

Projet - Langage Orienté Objet et I.A.

L'objectif du projet est de concevoir une application Java permettant de jouer au jeu de plateau **Finstere Flure** de Friedeman Friesel¹. Ce jeu est également jouable en ligne depuis le site www.boiteajeux.net.

1. Présentation du jeu

(Source : <http://www.trictrac.net/jeu-de-societe/finstere-flure/regle>)

Synthèse :

2 joueurs

1 plateau de jeu (labyrinthe) + 17 pions « éléments de décors »

8 cartes « pierre tombale déplacement monstre »

1 jeton « premier joueur » + 25 pions « personnages »

Présentation du jeu

Finstere Flure ("Sinistres Couloirs") est une course dans laquelle les joueurs doivent s'échapper vivants d'un donjon en un maximum de 14 tours.

Chaque joueur possède 3 ou 4 pions personnages, marqués chacun avec les 2 faces d'un dé : 1-6, 2-5 et 3-4. Pendant un tour, une face sera utilisée puis le pion sera retourné et l'autre face sera utilisée le tour suivant. Tous les joueurs ont donc la même capacité de mouvement.

Le donjon est conçu de telle façon qu'un personnage peut s'échapper en 7 tours - si tout se passe tranquillement. Malheureusement (ou par chance !), ça ne se passera pas comme ça : un Monstre affamé se balade dans le coin, prêt à manger tout ce qui se trouve à portée de main. Puisqu'il bouge selon des règles bien établies, les joueurs doivent se positionner de façon à ce que le monstre ne vienne pas vers eux mais plutôt vers les autres.



Figure 1 - Le plateau de jeu : le point de départ se situe en bas à droite et la sortie en haut à gauche.

¹ <http://www.trictrac.net/jeu-de-societe/finstere-flure/infos>

Les règles du jeu sont présentées plus précisément dans le fichier « FinstereFlure.pdf » en pièce jointe sur Arche. Toutefois, pour les besoins du projet, **le jeu ne pourra se jouer qu'à 2 joueurs**. Les images et illustrations du jeu sont également présentes sur Arche.

2. Consignes pour le projet

Ce projet est à réaliser **par groupe de 3 étudiants**. Le travail est à rendre pour le **vendredi 10 mai à 23h59 sur Arche**. Il s'agit d'un projet assez long, n'attendez pas la dernière minute pour vous y mettre. Vous devrez rendre le **code source commenté** (le dossier du projet NetBeans compressé), ainsi qu'un **rapport** pour le cours de Java justifiant vos choix d'implémentation et un **rapport d'analyse distinct** pour le cours d'I.A. d'Anne Boyer. Ne négligez pas les rapports, ils comptent dans la note du projet. Les **soutenances** auront lieu le mardi 14 mai à partir de 8h (10 minutes de présentation du projet Java + 5 minutes de présentation pour la partie I.A. + 5 minutes de démonstration + 10 minutes de questions). **Un rendu intermédiaire (code source du défi 1 ci-dessous) est attendu pour le 22 avril.**

Une attention particulière sera portée à la **qualité du code** (choix des structures de données adaptées, réutilisation des classes vues en cours, portée des variables appropriée private/protected/public, structure des classes, utilisation de l'héritage et des interfaces en Java, etc.). En outre, **la note sera également déterminée en fonction du niveau du défi que vous atteindrez**. Lorsque vous terminez un défi, faites une sauvegarde de votre projet pour éviter de faire planter tout votre programme en tentant de réaliser le défi suivant.

Le défi 2 est obligatoire et en lien avec le cours d'Intelligence Artificielle d'Anne Boyer (approches symboliques et numériques) et sera évalué dans le cadre de son module. L'objectif est de mettre en application les notions vues dans son cours. Si vous n'avez pas été en mesure de l'implémenter complètement, vous devez *a minima* produire un rapport d'analyse (proposition d'une solution, description de l'espace d'états, explications de l'algorithme, etc.).

Les défis 3 et 4 peuvent être réalisés dans n'importe quel ordre.

- **Défi 1 (minimum attendu) :** mettre en place la partie modèle dans votre programme Java, i.e. implémenter toute la mécanique du jeu sans l'interface graphique. Dans cette version du jeu, les informations sont affichées dans la console sous forme de texte et les joueurs saisissent les déplacements au clavier. Dans cette version, on considérera qu'il n'y a que des joueurs humains.

La saisie et l'affichage dans la console sont facultatifs si vous arrivez à réaliser le défi 3 avec interface graphique. Vous pouvez néanmoins prévoir les deux alternatives dans votre programme en laissant la possibilité de lancer le programme avec ou sans interface graphique (ce qui vous octroiera un bonus).

- **Défi 2 (obligatoire pour ceux qui suivent le cours d'Anne Boyer) :** la version 1 + la possibilité qu'il y ait un joueur géré par l'ordinateur. Ce dernier sera doté d'une intelligence artificielle et devra essayer de gagner à tout prix. Outre le code source, vous expliquerez dans votre rapport et lors de la soutenance la stratégie mise en œuvre par votre programme pour gagner (Ex. : parcours de graphe pour une recherche du plus court chemin, gain attendu dans chaque état, algorithme minimax, élagage alpha-beta, heuristiques, etc.) et la représentation des données. Un joueur ordinateur peut également chercher à mettre en défaut un joueur adverse en attirant le monstre vers lui, ou avoir une

stratégie plus ou moins risquée. N'oubliez pas qu'il vaut parfois mieux faire simple, que ne pas arriver à finir.

- **Défi 3 :** la version 2 + une (belle) interface graphique. Vous pouvez vous inspirer de l'interface que vous trouverez en ligne sur boiteajeux.net et essayez de la reproduire en Java, ou bien définir votre propre interface graphique du moment que tous les éléments nécessaires au jeu sont représentés. Dans cette version, les interactions ne se font plus via la console, mais via l'interface graphique. Vous devriez normalement tous être capables d'atteindre ce niveau, après les TD7, 8 et 9. Dans cette version, on suppose que les joueurs jouent sur le même ordinateur. **Indice :** pour pouvoir « poser » des pions sur le plateau, vous pouvez soit découper l'image du labyrinthe et positionner les cases une par une, ou superposer les images (une image pion par-dessus l'image du plateau) grâce à la classe [JLayeredPane](#).
- **Défi 4 :** la version 3 + l'utilisation d'une base de données MySQL pour gérer l'historique des joueurs (nom, prénom, avatar, nombre de parties gagnées, etc.). Il faudra donc créer et installer la base et les tables appropriées, faire communiquer votre programme avec la base en JDBC (cf. aide fournie sur Arche). Adaptez votre interface pour créer un compte utilisateur, afficher les informations sur le joueur, etc.

Pour réussir ce projet, il faut se poser les bonnes questions : Comment représenter intelligemment les données dans votre programme ? Quelle structure adopter (les variables, les classes nécessaires, héritage, etc.) ? Votre rapport devra faire état de vos réflexions et de vos choix sur ce projet.

Dans le rapport du cours de Java, il doit y avoir une page de garde (avec vos noms et prénoms, année universitaire, logo de l'IDMC, nom du cours, nom de l'enseignant), le descriptif des fichiers et des classes (éventuellement accompagné d'un diagramme de classes), la présentation et le fonctionnement de votre application (notamment comment l'installer et le lancer), la présentation de votre interface graphique (avec des captures d'écran) et un rappel (synthèse du rapport du cours d'I.A.) de la stratégie adoptée par votre programme d'intelligence artificielle (joueur ordinateur) pour gagner.

Un forum d'échange, divers documents utiles au projet et les images du jeu sont disponibles sur Arche dans la section projet.