

Université de Cergy-Pontoise

RAPPORT

pour le projet Génie Logiciel
Licence d'Informatique deuxième année

sur le sujet

Souris

rédigé par

**COLL Matthieu, GALOUZ Zinedine, RANDRIANANTOANINA
Rova**

Mai 2015

Table des matières

1	Introduction	3
2	Spécification	4
2.1	Fonctionnalités	4
2.1.1	L'environnement	4
2.1.2	Les souris	4
3	Réalisation	5
3.1	Realisation Globale	5
3.2	Conception Détaillée	5
3.2.1	Les souris	5
3.2.2	La simulation	6
3.2.3	La grille et sa génération	6
3.2.4	Calcul du chemin le plus court	7
3.2.5	Affichage de la grille (IHM)	7
3.2.6	Autres Fonctionnalités	8
4	Manuel Utilisateur	9
4.1	La Grille	9
4.2	Le Panneau de Contrôle	9
4.3	Les Onglets/Les Fonctionnalités	10
4.3.1	Le log	10
4.3.2	Gestion de la simulation	10
4.3.3	L'onglet d'information	10
4.3.4	L'onglet de génération	13
4.3.5	Les statistiques	13
5	Déroulement du projet	13
5.1	Répartition des tâches	13
5.1.1	Zinedine	13
5.1.2	Matthieu	14
5.1.3	Rova	14
5.2	Synchronisation du travail (SVN)	14
5.3	Calendrier	14
6	Bilan	14
6.1	Réponse au sujet	14
6.2	Apports du projet	14

Table des figures

1	UML prévu au commencement du Projet	5
2	IHM Globale	9
3	Le Control Panel	10
4	L'onglet Log	10
5	L'onglet Sim	11
6	Le Panneau d'information affichant une case	11
7	Le Panneau d'information affichant une case	11
8	Les différents yeux	12
9	Les différentes bouches	12
10	La vision d'une souris	12
11	Le panneau de génération	13

12	Les statistiques	13
13	Calendrier	15

Remerciements

Nous tenons à remercier en premier lieu Monsieur Tianxiao Liu, notre encadrant lors de ce projet, auprès duquel nous avons pu bénéficier d'un grand soutien. Nous remercions également Madame Tuyêt Trâm Dang Ngoc de faire partie du jury de projet.

1 Introduction

Contexte : Dans le cadre de nos études en licence d'informatique, nous devons réaliser, pour le module de génie logiciel, un projet de programmation java en trinome. Nous avons choisi le projet «Souris» nous avons trouvé intéressant d'étudier le concept de gestion de mémoire et d'intelligence artificielle.

Objet : Créer une simulation où des souris doivent survivre en se nourrissant et en communiquant entre elles.

Ressources : En plus des trois membres de l'équipe « Souris », nous avons reçu l'aide d'autres personnes comme Thomas COLL qui a réalisé les illustrations des visages des souris. Au niveau du matériel, nous possédons trois laptops dotés des systèmes d'exploitation Windows 8.1, Windows 7 et Ubuntu 14.04.

Outils de développement : Nos ressources logicielles sont celles qui nous ont été conseillées et que nous avons utilisées au long du module de Génie Logiciel et Projet. Nous utilisons la plateforme Java 6 ainsi que l'environnement de développement Eclipse. Pour synchroniser notre travail, nous avons utilisé le logiciel de versions SVN. Les tests unitaires automatisés ont été réalisés avec Junit. Enfin, ce document de rapport de projet a été rédigé avec LaTeX.

Structure du document : Notre première partie concernera les spécifications de notre projet, elle sera composée d'un paragraphe sur l'avant-projet et de la liste détaillée des fonctionnalités du projet. Ensuite nous parlerons de la partie réalisation du projet, nous expliquerons l'architecture globale de notre projet mais aussi de manière détaillé, comment a été conçu notre projet au niveau programmation orientée objet Java. Viendra après le manuel utilisateur où le fonctionnement du programme sera explicité à l'aide de captures d'écrans. Puis nous arriverons à la partie "déroulement" où sera défini le calendrier de réalisation du projet et la répartition des tâches. Enfin, nous apporterons notre réponse au sujet et montrerons les différents bénéfices du projet.

2 Spécification

Nous avons présenté l'objectif du projet dans la section 1. Dans cette section, nous présentons la spécification de notre logiciel réalisé. Ceci correspond principalement au cahier des charges.

2.1 Fonctionnalités

2.1.1 L'environnement

La simulation se déroule en tour par tour, dans une grille de $X*Y$ cases, générée aléatoirement. Cette grille est composée de plusieurs obstacles, et de différentes sources de nourriture, dont la quantité peut varier.

La grille peut être créée et modifiée par l'utilisateur. L'utilisateur doit avoir un certain contrôle sur le déroulement de la simulation.

2.1.2 Les souris

L'objectif des souris est de survivre, de ne pas mourir de faim. Elles se déplacent sur la grille case par case, à la recherche de source de nourriture. Les souris ont la possibilité de communiquer entre elles, et s'échanger des informations.

Une souris est aussi capable de se dupliquer, et ainsi créer une souris totalement identique, tout comme deux souris de sexes opposés peuvent donner naissance à plusieurs souris enfants, ayant des caractéristiques différentes.

Chaque souris possède une certaine mémoire, permettant de retenir chaque information vue, et de les considérer dans ses raisonnements. Des souris « spéciales » ont aussi été implémentées. Celles-ci ont des caractéristiques et des comportements bien particuliers, permettant de faire varier la simulation.

3 Réalisation

3.1 Realisation Globale

Le fonctionnement de la simulation est géré par différentes classes contenues dans des packages ayant une fonction bien définie.

- Le package « mouse » contient les classes de données caractérisant les souris, leur mémoire et leur mode de communication. Il contient également les classe des souris spéciales dérivant de la classe "Mouse".
- Le package « griddata » est également composé de classes données. Celles-ci caractérisent la grille, les différents types de case et les paramètres de la grille qui sont stockés indépendamment. Il contient les classes "PraticableSquare" et "ObstacleSquare" dérivant toutes deux de "Square" et composant la classe "Grid".
- Le package « engine » constitue le moteur de la simulation. Il contient les classes de traitement permettant de générer la grille, de lancer la simulation, et de stocker les statistiques relatives à la simulation.
- Le package « gui » constitue quant à lui le moteur graphique de la simulation. Il gère l’affichage des différents composants de la fenêtre de simulation. Il se charge aussi de faire avancer la simulation dans le temps.
- Le package « utilities » contient différentes classes pouvant s’avérer utiles à divers endroits du code.

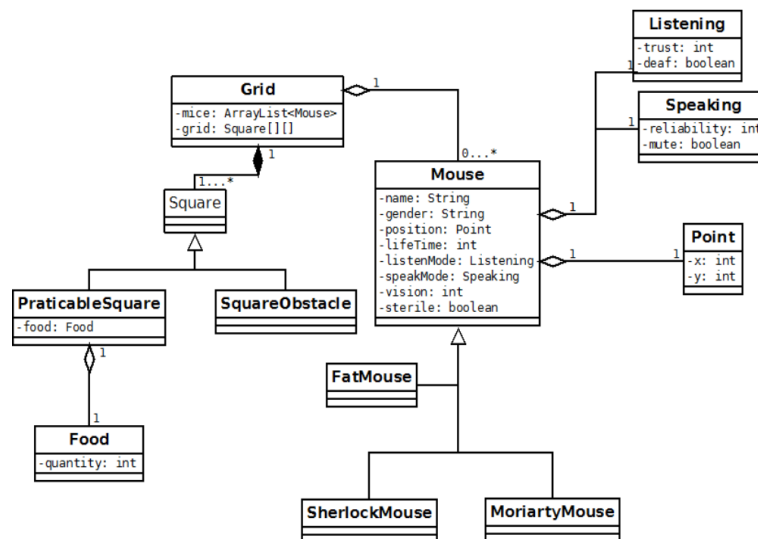


FIGURE 1 – UML prévu au commencement du Projet

3.2 Conception Détaillée

3.2.1 Les souris

Une souris possède plusieurs caractéristiques :

- Age : L’âge d’une souris représente le nombre de tours qu’elle a vécu depuis sa naissance. Ce nombre s’incrémente donc à chaque tour. Une souris meure de vieillesse au bout de 200 tours.
- Vision : Une souris peut voir autour d’elle sur un rayon déterminé aléatoirement lors de sa naissance. Celui-ci varie de 1 à 4.
- Mémoire : La mémoire de la souris est représentée par une grille contenant les cases que la souris a mémorisé. Cette mémoire est limitée à 100 cases. Si la limite est dépassée, les cases mémorisées le plus anciennement sont oubliées. Une souris peut également garder en mémoire

les souris qui lui ont menti. Une souris a une chance sur deux de mémoriser une souris menteuse après s'être faite berner par celle-ci.

- Communication :
 - Une souris peut communiquer avec d'autres souris sur la même case ou sur une case adjacente. La communication permet aux souris de s'échanger des informations concernant la position des sources de nourriture sur la grille. Elle se fait donc en fonction de la mémoire de la souris qui va donner une information.
 - Une souris donnant une information à une autre peut décider de lui mentir, lui donner une information erronée. Dans ce cas, la souris va chercher dans sa mémoire une case dont elle sait qu'elle ne contient pas de fromage, afin d'en indiquer la position à une autre souris. Une souris a plus de chance de mentir si sa fiabilité est faible. Ce paramètre varie de 0 à 5.
 - Une souris qui reçoit une information peut décider de l'accepter ou de la refuser. Ce choix se fait en fonction de la confiance de la souris. Ce paramètre varie de 0 à 5. Plus la confiance est élevée, plus la souris sera naïve, et inversement.
 - Une souris a 1 chance sur 5 d'être muette, c'est-à-dire qu'elle ne peut pas donner d'informations à d'autres souris. Dans le même sens, une souris a 1 chance sur 5 d'être sourde, elle ne pourra donc pas écouter les informations données par les autres souris.
- Pendant un tour, une souris peut :
 - Se dupliquer.
 - Se reproduire avec une autre souris de sexe opposé.
 - Se nourrir.
 - Choisir une cible vers laquelle se déplacer.
 - Se déplace vers sa cible.
 - Se déplacer aléatoirement.
 - Explorer la grille.
 - Accoucher.
 - Mourir.

3.2.2 La simulation

Déroulement de la simulation

- Au commencement, la simulation est instanciée et la génération est lancée. La boucle de simulation se lance alors. Le compteur de tours est incrémenté.
- Les souris agissent une par une et ont un comportement différent selon leur environnement et leurs besoins. Leurs actions font appel à des données plus ou moins globales et modifient les données de la souris ainsi que de la grille. (Voir figure ALGORITHME)
- Les souris venant de mourir ce tour sont supprimées de la grille puis les souris venant de naître sont ajoutées à la grille.
- On décrémente le compteur de régénération de la nourriture. Si le compteur vaut zéro, 5 unités de nourriture sont ajoutées sur une case praticable de la grille. Le compteur revient alors à sa valeur initiale définie dans l'onglet "Sim" de la simulation.
- On boucle afin de revenir à l'incrémementation du nombre de tours.

3.2.3 La grille et sa génération

Une grille est composée de $X*Y$ cases, la taille minimum étant de $5*5$, et la taille de maximum de $50*50$. Les tailles rectangulaires sont autorisées.

Une case peut soit être vide, soit posséder une source de nourriture, soit être un obstacle (Non praticable). // Une source de nourriture est simplement composée d'une quantité de nourriture.

Une grille est générée aléatoirement, en fonction de plusieurs paramètres :

- * La taille initiale de la grille
- * La densité des obstacles (% d'obstacle sur la grille)
- * La densité des sources de nourritures

* Le nombre de souris de départ

On peut également activer ou désactiver les souris spéciales lors de la génération. Tous ses paramètres sont stockés dans une classe "GridParameters", ceci permettant de centraliser les informations, et de facilement pouvoir en rajouter si nécessaire.

C'est la classe factory "Génération" qui va s'occuper de créer la grille.

La génération se déroule en plusieurs étapes :

1. On crée d'abord le tableau (la grille) de la taille voulue.
2. On génère ensuite aléatoirement les obstacles, de manière structurée.
3. C'est à dire que l'on ne va pas juste placer aléatoirement les obstacles, mais bien créer une zone particulière, où chaque case est atteignable depuis n'importe quelle autre case.

Ainsi, si on génère des obstacles sur une densité de 95%, les 5% restant formeront un simple chemin sur lequel les souris pourront se déplacer.

Pour cela, on va sélectionner une case aléatoirement, et vérifier si les 4 cases autour sont peut-être liées (On utilise le calcul du chemin le plus court)

3.2.4 Calcul du chemin le plus court

Les souris auront souvent besoin de déterminer leur chemin pour se déplacer vers leurs cibles. Nous avons donc pour cela écrit un algorithme du chemin le plus court, pour rapidement trouver le chemin le plus optimisé pour relier deux points. L'algorithme que nous avons utilisé est une adaptation de l'algorithme A* à notre projet.

Le raisonnement consiste à marquer au fur et à mesure les cases en notant leur distance avec la case de départ, en formant tous les chemins possibles entre une case et la case de départ, et en conservant uniquement le plus court. On arrivera forcément à la case d'arrivée, sauf si le chemin n'est pas calculable.

3.2.5 Affichage de la grille (IHM)

Au niveau de l'Interface graphique, la grille est affichée dans un JPanel dans lequel nous avons modifié la méthode PaintComponent.

De nombreuses méthodes ont été créées pour afficher différents éléments à un point p.

Nous pouvons ainsi dessiner les images importées dans le projet :
un sol, un obstacle (en fonction de son type), une souris (en fonction de son type et de sa direction)

Ainsi que différentes formes :

Des carrés/cercles/croix, un chemin, un champ de vision, etc.

L'affichage se fait donc en appelant ces différentes méthodes sur chaque case, en fonction de ce que l'on doit afficher.

Plusieurs modes d'affichage ont été créés, appelant d'autres fonctions pour afficher le même contenu d'une manière différente :

- * Mode Global

Affiche la grille avec les images de base

- * Mode Schématique

Affiche la grille avec uniquement des formes géométriques de différentes couleurs et tailles (Utile pour visualiser la forme de la grille, et la position des objets)

* Mode Vision

Affiche la grille du point de vue de la souris. Les couleurs sont différentes, et on ne voit uniquement la mémoire de la souris. (Les choses dont elle se souvient, pas obligatoirement la réalité)

3.2.6 Autres Fonctionnalités

- * Un système de journal est disponible dans l'onglet "Log" de la simulation. Il permet d'avoir un regard précis sur ce qui se passe dans la simulation.
- * Les statistiques de la simulation sont stockées dans la classe "Statistics" et sont affichées dans l'onglet "Infos" de la simulation.
- * L'onglet "Sim" permet à l'utilisateur de paramétrer le taux de régénération des sources de nourriture et la fréquence de reproduction/duplication. Un case à cocher permet aussi de passer la grille en mode schématique.
- * Il est possible de générer la grille autant de fois que l'utilisateur le souhaite par le biais de l'onglet "Génération". Il est ainsi possible de modifier les différents paramètres de la grille.

4 Manuel Utilisateur

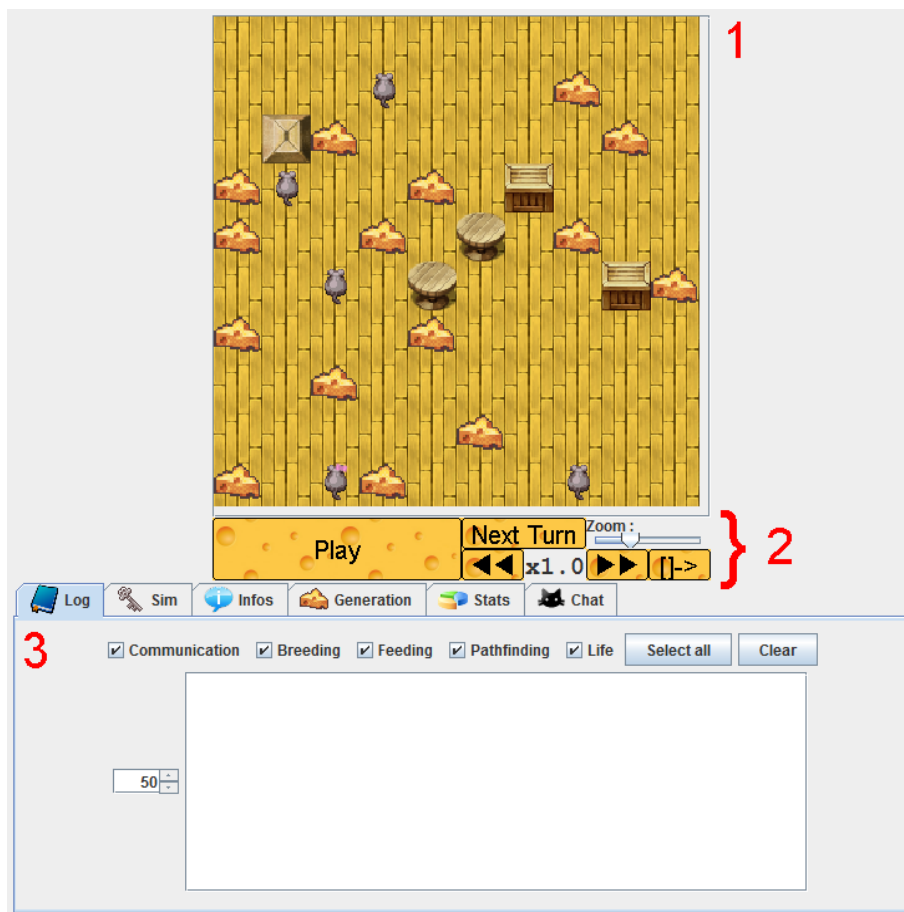


FIGURE 2 – IHM Globale

La figure 5 donne un aperçu des différentes parties de notre IHM. Celle-ci est donc décomposée en 3 parties.

4.1 La Grille

La grille nous permet d'avoir un aperçu général de la simulation. Les cases peuvent être sélectionnées et modifiées, certains paramètres peuvent être réglés, et plusieurs modes d'affichages sont possibles.



Représente les souris de sexe masculin



Représente les souris de sexe féminin



Représente une case praticable



Représentent un obstacle



Représente une source de nourriture

4.2 Le Panneau de Contrôle

Le panneau de contrôle permet de gérer le déroulement de la simulation mais aussi la mise en forme de la grille.

Le panneau de contrôle de la figure 3 est décomposé en 5 parties :



FIGURE 3 – Le Control Panel

1. permet de lancer ou de mettre en pause la simulation.
2. permet d'exécuter un tour.
3. permet d'agrandir ou de réduire la taille de la grille.
4. permet de modifier la vitesse d'exécution d'un tour de x0.25 à x4 la vitesse de base (500 ms).
5. permet de détacher la grille de la fenêtre principale, et de la rattacher.

4.3 Les Onglets/Les Fonctionnalités

La fenêtre principale comporte cinq onglets : "Log", "Sim", "Infos", "Generation" et "Stats".

4.3.1 Le log

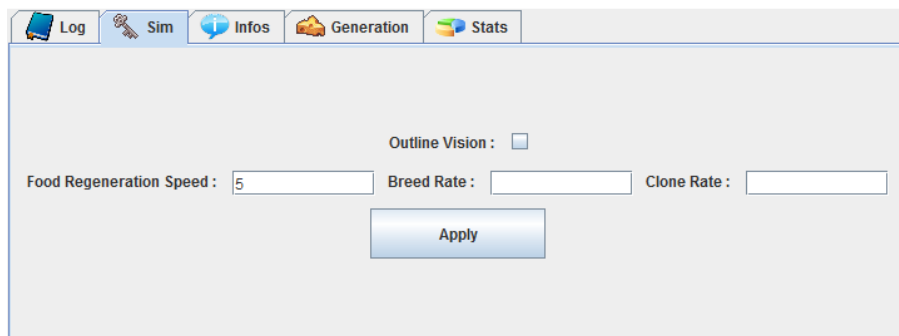


FIGURE 4 – L'onglet Log

C'est dans le log (voir figure 4) que sont affichées les lignes du système de journal de notre simulation . Il est possible de filtrer ces informations selon leur type (communication, reproduction, alimentation, mouvement et vie) . Il est également possible de limiter le nombre de lignes affichées. Un bouton est disponible pour vider le journal et un autre pour cocher simultanément toutes les cases du filtre.

4.3.2 Gestion de la simulation

L'onglet de gestion de la simulation (voir figure 5) permet de modifier plusieurs paramètres utilisés durant le déroulement de la simulation :

- * La vitesse de régénération du fromage
- * La fréquence des accouplements
- * La fréquence des duplications.

On peut aussi passer de la vue globale à la vue schématique à l'aide de la case "Outline Vision".

4.3.3 L'onglet d'information

C'est dans cet onglet que l'on va accéder aux informations précises des cases, et que l'on va pouvoir modifier les données. Il est possible, en cliquant sur la grille, de sélectionner une case. C'est dans cet onglet que sera affiché les informations (voir figure 6)

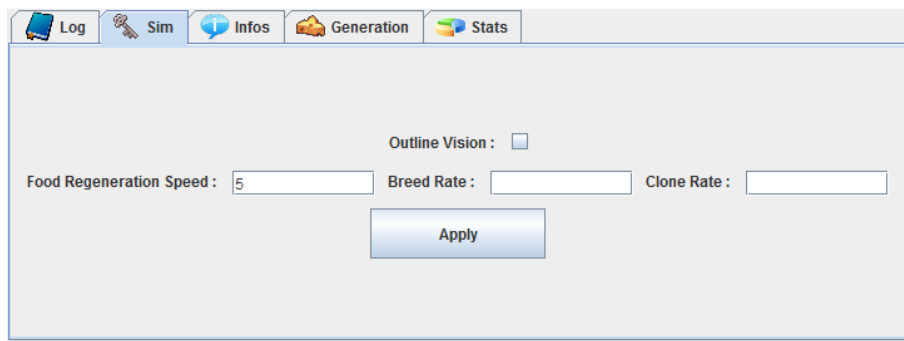


FIGURE 5 – L'onglet Sim

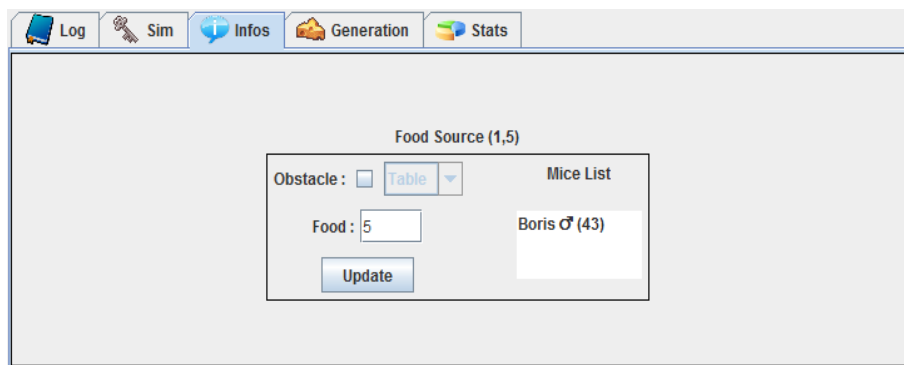


FIGURE 6 – Le Panneau d'information affichant une case

Si la case est praticable il est possible de rajouter des unités de nourriture ou de transformer cette case en obstacle. Dans "Mice List" est répertoriée le nom, le sexe et les points de vie des souris occupant la case. Un double-clic sur une souris de la liste permet d'afficher ses caractéristiques (voir 7)

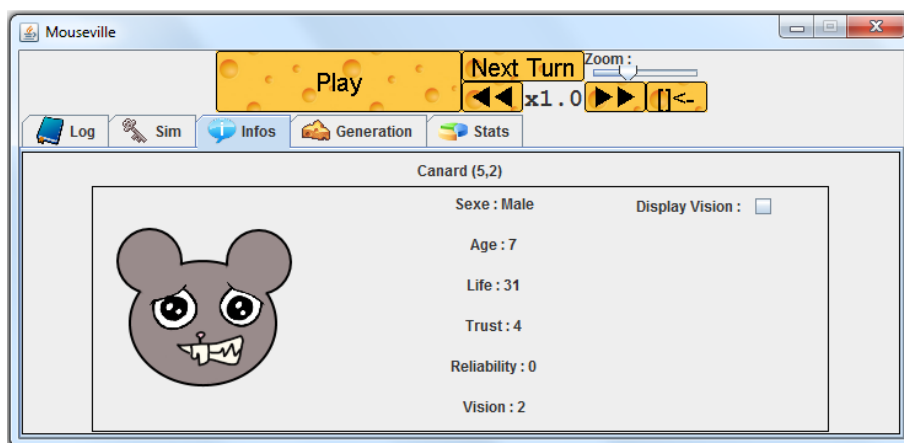


FIGURE 7 – Le Panneau d'information affichant une case

Cette partie affiche également l'apparence de la souris. Cette fonctionnalité n'est que visuel, et permet simplement de représenter le degré de fiabilité et de confiance d'une souris.

Le degré de confiance est représenté à travers les yeux



FIGURE 8 – Les différents yeux

Sur la figure 8, on peut voir les yeux pour les différents niveaux de confiance de la souris allant de 0 à 5 (de très méfiant à très crédule).

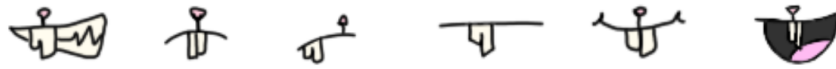


FIGURE 9 – Les différentes bouches

Sur la figure 9, on peut voir les bouches pour les différents niveaux de fiabilité de la souris allant de 0 à 5 (de menteuse à très fiable).

Quelques accessoires et modifications visuelles ont été ajoutées pour différencier certaines particularités. (un noeud papillon pour les souris de sexe féminin, un chapeau pour la souris Sherlock, etc..)

Cette partie permet aussi d'afficher la vision de la souris (sa mémoire, la position des fromages et les obstacles dont elle se souvient)(voir figure 10), en cochant la case "Display Vision".

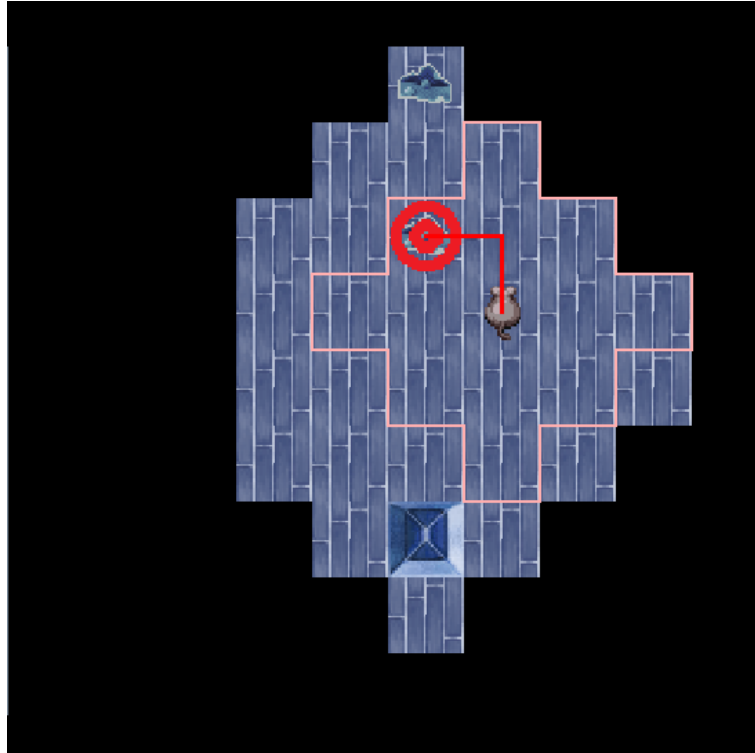


FIGURE 10 – La vision d'une souris

Le tracé rouge et la cible indiquent alors la trajectoire que va suivre la souris.

4.3.4 L'onglet de génération

Cet onglet (voir 11) permet de régénérer une grille en fonction de plusieurs paramètres (taille, densité d'obstacles/nourriture, souris spéciales, etc..)

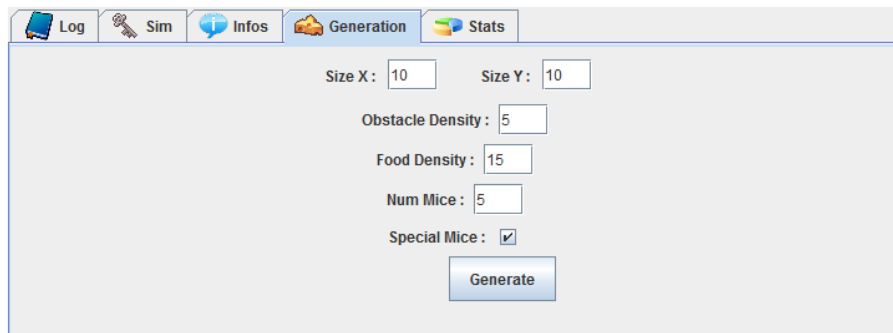


FIGURE 11 – Le panneau de génération

4.3.5 Les statistiques

Cet onglet voir (12) permet d'afficher des statistiques sur la simulation en cours.

De base sont affichées le nombre de souris vivantes, le nombre de souris mortes, le nombre de souris mâles et femelles présentes, le nombre de source de nourriture, la quantité totale de nourriture restantes sur la grille, le nombre de souris stériles et le nombre de souris enceintes.

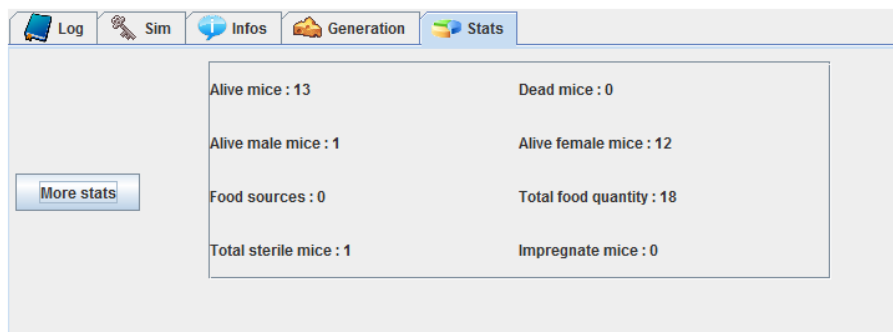


FIGURE 12 – Les statistiques

Le bouton "More stats" permet d'afficher plus de statistiques dans une nouvelle fenêtre. On aura donc à des informations plus détaillées.

5 Déroulement du projet

5.1 Répartition des tâches

5.1.1 Zinedine

- Raisonnement des souris
- Gestion de la nourriture
- Mémoire des souris

5.1.2 Matthieu

- Génération structurée de la grille
- IHM : Grille, Generation, Panneau de controle, Informations cases/souris
- Visuel des Souris
- Contrôle de la simulation

5.1.3 Rova

- IHM : Optimisation, Layout, Mémoire/Vision des souris, Statistiques
- Calcul des Statistiques

5.2 Synchronisation du travail (SVN)

Nous avons utilisé le logiciel Apache Subversion afin de synchroniser nos travaux. Ce logiciel nous a permis de travailler en équipe de manière simple et rigoureuse. Nous avons effectué des commits au rythme de un par semaine en moyenne. Les commits furent très rapprochés dans le temps au début du projet afin que tout le monde ait accès aux classes de données. À l'inverse, à la fin du projet, les modifications apportées par les membres de l'équipe étant mineures, les commits étaient beaucoup plus éloignés dans le temps. Nous avons fixé une certaine norme à respecter par rapport aux messages de commit : les messages doivent être rédigés intégralement en anglais. Les modifications doivent être spécifiées sous forme de listes précédées du type de modification (ajout, modification, amélioration).

5.3 Calendrier

Le calendrier (voir figure 13) liste chronologiquement, sans détail, les différentes tâches (regroupées par semaine) que nous avons exécuté au cours de ces mois de projet.

6 Bilan

6.1 Réponse au sujet

Nous avons réalisé une simulation de société de souris, c'est à dire une représentation du comportement de souris vivant en communauté, en prenant en compte les besoins de nourriture et reproduction. Les souris survivent en explorant et en gardant en mémoire les sources de nourriture visitées ou en communiquant avec d'autres souris coopératives. Une fois bien nourries, les souris peuvent assouvir leur second besoin, celui de se reproduire (en plus la duplication). Comme dans toutes sociétés, nous avons mis en place différents niveaux de confiance et de fiabilité. De plus, le comportement d'une souris évoluent en fonction de son environnement et des caractères de leur entourage. Ainsi, une souris peut devenir très méfiante avec celles qui ont tenté de la duper.

6.2 Apports du projet

Ce projet nous a donc beaucoup apporté, notamment au niveau des notions de programmation orientée objet et de Java ; mais surtout sur le plan gestion de projet. Ainsi durant nos séances de travaux dirigées de génie logiciel projet, nous avons découvert progressivement les différentes étapes de conception d'un projet et son organisation. Nous avons dû aussi gérer un calendrier de tâches et la synchronisation du travail de groupe, ce qui ne fût pas une tâche facile..

Semaine	Zinedine	Matthieu	Rova
2 Février	Classes de données	Classes de données	Classes de données
9 Février	Mémoire des souris	Simulation et déplacements	Génération
16 Février	Concertation	Concertation	Concertation
23 Février	Comportement des souris	Pathfinding et IHM	
2 Mars	Point d'avancement 1	Point d'avancement 1	Point d'avancement 1
9 Mars	Infos dans l'IHM	Sprites dans l'IHM	Gestion des clics
16 Mars	Communication et Log	Correction de la Génération	Mode mémoire de la grille
23 Mars	Apparence de la vision		Correction du Log
30 Mars	Concertation	Concertation	Concertation
6 Avril	Point d'avancement 2	Point d'avancement 2	Point d'avancement 2
13 Avril			Statistiques
20 Avril			
27 Avril			
4 Mai			
11 Mai			
18 Mai	Corrections diverses	Corrections diverses	Corrections graphiques

FIGURE 13 – Calendrier