissent dès le lancement de la simulation, au bout de ce temps, une nouvelle génération de nourritures aura lieu. Une source de nourriture est supprimée lorsque la position de la souris et la position de nourriture sont égales (décrémenter à chaque fois qu'une souris mange).
- Les obstacles sont fixes pendant toute la partie.

3.3.4 Statistique

Dans la classe ChartManager on a utilisé un Hmap pour la clé où on a pris un caractère (caractère de la souris) et la valeur on a pris un entier (nombre de souris vivantes). Dans la classe MobileElementManager on incrémente le nombre de souris en vie en fonction de leurs caractères(count Type) lorsqu'une souris est générée et on décrémente le nombre de souris en vie en fonction(RemoveCountType) lorsqu'une souris est morte.

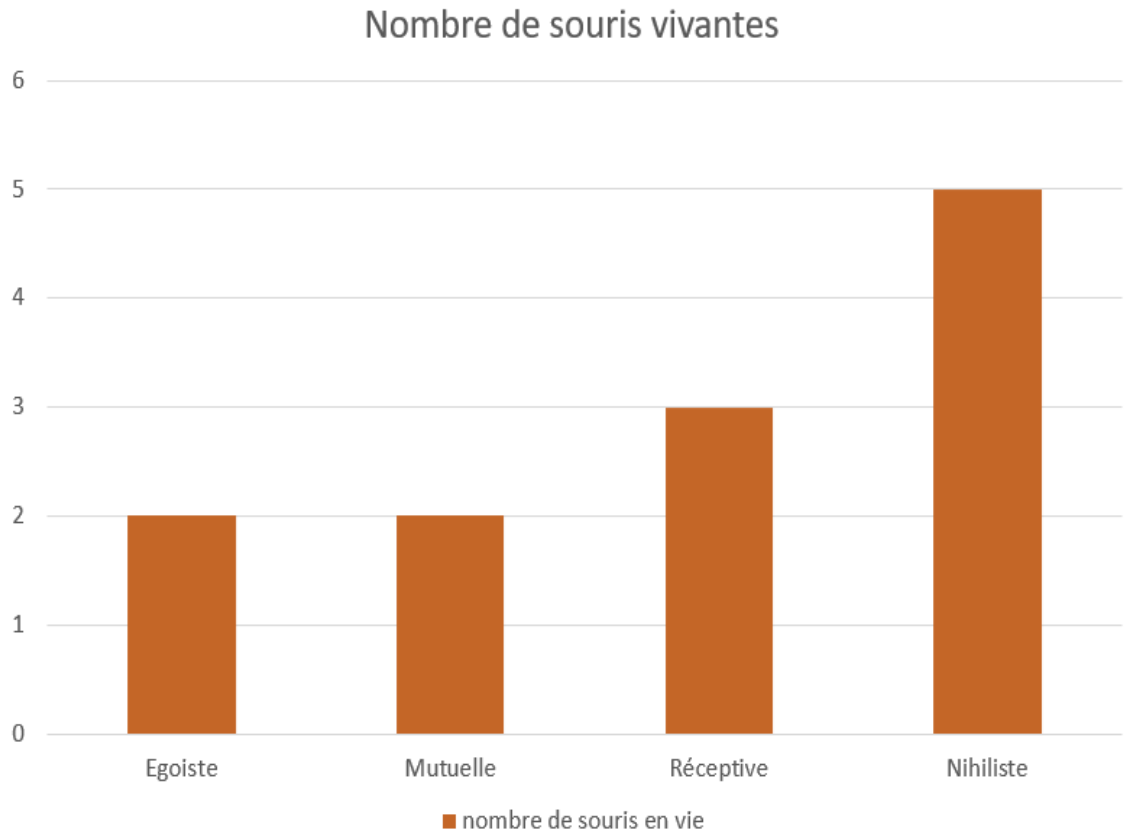


FIGURE 4 – Statistiques

3.4 Conception de l'IHM graphique

On lançant la simulation on aura l'accueil ou l'utilisateur choisira les paramètres de la partie, ont cliquant sur start on aura le MainGui, On a MobileElementManager qui met à jour des événements avec la méthode nextround, qui envoie aussi des informations à GameBuilder qui est lui-même initialise les informations avec buildinitmobile et le MobileElementManager envoie ainsi des informations pour Statistiques comme le nombre de souris encore vivantes qui s'incrémente une fois ya des souris vivantes et ça décrémente une fois les souris meurent et y a la classe ChartManager qui met à jour les statistiques et qui les affichent sur le MainGui.

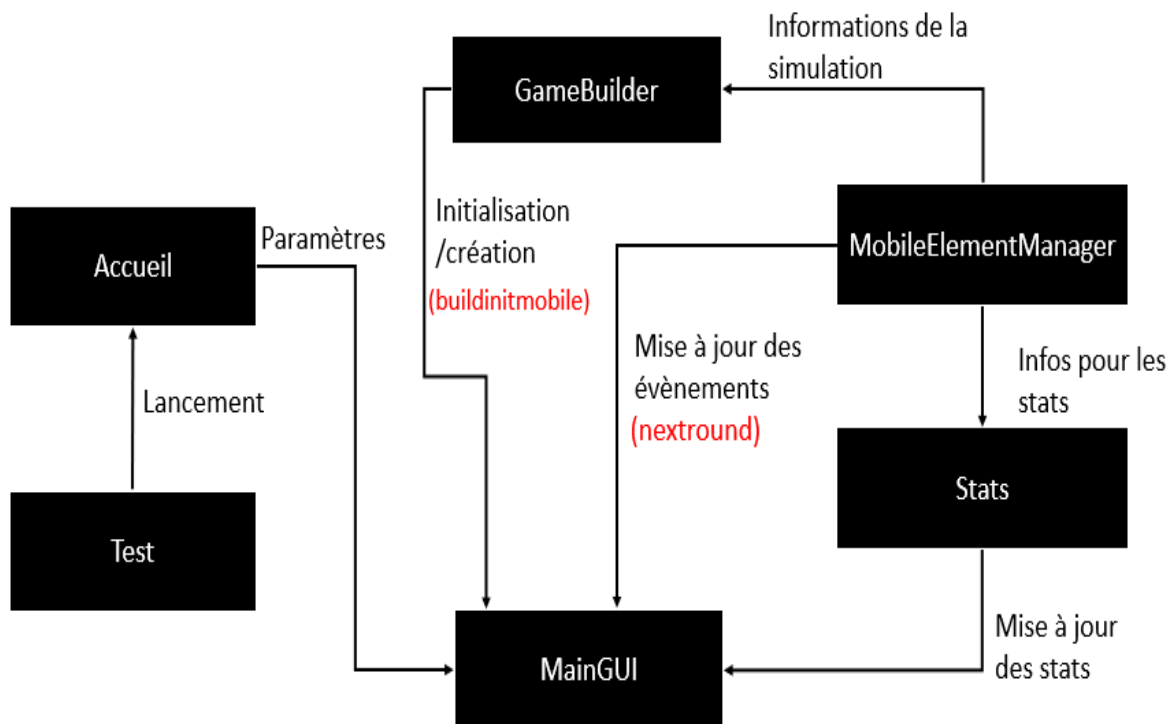


FIGURE 5 – Conception de l'IHM

3.5 Les aspects techniques

Dans notre projet on a utilisé beaucoup de notions y compris des notions mathématiques. Pour gérer les déplacements, la naissance et l'apparence, on a employé des probabilités :

D'abord pour le déplacement avant que la souris mange du fromage elle a une probabilité de $\frac{1}{4}$ d'aller vers une des quatre positions, idem pour l'apparence.

Une fois la souris a mangé un fromage elle a tendance avoir une probabilité de $\frac{1}{2}$ pour aller vers une position au bien d'avoir une apparence moins que les autres. Dans l'accouplement pour donner naissance à une autre souris, une fois elle a mangé un fromage la nouvelle souris elle a une chance sur 2 pour avoir un des caractères des parents.

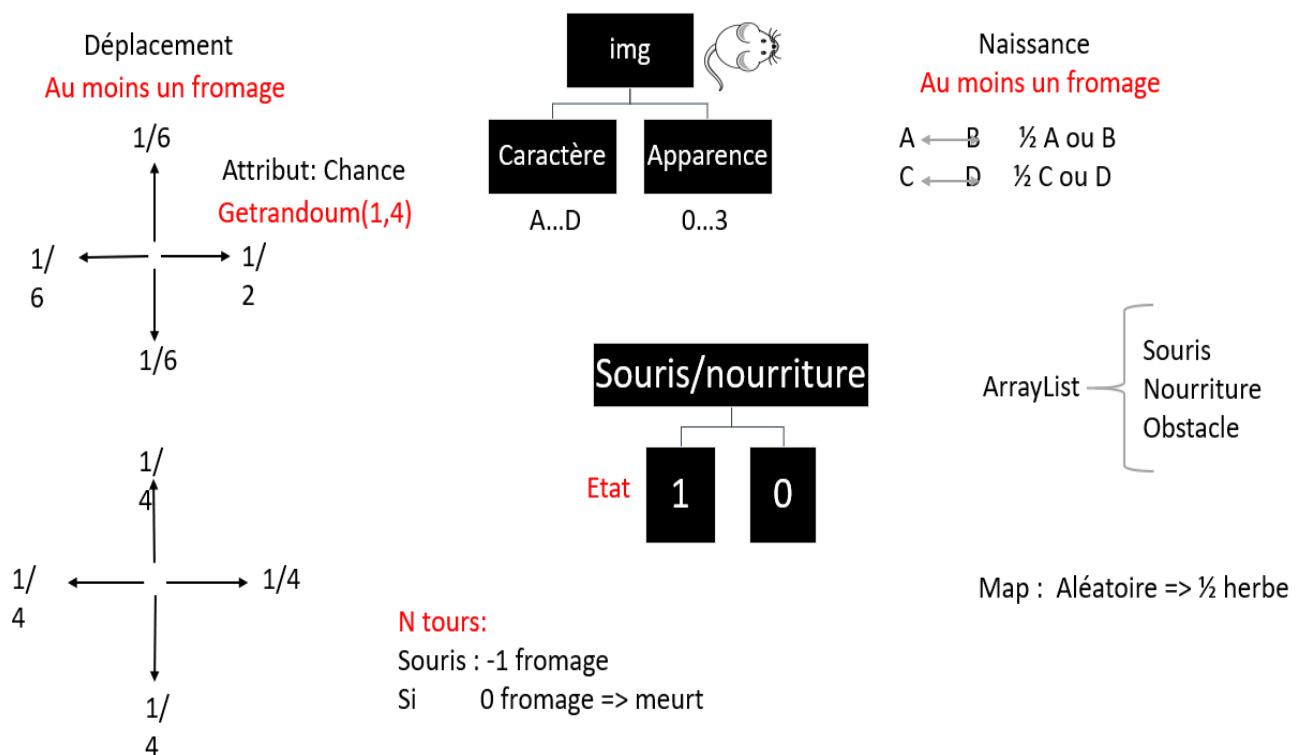


FIGURE 6 – Aspects techniques

4 Manuel utilisateur

4.1 Introduction :

Ceci est la première fenêtre quand on lance la simulation, y a le bouton start pour commencer la partie, le bouton help pour aider l'utilisateur en cas de difficultés à propos de la simulation, et on a aussi quatres boutons pour choisir les paramètres de la partie comme le nombre de souris, nourritures, obstacles ainsi que la dimension de la grille.

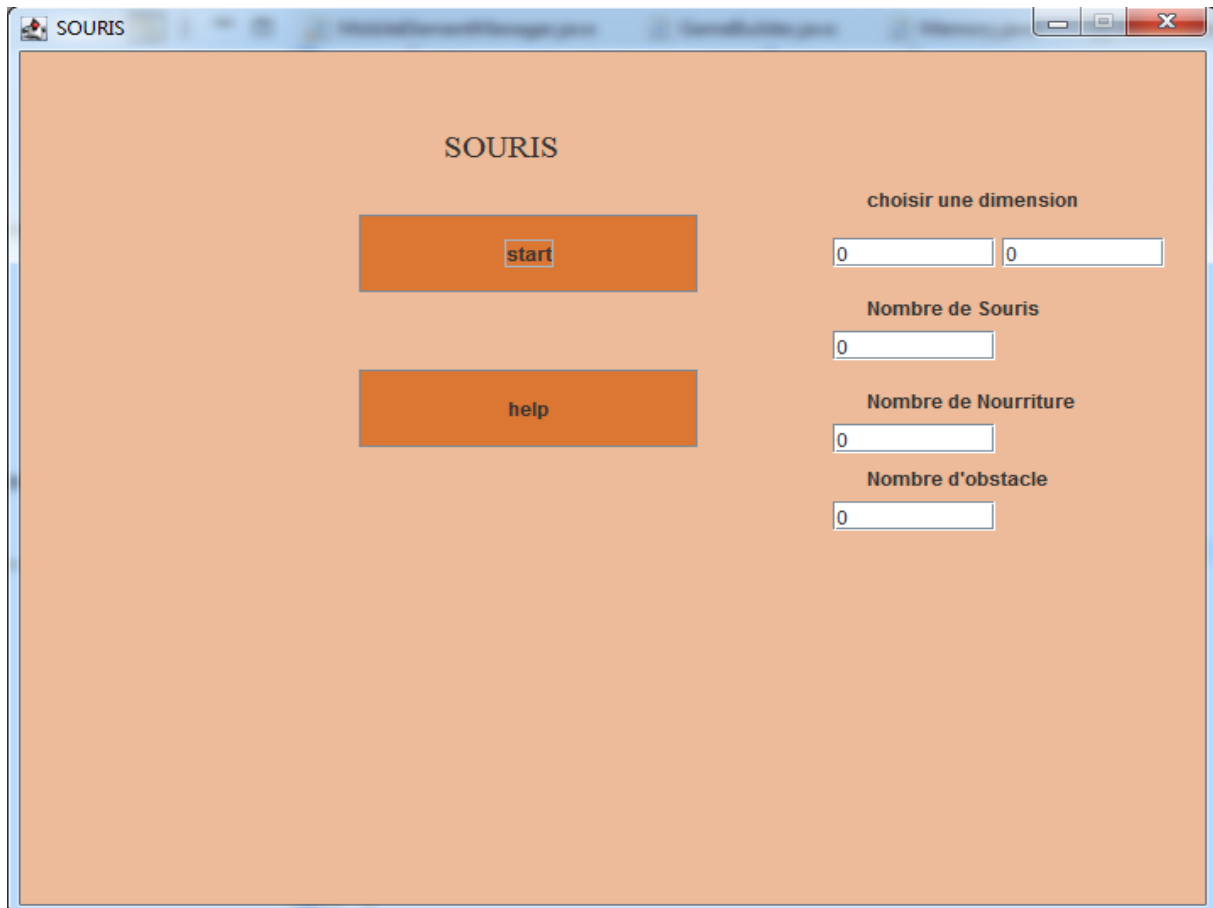


FIGURE 7 – Accueil et paramétrages de la simulation

4.2 La fenêtre principale :

Vue de notre simulation sur (la figure 4) avant de mettre start sur le bouton en bas et on peut mettre pause aussi à tout moment.

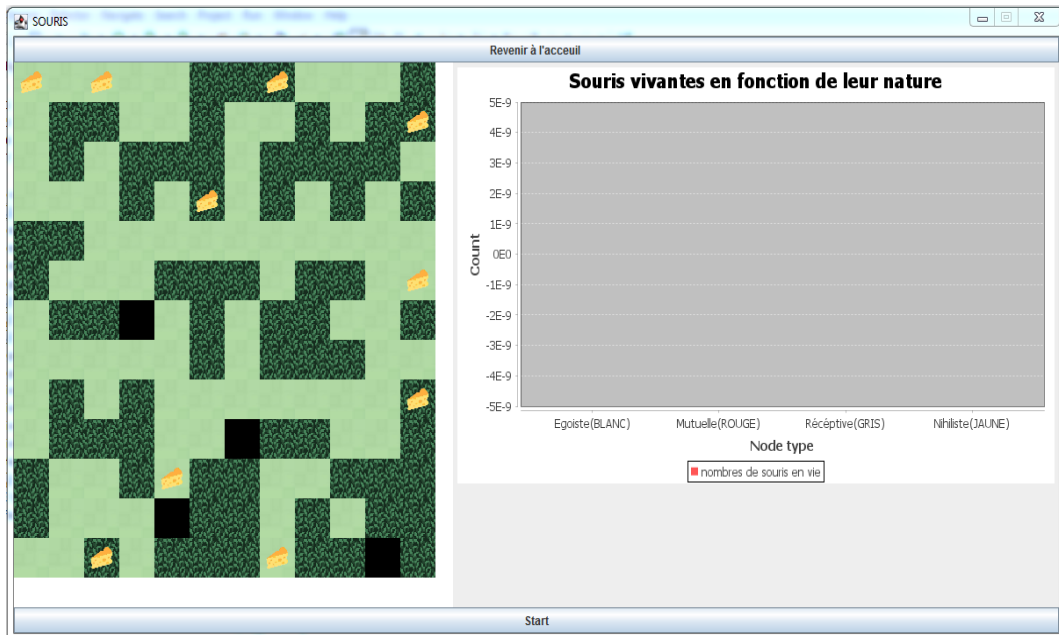


FIGURE 8 – La fenêtre principale

Après sur (la figure 5) on a notre simulation une fois on la lancer y a les souris qui circulent sur la grille en quatre couleurs, la nourriture, les obstacles, on a aussi de l'autre côté les statistiques de la partie pour voir le nombre de souris en vie. suite à ça on a le bouton revenir à l'accueil qui permet à l'utilisateur réinitialiser la partie et de revenir sur la page d'accueil et de choisir de nouveaux paramètres .

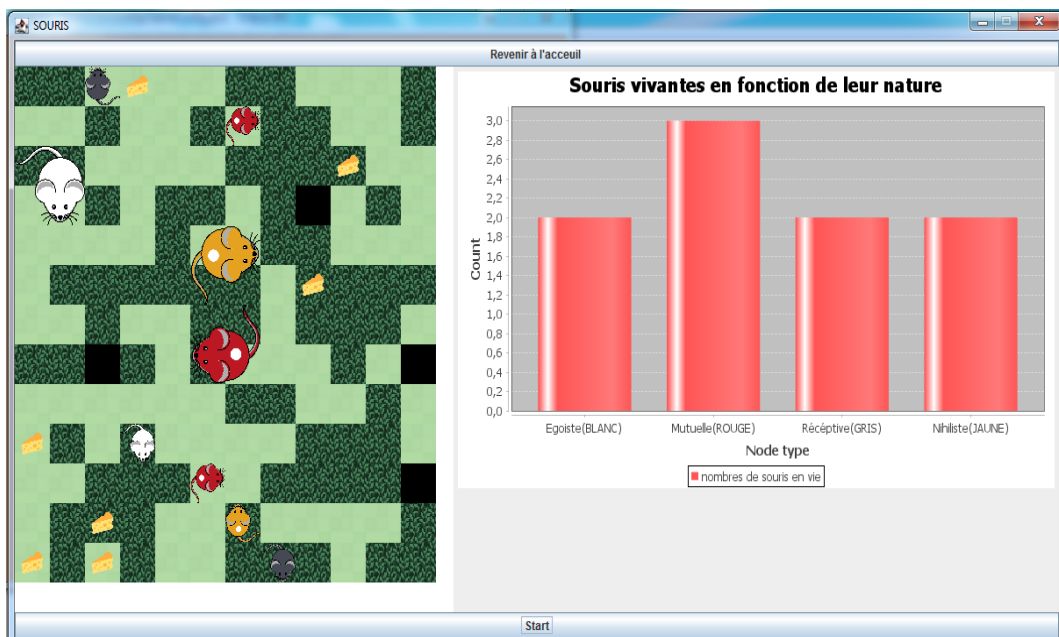


FIGURE 9 – Encore de la simulation

5 Déroulement du projet

Dans cette section, nous décrivons comment le projet a été réalisé en équipe : la répartition des tâches, la synchronisation du travail en membres de l'équipe, etc.

5.1 Réalisation du projet par étapes

- «SOURIS» : Présent en tête de page.
- Avant de commencer la réalisation du projet, nous avons d'abord effectué beaucoup de recherches et on s'est renseigné concernant les éléments essentiels de notre projet.
- Premièrement, nous nous sommes inspirés du logiciel "Aircraft" du professeur vu en cours pour nous aider à réaliser notre jeu.
- Nous avons ensuite pour les caractères des souris dans la classe «Souris» : A, B, C, D quatre caractères, chaque caractère on a spécifié une couleur et un mouvement par exemple pour les caractères on a :
 - Des egoistes avec la couleur gris.
 - Des mutuelles avec la couleur blanc.
 - Des receptives avec la couleur noir.
 - Des nihilistes avec la couleur rouge.
- Ensuite nous avons créé les bornes de nourritures et les obstacles.
- Puis on a pris des photos pour chaque position (bas, haut, gauche, droite) pour faire pivoter les souris dans la classe <PaintStrategy>
- Agrandir les souris à condition de manger au moins une source de nourriture dans la classe <MobileElement Manager>
- Avec la classe "ChartManager" on a géré les statistiques pour voir le nombres de souris encore en vie.
- Ensuite, nous avons créé une nouvelle fenêtre "page d'accueil" permettant d'accéder au différent fonctionnalités avec 2 boutons : Start, help et les 4 boutons pour choisir le nombre de souris, d'obstacles, de nourritures et la dimension de la grille.
- Bouton «Start » : qui permet à l'utilisateur de commencer la partie.
- Deux boutons «choisir une dimension » : Qui proposera tout d'abord à l'utilisateur de choisir la taille de son de sa grille. Si aucune taille est déterminée préalablement par l'utilisateur, une taille par défaut* sera attribuée dans la classe «GameBuilder».
- Bouton «Nombres de souris» : Qui permettra de proposer à l'utilisateur de choisir le nombres de souris.
- Bouton «Nombres de nourriture» : Qui permettra de proposer à l'utilisateur de choisir le nombres de nourriture.
- Bouton «Nombres d'obstacle» : Qui permettra de proposer à l'utilisateur de choisir le nombres d'obstacles
- Bouton «Help» : Redirigera l'utilisateur vers une page web qui présentera l'intégralité du fonctionnement de base du logiciel et définira chaque outil. Cette page web s'accompagnera de texte explicative afin de faciliter la compréhension.
- Bouton «Revenir a l'accueil » : qui permet à l'utilisateur de revenir a la page d'accueil et choisir de nouveaux paramètres.

5.2 Répartition des tâches entre membres de l'équipe

ci-dessous on a le tableau de répartition des tâches entre les membres, on a fait les choses ensemble.

Semaine	ROUAS Lila	MALEK Feriel
semaine 3	Documents de spécifications	Documents de spécifications
semaine 4	Classe de données	Classe de données
semaine 5	Classe de traitements	Classe de traitements
semaine 6	Création d'une grille	Création d'une grille
semaine 7	package Souris	package Souris
semaine 8/9	Déplacement d'une souris	Comportement d'une souris
semaine 10	Moteur	Moteur
semaine 11	IHM graphique	IHM graphique
semaine 12	Statistiques	Statistiques
semaine 13/14	Slides non techniques	Slides techniques
semaine 16	rapport de projet	rapport de projet

FIGURE 10 – Les tâches entre les membres

5.3 Synchronisation du travail

- Pour rédiger ce rapport nous avons utilisé Overleaf cela nous a permis de chacun de s'occuper d'une section et ainsi rédiger ce rapport à un rythme élevé.
- Nous avons utilisé Teams et discord pour synchroniser nos travaux, après les mettre sur GitHub.

6 Conclusion et perspectives

Pour conclure, nous avons réalisé une simulation montrant l'évolution d'un ensemble de souris au fil des reproductions, en affichant les informations statistiques. Les souris explorent régulièrement la grille pour chercher de la nourriture afin de survivre.

Ce projet nous a pris beaucoup de temps, nous avons réalisé une grande partie de ce que nous avions prévu. Néanmoins la simulation ne permet pas une expérience entièrement satisfaisante car l'utilisateur n'a que peu d'interactions avec la simulation.

6.1 Résumé du travail réalisé

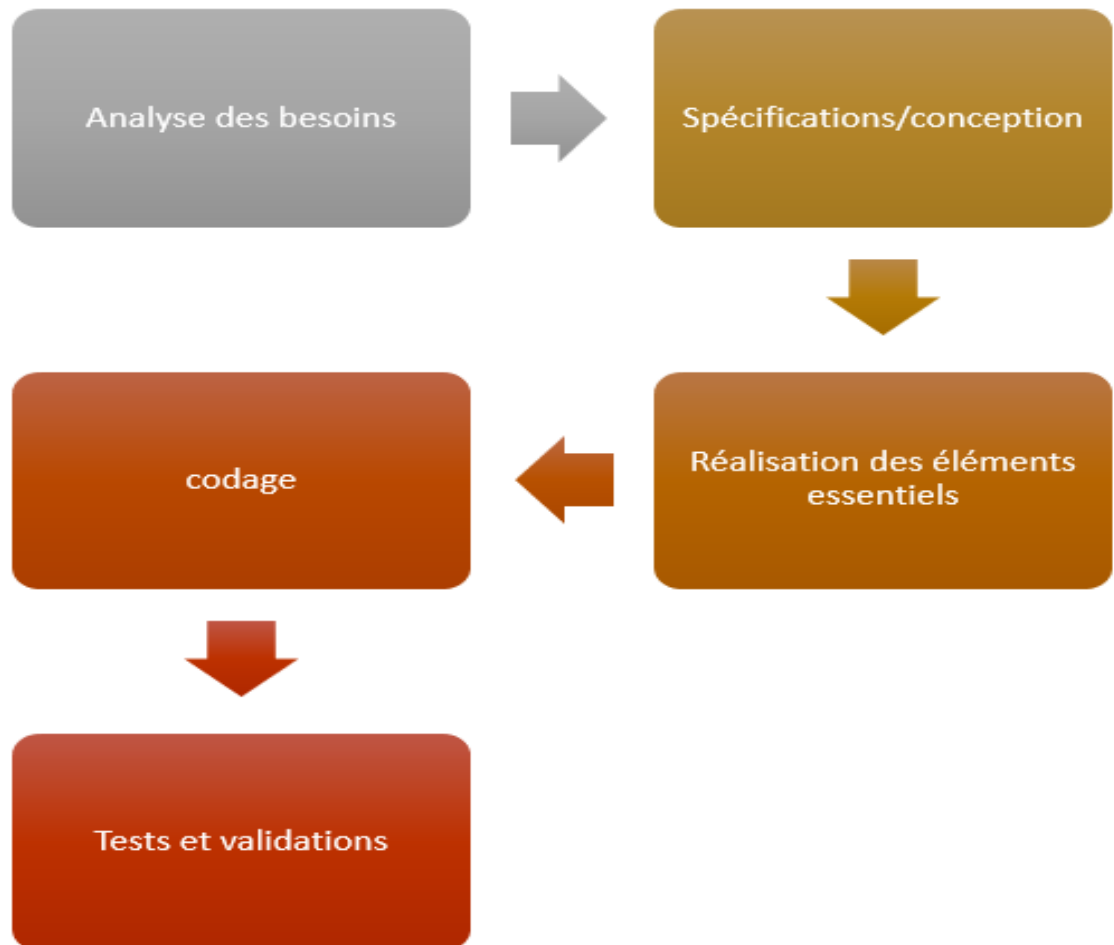


FIGURE 11 – Résumé du travail réalisé

6.2 Améliorations possibles du projet

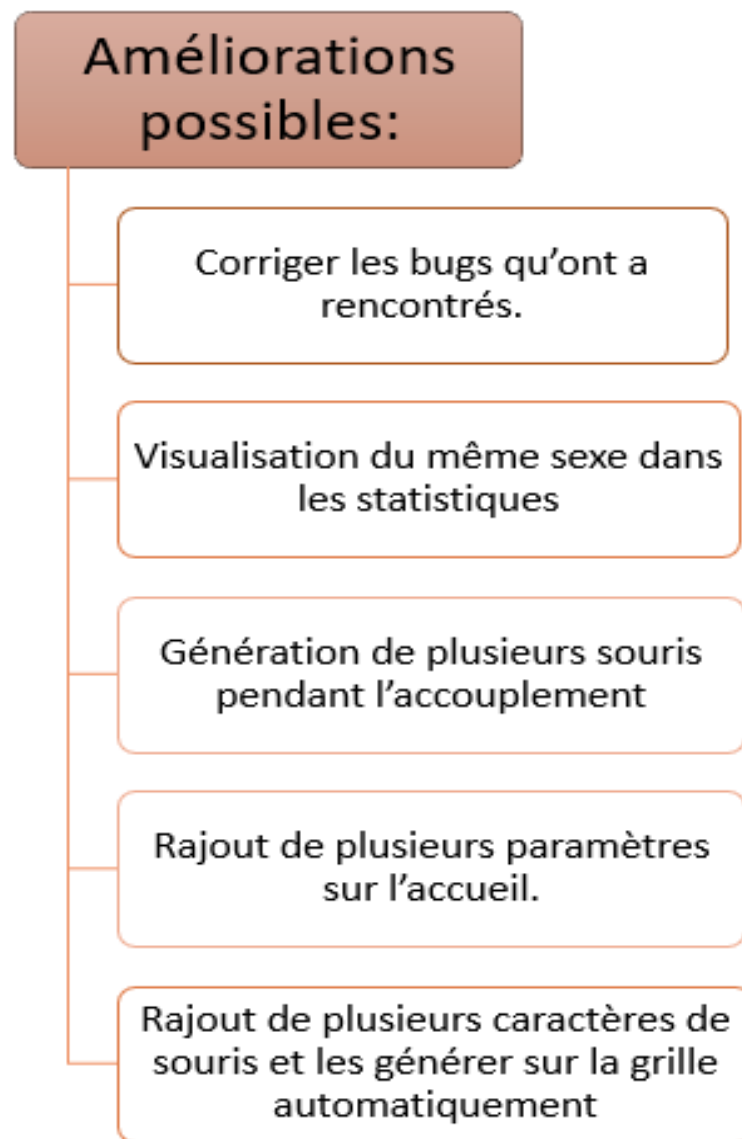


FIGURE 12 – Améliorations possibles

Références

[Fer] MALEK Feriel. *Projet GL*. feriel.malek@etu.u-cergy.fr.

[Lil] ROUAS Lila. *Projet GL*. lila.rouas@etu.u-cergy.fr.

Librairies externes qu'ont a utiliser :

- jfreechart-1.0.19.jar
- jfreechart-1.0.19-experimental.jar
- jfreechart-1.0.19-swt.jar
- jcommon-1.0.23.jar
- jfreesvg-2.0.jar
- junit-4.11
- log4j-1.2.17.jar
- hamcrest-core-1.3.jar
- swtgraphics2d.jar
- servlet.jar
- orsonpdf-1.6-eval.jar
- orsonpdf-1.4-eval-nofx.jar