

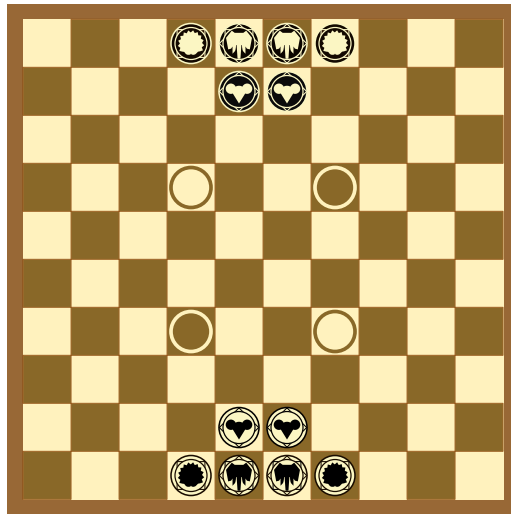
Projet de Programmation en Java master BI 2018-2019

Barca

1 Barca

Barda est à deux joueurs (blanc, noir). Il se joue sur un damier de taille 10×10 . Chaque joueur dispose de trois sortes de couples de pièces de sa couleur : éléphants, lions et souris. Autour du centre du plateau se trouvent 4 cases spéciales marquées par des cercles, les points d'eau.

En début de partie, les pièces sont disposées sur le damier de la manière suivante :



Le but du jeu est de parvenir à placer trois de ses pièces sur les points d'eau. Les joueurs jouent à tour de rôle, en commençant par les blancs. Les règles de déplacement et leurs restrictions sont détaillées à la section suivante.

1.1 Règles du jeu

Règles générales. Chaque pion peut être déplacé d'un nombre arbitraire de cases dans une même direction, à condition qu'il ne passe au-dessus d'aucun autre pion, ami ou ennemi.

- Les souris ne peuvent se déplacer qu'horizontalement ou verticalement.
- Les lions ne peuvent se déplacer qu'en diagonale.
- Les éléphants peuvent se déplacer dans toutes les directions.

Restrictions des déplacements. Une pièce est effrayée par une pièce adjacente de l'autre camp en suivant les règles suivantes :

- Les lions effraient les souris adverses.
- Les souris effraient les éléphants adverses.
- Les éléphants effraient les lions adverses.

Il est interdit à une pièce de se déplacer vers une case où elle deviendrait *effrayée* en partant d'une case où elle ne l'est pas.

Déplacements imposés. Lorsqu'une ou plusieurs pièces sont ou deviennent effrayées après le coup de l'un des deux joueurs :

- Le joueur suivant ne peut déplacer que l'une de ses pièces effrayées.
- Si un déplacement au moins permet de déplacer une pièce effrayée vers une case où elle ne l'est plus, le joueur doit jouer l'un de ces déplacements.

2 Coeur du projet

Le travail que nous vous demandons de réaliser est une implémentation du jeu Barca en java. Le jeu n'est pas très complexe en lui-même, et nous vous demandons de produire un programme raisonnablement bien conçu et utilisant les possibilités de la programmation orientée objets Java. Votre programme devra implémenter les fonctionnalités suivantes :

1. **Il devra permettre de jouer à deux joueurs**, chaque joueur proposant à tour de rôle un déplacement à la machine. Le plateau de jeu devra être affiché (en ASCII) au début du jeu et réaffiché après chaque coup joué, au moins sous la forme d'un texte dans la fenêtre du shell. Le programme devra bien sûr vérifier que chaque déplacement proposé est conforme aux règles du jeu. Si nécessaire, il devra indiquer pourquoi ce déplacement est invalide, et en redemander un nouveau. Il devra également détecter la fin de la partie, et déterminer dans ce cas quel est le joueur gagnant.
2. **Il devra permettre de jouer contre l'ordinateur**, ce dernier jouant éventuellement avec une stratégie simpliste (*eg.* choix aléatoire parmi tous les coups possibles) ou plus élaboré (*cf.* les extensions), mais toujours conforme aux règles du jeu.
3. Le jeu se prête idéalement à la conception et la réalisation d'une interface graphique. Les outils nécessaires pour cette extension n'ont pas encore été présentés en cours, mais sont suffisamment simples en Java pour que vous puissiez développer cette interface graphique dès maintenant en utilisant la documentation Java standard. On vous conseille d'utiliser l'interface graphique SWING, mais ce choix est à votre convenance.

2.1 Extensions

De nombreuses extensions et variantes du jeu sont envisageables. Les 2 premières sont obligatoires, la troisième est optionnelle :

1. **Variantes du jeu.** (obligatoire) Les éléments suivants du jeu pourront être choisis par l'utilisateur au lancement du programme, ou au début d'une nouvelle partie :
 - dimensions et forme du plateau,
 - configuration initiale des pions, parmi un ensemble de configurations,
 - nouvelles sortes de pièces, nouvelles règles pour ces pièces, etc.
2. **Historique de partie et sauvegarde.** (obligatoire) Votre programme sera capable de garder en mémoire la suite de coups d'une partie (complète ou non), d'annuler un ou plusieurs coups parmi les derniers joués, de sauvegarder et de recharger cet historique – avec un encodage quelconque, ou encore, sous la forme d'un texte décrivant la suite des coups à l'aide d'une notation claire et complète – de dérouler son contenu étape par étape à la demande de l'utilisateur, de le dérouler à rebours, d'aller à sa fin ou à son début, etc.
3. **Intelligence artificielle avancée.** (optionnel) Il est possible pour l'ordinateur de faire beaucoup mieux qu'un choix de coup totalement aléatoire, en explorant l'arborescence des coups possibles jusqu'à une certaine profondeur¹.

3 Organisation du projet

Le projet est à faire en groupes de deux personnes au plus. Votre code doit évidemment être strictement personnel : il ne doit s'inspirer ni de code trouvé sur le web, ni de celui d'un autre groupe, ni d'une quelconque "aide" trouvée sur un forum. Votre rendu consistera en un fichier *jar* (cf par exemple <https://docs.oracle.com/javase/tutorial/deployment/jar/index.html>) portant obligatoirement le nom *nom.jar* si vous êtes en monôme ou *nom_1-nom_2.jar* si vous êtes en binôme :

- ce fichier devra pouvoir être exécutable par un `java -jar`
- le code-source de votre programme (fichiers *.java*) sera aussi dans votre fichier
- un rapport de quelques pages en pdf décrivant vos choix de conception et d'implémentation, les extensions réalisées, les algorithmes non triviaux utilisés, les problèmes non résolus, tout autre élément non immédiat, etc. figurera aussi dans cette archive.

Les dates de rendu et de soutenance seront précisées ultérieurement sur la page du cours (les soutenances seront début janvier).

1. Voir par exemple https://fr.wikipedia.org/wiki/Algorithme_minimax

L'évaluation du projet tiendra compte (bien sûr !) de la qualité de la conception de votre solution en particulier dans l'organisation en classes de votre projet.