

Mice'n'Cats

Sujet de BE de TC1, master 1ère année

L. Broto, H. Cassé, J.P. Georgé, G. Picard

22 octobre 2004

1 Organisation du projet

Ce document décrit le sujet du projet de BE pour le module TC1 des mastères 1ère année.

1.1 Déroulement du projet

Il s'agit de réaliser le projet « Mice'n'Cats » décrit dans la suite de ce document. Ce projet devra être développé durant les séances de TP et sera sanctionné par :

- la remise du programme « clé-en-main », prêt à être utilisé avec une notice d'utilisation ;
- la remise d'un mini-rapport précisant l'architecture du programme, les choix en matière de parallélisme et de synchronisation et les schémas UML utilisés pour réaliser le logiciel.

Il s'agit en premier lieu de se mettre en situation réelle de développement. A partir de la remise du sujet, vous devrez concevoir, réaliser et tester le projet ainsi qu'aménager le planning de réalisation. Vous pouvez travailler pendant le TP ou hors du TP mais le projet devra être impérativement rendu à la fin du dernier TP.

Les enseignants sont bien sûr à votre disposition pour répondre à toute question technique ou autre.

Le sujet correspond à l'étude du besoin du client. Il comporte également des contraintes structurales techniques qui devront être satisfaites.

1.2 Références

La réalisation du projet peut demander la mise en oeuvre de classes et d'interfaces non-vues en cours ou en TD. De plus, l'utilisation de classes déjà développées et mises au point dans la bibliothèque Java permet d'éviter de dépenser du temps à redévelopper ces classes.

Vous êtes donc invités à vous auto-documenter et, plus particulièrement, avec les références suivantes :

<http://java.sun.com/j2se/1.4.2/docs/api/index.html> auto-documentation décrivant les classes livrées avec la distribution de Java.

<http://java.sun.com/docs/books/tutorial/> initiations aux technologies de Java et, plus particulièrement,

<http://java.sun.com/docs/books/tutorial/uiswing/index.html> initiations à Swing.

2 Le projet

2.1 Description

Ce projet a pour objectif de réaliser un jeu informatique nommé « Mice'n' Cats ». Ce jeu affiche un décor en deux dimensions découpé en cases où évoluent des souris et des chats. Des murs permettent de fermer le décor et de donner une configuration spécifique pour le déplacement des chats et des souris.

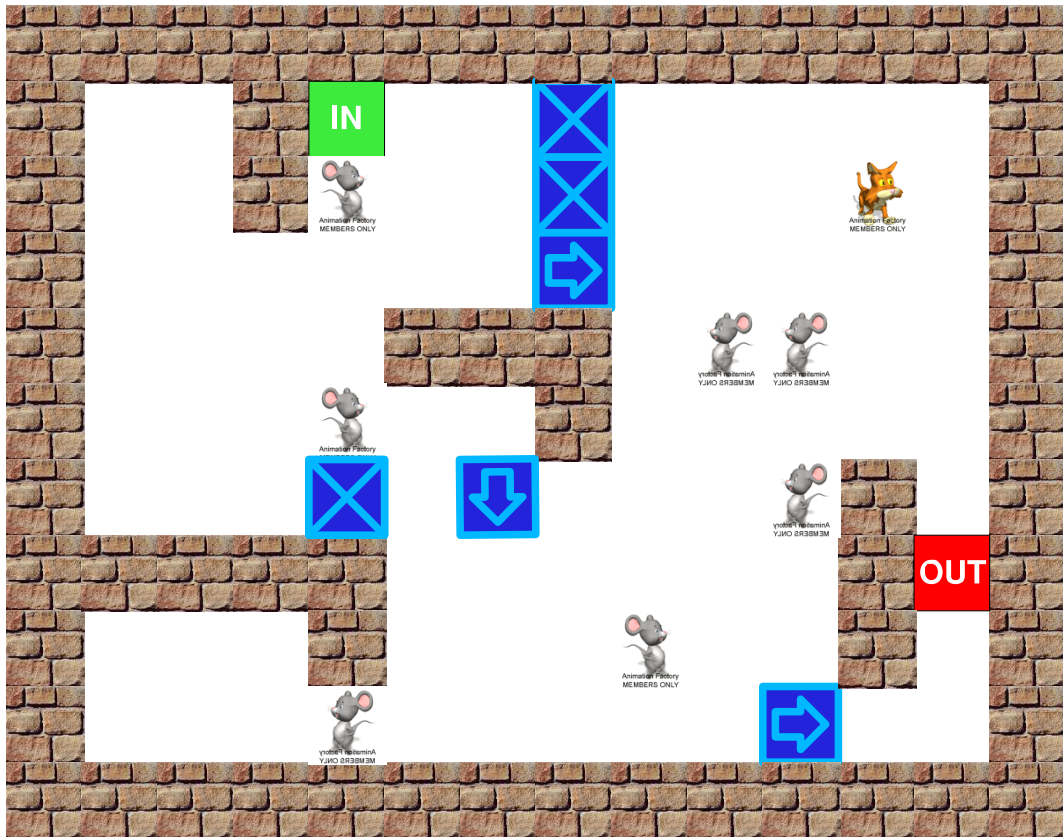


FIG. 1: Exemple de partie en cours.

Le ou les chats sont présents dans le décor à des positions spécifiques au démarrage du jeu. Les souris entrent par une case particulière appelée « entrée » et se déplacent dans le décor jusqu'à trouver une case « sortie ».

- Les souris se déplacent selon l'algorithme suivant :
- elles entrent sur le décor avec une direction prédéfinie (il n'y a que quatre directions : nord, sud, est, ouest) ;
 - lorsqu'elles ne peuvent plus avancer en raison de la présence d'un mur, une direction différente et acceptable est choisie au hasard ;
 - si une souris rentre en collision avec chat, c'est-à-dire, arrive sur la même case, elle est dévorée et retirée du jeu.

Le joueur humain n'intervient pas directement sur les éléments animés du jeu mais peut changer la topologie du décor. Il peut ajouter des murs ou des cases spéciales permettant de modifier la trajectoire des souris. Les cases ajoutées sont en nombre limité et défini au début du tableau. Les murs initiaux du

décor ne peuvent pas être retirés. Les murs et les cases spéciales ajoutées en cours de partie peuvent être retirées et reviennent dans ce cas dans le lot des cases disponibles.

Un décor ou « tableau » est gagné lorsque le joueur a réussi à faire sortir un certain nombre de souris en un temps donné. Le nombre de souris entrant sur le tableau peut être lui aussi limitées. Lorsqu'un tableau est gagné, le jeu passe au tableau suivant. Les tableaux s'enchaînent selon une difficulté croissante : les tableaux sont fournis par le maître d'oeuvre.

A la fin du jeu, c'est-à-dire à la fin des tableaux ou lorsque le joueur a perdu, son score est évalué et stocké avec son nom s'il fait partie des dix meilleurs scores. Le score est évalué en partant d'une base fournie dans la description du tableau modifié par :

- une soustraction de points selon le nombre de souris dévorés,
- un ajout de points lorsque le joueur termine avant le temps limite (s'il y en a un).

2.2 Fonctionnalités

Tous les éléments du jeu sont représentés de manière graphiques à partir des images fournis avec cette description du projet (<http://marine.edu.ups-tlse.fr/M1-TC1-TP>).

Les cases spéciales sont les suivantes :

- des cases de changement de direction (lorsque la souris pénètre dans cette case, sa direction est changée) ;
- des cases de protection (dans lesquelles le chat ne peut entrer) ;
- des cases de téléportation en lien deux à deux.

D'un démarrage à l'autre, les dix meilleurs résultats doivent pouvoir être retrouvés et sont donc stockées dans un fichier.

Le joueur doit pouvoir réaliser les actions suivantes :

- démarrer une partie ;
- arrêter une partie ;
- mettre en pause ;
- reprendre une partie ;
- sélectionner l'action en cours, c'est-à-dire, le type de case qu'il va poser ;
- supprimer une case posée.

Il est clair que, quelle que soit la machine exécutant le programme, ce dernier doit respecter une vitesse adaptée au joueur humain. De même, il est important que l'utilisation du jeu respecte des minimums ergonomiques : réponse fluide aux interactions avec le joueur, affichage synchrone avec l'état du jeu, etc.

Les tableaux doivent être chargés à partir de fichiers qui sont décrits en ASCII et ont le format de l'algorithme 1.

Les nombres sont soit des valeurs entières décrites en décimal ou la valeur « -1 » pour indiquer qu'il n'y a pas de limite.

La *description du decor* est un tableau de la *largeur* donnée (une ligne de texte par ligne du tableau) et de la *hauteur* donnée où les cases sont représentées par des caractères :

- ' ' une case libre,
- 'C' une case libre avec un chat (position initiale du chat),
- 'E' l'entrée des souris,
- 'S' la sortie des souris,
- '#' un mur.

Algorithm 1 Format des fichiers de description de tableau.

[*nom du tableau*]
[*largeur*]
[*hauteur*]
[*duree en secondes*]
[*nombre de souris entrantes*]
[*nombre de souris gagnantes*]
[*nombre de murs*]
[*nombre de teleportations*]
[*nombre de protections*]
[*nombre de direction nord*]
[*nombre de direction sud*]
[*nombre de direction est*]
[*nombre de direction ouest*]
[*base du score*]
[*perte par souris devorees*]
[*gain de point par seconde gagnee*]
[*ligne₁ de la description du tableau*]
[*ligne₂ de la description du tableau*]
...

2.3 Contraintes d'implantation

Lors de l'implantation du projet « Mice'n'Cats », un certain nombre de contraintes devront être satisfaites :

- il doit être programmé en Java ;
- il doit être réalisé de manière multi-threadée (au moins 1 thread par souris et par chat) ;
- il faut maximiser la ré-utilisabilité des composants ;
- le découpage en classes doit être logique et compréhensible ;
- les sources doivent être lisibles et commentés (style Javadoc).

2.4 Extensions

Si la réalisation prend moins de temps que prévu par les TP, il est possible d'ajouter des extensions au jeu initial (ces extensions ne sont pas imposées mais peuvent améliorer l'appréciation du sujet) :

- vitesse différente des souris et de chats et cases d'accélération ou de ralentissement,
- génération chez les souris (elles arrivent jeune puis deviennent adultes),
- gestion du fromage (permettant d'accélérer une souris ou de la faire devenir adulte),
- souris intelligentes (elles détectent si le chat est proche et elles essayent de s'enfuir, lorsqu'une se fait dévorer alors elles préviennent les autres souris qui évitent cet endroit),
- chat intelligent (lorsqu'il dévore une souris, il se déplace plutôt dans la direction d'où elle vient).