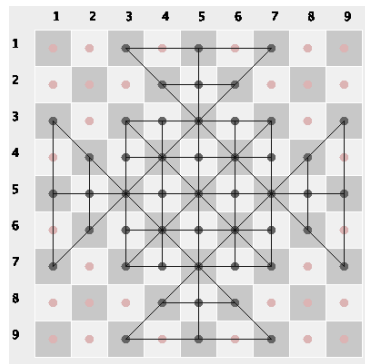


16 Soldats

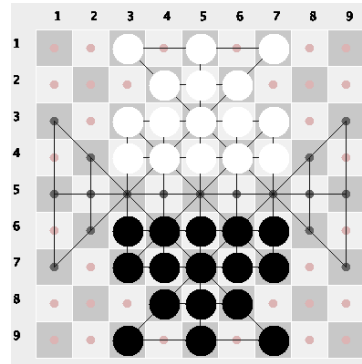
Intelligence Artificielle

Modalités pratiques : Si vous trouvez des références ou des programmes permettant de jouer à ce jeu, vous pouvez vous en inspirer (sans pour autant recopier le code), mais vous devez très clairement indiquer dans votre rapport de devoir les sources et la bibliographie que vous utilisez.

L'intégralité du projet est à faire **exclusivement** en binôme, et à deux. Contactez moi au plus vite si cette modalité pose problème.



(1) Déplacements possibles



(2) Plateau de départ

Règles du jeu

Le jeu **16 soldats** est une adaptation du jeu d'Alquerque, ancêtre des dames. Les pions suivent des lignes de direction très particulières, qui caractérisent cette famille de jeux. Vous trouverez de nombreux sites web qui peuvent préciser les règles du jeu qui vous sembleraient floues de prime abord, mais attention, nous allons nous baser ici sur une version simplifiée. En cas d'ambiguïté sur les règles, prenez contact par e-mail avec J. Azé (aze@lri.fr) pour vous les faire préciser. L'arbitre vous étant fourni, vous pourrez vérifier l'adéquation entre votre joueur et l'arbitre officiel du tournoi. On trouvera notamment une référence au jeu sur <http://jeuxsoc.fr/jeu/16sol>.

- Le jeu se joue sur un damier 9×9 , en suivant les lignes de déplacements comme précisé sur le dessin. La position initiale des pièces est montrée sur la figure (2). Les Blancs commencent à jouer.
- L'objectif du jeu est de capturer toutes les pièces de l'ennemi.
- Toutes les pièces se déplacent d'une case à une case voisine, connectée par un segment. Contrairement aux dames, on n'impose jamais ni d'avancer ni de reculer.
- Une pièce est capturée lorsque un pion saute par dessus, mais toujours en suivant la direction initiale du déplacement, en poursuivant la ligne dans la même direction. Le déplacement est donc en ligne droite, de deux segments de la même ligne. On ne peut capturer qu'une seule pièce par tour. Il n'y a pas de prises multiples (comme par exemple on peut en trouver aux dames). Si un pion saute au dessus d'un pion ennemi, ce dernier est obligatoirement capturé.
- Il n'y a pas de restrictions de coups à la fin du jeu (comme on peut en trouver lorsqu'il ne reste que 5 pions par exemple, où l'on voit parfois des variantes qui obligent les pions à avancer).
- Il n'y a pas de prise de pion obligatoire (comme on peut en trouver aux dames). Vous avez le droit de ne pas faire le mouvement de capture d'un pion, si un mouvement sans capture est possible.
- Si l'un des deux joueurs ne peut plus bouger, la partie est déclarée "nulle" immédiatement.

Autres modifications des règles initiales

Le plateau est numéroté de 1 à 9 pour les colonnes, et de 1 à 9 pour les lignes, en commençant la numérotation dans le coin supérieur gauche du damier. Les blancs seront en haut et les noirs en bas. Un mouvement s'écrit en quatre caractères (les deux caractères du pion à bouger suivi des deux caractères de sa case destination). On utilise la notation colonne × ligne.

La version finale de votre programme aura uniquement 10 minutes de temps réel pour jouer tous ses coups. Si vous dépassez ce temps total de réflexion, vous serez disqualifié pour la partie.

Si une partie dure plus de 200 coups, la partie est déclarée nulle à l'issue du deux centième coup.

Votre IA ne doit pas écrire sur disque (elle peut cependant lire). Elle ne doit pas non plus consommer de temps CPU lorsque ce n'est pas son tour de jouer. Votre IA serait disqualifiée du tournoi si elle ne respectait pas cela.

Rendu du projet

À rendre avant le dimanche 23 juin, 20h (strict)

Tout rendu au delà de cette date ne sera pas pris en considération car le tournoi aura déjà été lancé.

Vous devrez rendre vos codes sources et une version compilée permettant d'être interfacée avec l'arbitre. Vous devez écrire votre joueur en Java et ne pas dépasser la version 1.7.

L'interfaçage de votre programme avec les autres se fait par l'intermédiaire d'un arbitre qu'il faut contacter via le réseau, qui vous est donné via le site web du cours. Le programme et le code Java de votre programme seront aussi à rendre. Une classe particulière, `Solo` vous permettra de simuler une partie via le réseau, sans le réseau (et sans l'arbitre), pour faciliter la mise au point de votre IA. Un joueur aléatoire vous est également donné avec le serveur jeu réseau, pour tester votre IA en conditions réelles de tournoi, et pour pouvoir tester votre IA contre celles d'autres groupes. Tous les détails d'implantation et d'interfaçage de votre code avec l'arbitre sont donnés sur le site du cours,

`http://www.lri.fr/~aze/enseignements/App5`

Modalités de rendu du projet

Le projet est à rendre par email adressé à `aze@lri.fr`. Le sujet de votre email doit être de la forme [App5 - 16soldats] Nom des membres du binôme.

Votre email doit contenir les pièces jointes suivantes :

- L’archive de vos sources Java et le rapport (en PDF) ;
- Un répertoire contenant les binaires java (ainsi que `ClientJeu`, `IJoueur`, ...) accompagné de la ligne de commande exacte que l’on doit lancer pour votre client réseau (notamment le nom de votre classe implantant l’interface `IJoueur` que l’on doit lancer lors du tournoi). Sans les binaires Java et sans les explications, votre joueur sera disqualifié.

La moitié des points sera donnée sur le rapport uniquement, pour peu que celui-ci reflète fidèlement les choix effectués dans le code source.

Quelques consignes pour le rapport

Dans le rapport, vous préciserez notamment les raisons de vos choix d’implantation et de vos choix d’heuristiques. Vous préciserez la solution adoptée pour la recherche du meilleur coup à jouer ainsi que toutes les astuces que vous avez implanté (structures de données, calcul heuristique efficace, bibliothèque d’ouverture éventuelle, ...), sans oublier les solutions par rapport à la gestion du temps réel et l’analyse finale, les tests effectués et, en conclusion, les difficultés rencontrées dans ce devoir.

Faites un document concis et simple à lire, mais informatif et précis. À titre indicatif, entre quinze et vingt pages maximum semble raisonnable.

Vous pourrez discuter les différents points suivants :

1. Y-a-t-il des possibilités de parties nulles dans ce jeu, sans tenir compte de la limite des 200 coups ?
2. Quel est le facteur de branchement du jeu ? Vous donnerez une estimation que vous justifierez.
3. Estimez l’ordre de grandeur d’une partie typique.
4. À quelle profondeur peut-on envisager de lancer un Minimax ? Un $\alpha - \beta$? Pour cela, vous vous baserez sur une estimation de la taille des arbres de recherche.
5. Proposez au moins deux fonctions heuristiques pour ce jeu. Laquelle avez-vous intégré au jeu et pourquoi ?