

Université de Cergy-Pontoise

RAPPORT

pour le projet Génie Logiciel
Licence Informatique 2ème année

sur le sujet

GARDIEN DE PARC

rédigé par

Makoura Karabouali, Antoine Phetramphand,
Ramatoulaye Tangara

Mai 2014

Tout d'abord, nous tenons à remercier notre tuteur de projet Monsieur Liu Tianxiao. Son aide a été précieuse voir indispensable à la réalisation de ce projet. Il a toujours été présent par l'intermédiaire des emails et des TD pour répondre à nos questions et nous fournir de l'aide. Nous remercions également les enseignants de l'équipe pédagogique de l'UFR des Sciences et Techniques de Cergy-Pontoise. Enfin, nous remercions nos camarades pour leur aides et leur conseils.

Table des matières

0.1	Introduction	3
0.1.1	introduction générale	3
0.1.2	Présentation du projet	3
0.2	Fonctionnalités du jeu	3
0.2.1	But du jeu	4
0.3	Fonctionnement	5
0.3.1	Organisation	5
0.3.2	Réalisation	5
0.4	Manuel d'utilisation	8
0.4.1	Mode Console	8
0.4.2	Mode graphique(IHM)	9
0.5	Difficultés rencontrées	11
0.6	Résultats	12
0.7	Conclusion	12

graphicx

0.1 Introduction

0.1.1 introduction générale

L'intelligence artificielle est de plus en plus utilisée dans le monde d'aujourd'hui. Le multi-agents réactifs est une méthode de programmation pour créer des intelligences artificielles. Dans le cadre de notre formation à l'Université de Cergy Pontoise, les étudiants en Licence informatique deuxième année réalisent des projets personnels afin de valider leur année scolaire. Ce type de réalisation exige donc un certain temps, un investissement personnel et l'usage de toutes les ressources disponibles. Ici, nous allons réaliser notre projet sous la forme d'un jeu vidéo appelé « le gardien de parc ».

0.1.2 Présentation du projet

Afin d'appliquer les méthodologies et les notions enseignées au cours de l'année, nous devons réaliser un projet consistant à concevoir un jeu de type multi-agents. Celui-ci consiste à initialiser une grille, soit de manière aléatoire suivant certains paramètres données par l'utilisateur, soit de manière manuelle par l'utilisateur. Dans cette grille, nous devons pouvoir placer des personnages (gardiens et intrus) ainsi que des décors qui représenteront des obstacles (arbre, mur et eau) pour empêcher les déplacements de certains personnages. Par la suite, nous avons la responsabilité de simuler les actions des gardiens et des intrus sur une même grille pour que le gardien puisse attraper et chasser les intrus présentant sur son champ de vision. Cependant, le gardien doit tenir compte des obstacles et les éviter lors de sa quête.

Au cours du jeu, l'utilisateur doit pouvoir soit contrôler manuellement les actions (déplacement etc) du gardien, soit de manière aléatoire en laissant le gardien faire ces actions aléatoirement.

0.2 Fonctionnalités du jeu

L'application du jeu vidéo réalisée nous a permis de réaliser les fonctionnalités suivantes :

- Présenter au joueur une interface d'entrée dans le jeu : cette interface expliquera à l'utilisateur la tâche qu'il devra accomplir,
- Mettre en évidence des éléments importants de la scène en les surli-
gnant : on met en valeur, en premier plan, les gardiens et les intrus afin
de savoir pour l'utilisateur qui on peut contrôler et qui sont les cibles à
éliminer ; puis en second plan les obstacles du jeu(eau, arbre, mur...),
afin de permettre au joueur de prendre en compte ces éléments et leurs
fonctions,
- Offrir une simplicité d'utilisation : le jeu devra être facile à utiliser et
à manipuler, c'est-à-dire moins de touches du clavier à utiliser, suivi
des touches de la souris.
- Offrir une rapidité de mise en œuvre : le jeu doit permettre à des
joueurs de mesurer leurs réflexes mentaux afin qu'il puisse prendre des
décisions (ex : trouver un chemin le plus court possible pour arriver
jusqu'à l'intrus) le plus rapidement possible lors de la partie du jeu
- Assurer des temps d'exécution faibles : lors du lancement du logiciel,
les temps de chargement des programmes doivent être rapides.
- Assurer des temps d'interactions joueur / machine faibles : durant
l'exécution du logiciel, l'utilisateur ne doit sentir aucune latence, que
ce soit avec le clavier ou la souris. Les déplacements du joueur doivent
être fluides.

0.2.1 But du jeu

Le jeu consiste à chasser des personnages de type intrus par des per-
sonnages de type gardien dans un parc composé par des obstacles qui rend
difficile les déplacements.

graphicx

0.3 Fonctionnement

0.3.1 Organisation

Notre équipe est composée de trois membres à la charge du projet : Karabouali Makoura , Phetramphand Antoine et Tangara Ramatoulaye. Pour mieux appréhender le projet, nous avons entamé des recherches et approfondir les non acquis et mettre le problème en oeuvre techniquement en utilisant le processus «modèle en cascade» Chaque semaine, nous nous sommes réunis

pour réfléchir à l'avancement de nos différents travaux et voir les solutions possibles en cas d'erreur. Et pour mieux travailler en parallèle, nous avons utilisé les messages électroniques afin de s'informer sur l'avancement de nos travaux respectifs pour permettre de mieux fixer la prochaine mise en commun.

Tout d'abord, nous avons commencé par le développement du mode console, puis du mode graphique car celui-ci utilise une partie du code du mode console. Pour chaque avancement nous avons établi des tests afin d'être certain du bon fonctionnement avant de passer à autre chose.

0.3.2 Réalisation

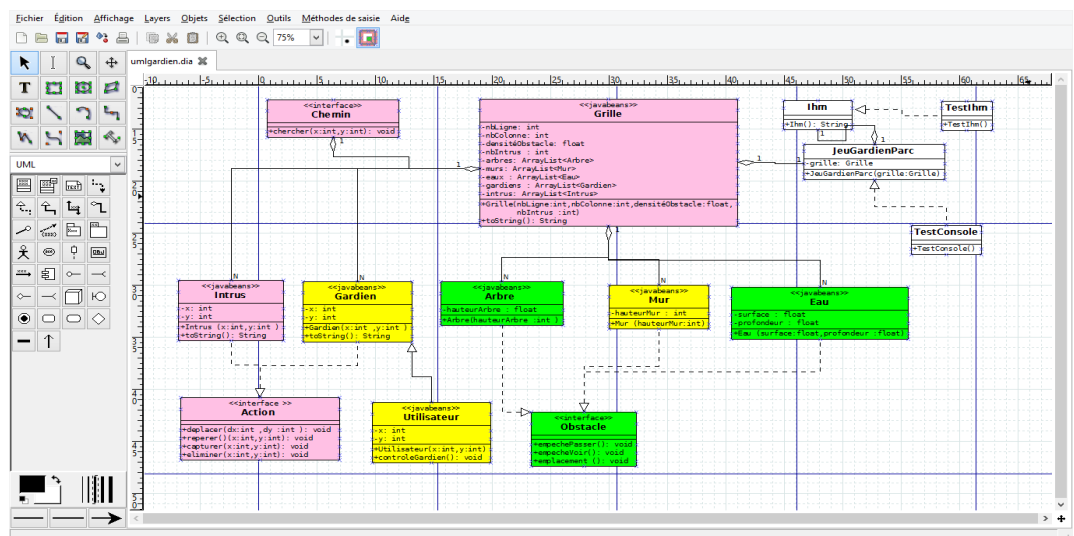
Pour pouvoir bien mener le projet nous avons suivi les étapes suivantes :

- 1ème Etape : Recherches Nous avons effectué des recherches sur internet ainsi que des recherches bibliographiques sur notre sujet afin d'approfondir et mieux résoudre le problème.
- 2ème Etape : Répartition des tâches Le sujet étant libre, nous devons nous fixer notre propre cahier des charges. Nous avons décidé de fonctionner avec plusieurs versions. Pour chaque nouvelle version, nous nous réunissions pour conceptualiser un nouveau problème et une nouvelle solution (nouveau cahier des charge), puis après avoir défini le diagramme UML de notre projet, nous nous répartissions les tâches pour effectuer chacun une partie du travail. Enfin, nous nous réunissions pour mettre en commun, effectuer une batterie de test et corriger les éventuelles erreurs. Cependant, chacun avait des tâches prioritaires à accomplir.
- 3ème étape : développement des classes

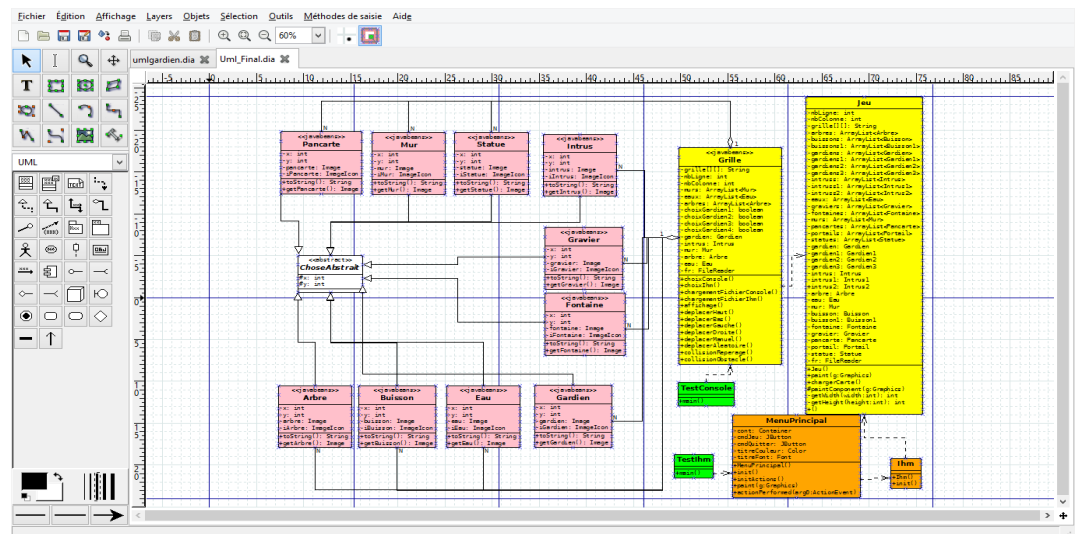
Tâches	Personne
mode graphique	Antoine
mode console	Ramatoulaye
documentation	Makoura

TABLE 1 – Répartition des tâches

Pour développer nos classes, nous nous sommes tout d'abord basés sur le diagramme UML suivant :



Puis par la suite, nous nous sommes rendus compte que la version conçue n'était pas celle qui nous fallait car elle ne permettait pas à l'utilisateur de choisir ces propres dimensions (taille de la grille), ni de placer lui-même les éléments ou encore de choisir le nombre d'éléments qu'il souhaite mettre ne place. Nous en sommes venus alors après élaborations de plusieurs versions à la version finale dont le digramme UML est le suivant :



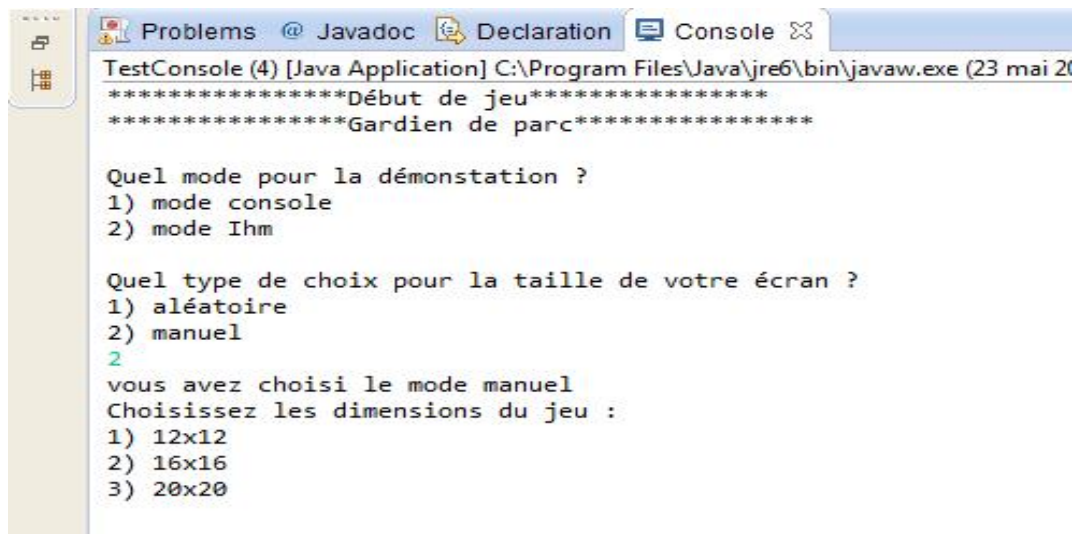
Cette version donne plus de liberté qu'auparavant.

graphicx

0.4 Manuel d'utilisation

0.4.1 Mode Console

Pour l'utilisation du mode console, l'utilisateur doit lancer dans le package test le TesteConsole. Puis En lançant le « TestConsole », on nous invite à faire un choix en appuyant soit sur « 1 » pour le mode aléatoire ou « 2 » pour le mode manuel. Ici, nous allons appuyer sur « 1 » pour exécuter le mode Console. Si c'est un choix aléatoire, cela initialisera les variables aléatoirement le nombre de lignes et le nombre de colonnes de la grille. Si c'est un choix manuel, le programme invite à l'utilisateur de choisir les dimensions proposées de la grille : soit d'un format 12x12, 16x16 ou 20x20 (de type « nombreLignesxnombreColonnes »).



```
TestConsole (4) [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (23 mai 2011)
*****Début de jeu*****
*****Gardien de parc*****

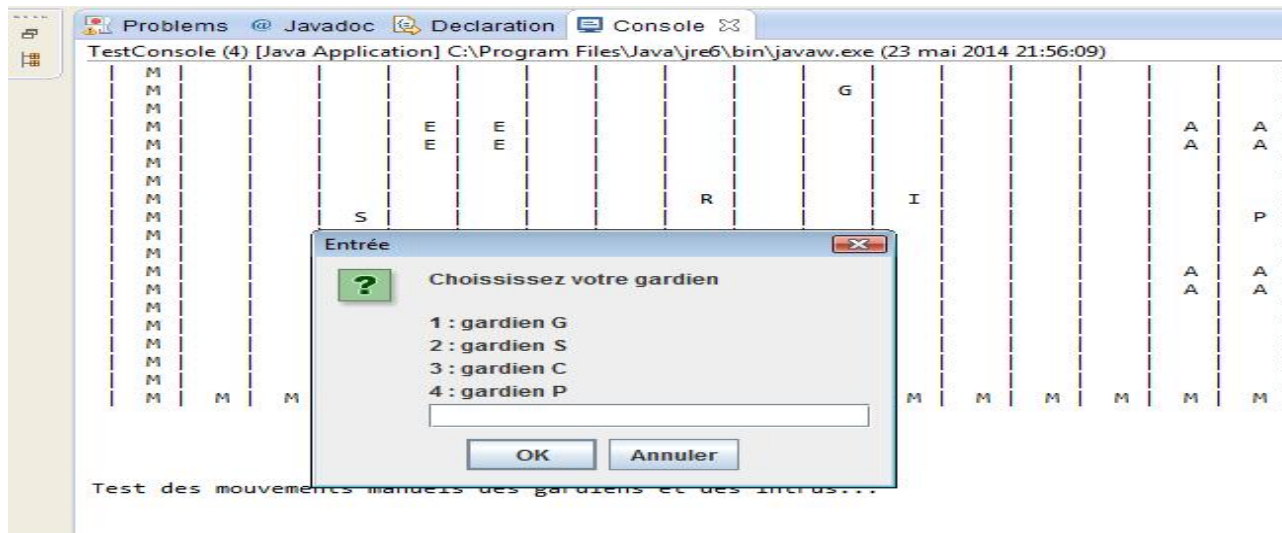
Quel mode pour la démonstration ?
1) mode console
2) mode Ihm

Quel type de choix pour la taille de votre écran ?
1) aléatoire
2) manuel
2
vous avez choisi le mode manuel
Choisissez les dimensions du jeu :
1) 12x12
2) 16x16
3) 20x20
```

Vue d'ensemble du mode console. Ensuite, cela affichera la grille du jeu,

ainsi que les éléments : Les gardiens sont représentés par G, S, C et P ; les Intrus sont représentés par I, R et D ; le reste par M(les murs), A(les arbres) et E(les eaux). Après avoir initialisé la grille, on passe en phase de test manuel des éléments du jeu. Par défaut, Si vous choisissez de saisir les touches différentes de « 5 » et « 6 », cela déplacera uniquement les Intrus (I, R et D). Si vous voulez déplacer l'un des gardiens (G, S, C et P), vous devez

saisir la touche « 5 » pour sélectionner l'un d'eux. Une fois appuyé, vous devez choisir un chiffre entre 1 et 4 suivant les informations indiquées. Si vous saisissez la touche « 6 », on passera en phase de déplacement aléatoire où tous les gardiens et intrus bougeront sans cesse aléatoirement.



Vue du mode

0.4.2 Mode graphique(IHM)

En lançant le « TestIhm », cela ouvrira un menu qui offrira deux options : Soit vous cliquez sur « Commencer » et vous lancez la partie ; soit vous cliquez sur « Quitter » et le programme sera fermé .Une fois le jeu lancé, on nous donne les règles du jeu.
Ensuite, cela affichera l'écran du jeu :



0.5 Difficultés rencontrées

La réalisation de ce projet n'était pas chose facile, il nous a demandé beaucoup de réflexion, de temps, de recherches, de motivation, ainsi que d'organisation et de discipline. La difficulté majeure que nous avons rencontrés est la mise en place du mode IHM, c'est à dire de pouvoir simuler les déplacements des personnages et de leurs actions.

Le repérage et la capture des intrus par les gardiens ont été aussi un souci majeur du fait qu'il fallait définir le champ de vision du gardien à partir duquel on calcule le chemin le plus rapide (grâce à ASTAR) pour permettre le captelage de l'intrus par le gardien le plus proche. Pour finir la

mise en commun des différentes versions, cela a posé à des petits problèmes. à cause de cela notre projet présente 2 versions du mode console car nous avons pas eu suffisamment de temps pour raccorder les 2. Donc nous pensons que la gestion du temps a été un problème non négligeable dans notre groupe.

0.6 Résultats

Après des semaines de travail acharné , nous pensons avoir effectué que 3/4 du travail demandé en mode console et pour le mode graphique, nous avons pu obtenir que l’affichage des éléments sans actions derrière.

Si nous devons améliorer ce projet, nous voudrions d’abord optimiser la simulation des actions des personnages avec plus de fonctionnalités .

0.7 Conclusion

La réalisation de ce projet nous a beaucoup apporté. Tout d’abord, elle nous a appris ce qu’est le travail en groupe sur une longue durée. Entre autre, nous avons appris à :

- répartir les tâches entre des personnes
- sécuriser les données pour éviter toute perte
- mettre en commun le code créé individuellement

De plus, au niveau informatique, nous avons découvert et maîtrisé de nouvelles techniques de programmation (multi-agents) et perfectionné notre programmation Java (notamment le graphisme). Enfin, ça nous a obligé à rédiger un rapport conséquent, structuré et destiné à des lecteurs appartenant à différents domaines. Ceci nous aura été bénéfique pour des projets futurs que ce soit dans les études ou dans la vie professionnelle.