

RAPPORT DE PROJET IA

PROJET : Dames Anglo-Américaines

GROUPE : LATTAB Nassim, IDRISSE-MACHICHI Ali

Sommaire

1. Introduction	2
2. Rappel des règles du jeu	2
2.1. But du jeu	2
2.2. La prise	2
2.3. Promotion Dame.....	2
2.4. Match Nul	2
2.5. Gagner une partie	2
3. Implémentation des IAs	3
3.1. Fonction d'évaluation	3
3.2. Minimax et profondeur de recherche	3
3.3. Élagage Alpha-Beta	4
3.4. Difficultés : Simple, Intermédiaire, Difficile	4
4. Résultats statistiques	5
4.1. IA facile vs IA intermédiaire	5
4.2. IA facile vs IA difficile	5
4.3. IA intermédiaire vs IA difficile	5
4.4. Analyse des résultats	6
5. Conclusion	6
6. Sources	6

1. Introduction

Dans le cadre de l'UE Intelligence Artificielle de la troisième année de licence Informatique et Applications à l'Université Paris Cité, nous avons été amenés à concevoir et développer un jeu tour par tour à deux joueurs et à connaissance parfaite. Ce jeu a pour objectif de proposer des parties joueur humain contre joueur artificiel, et doit permettre de pouvoir choisir entre 3 difficultés différentes. Pour ce projet, notre choix s'est porté sur une variante du jeu de dames : les dames Anglo-Américaines.

Dans ce document, nous tenterons d'expliquer le plus clairement possible le cheminement que nous avons suivi afin de parvenir à nos objectifs, en prenant soin de détailler les résultats statistiques obtenus.

2. Rappel des règles du jeu

Le jeu de dames Anglo-Américaines se joue sur un plateau de 64 cases, alternativement foncées et claires, disposées en damier. Les joueurs disposent chacun de 12 pions de couleur claire ou foncée, qui sont placés sur les cases noires des trois rangées les plus proches de chaque joueur.

2.1. But du jeu

Le but du jeu est de capturer tous les pions de l'adversaire ou de bloquer ses pions de sorte qu'ils ne puissent plus bouger. Les pions peuvent se déplacer en diagonale sur les cases adjacentes, mais uniquement vers l'avant. On se déplace donc exclusivement sur les cases foncées.

2.2. La prise

Un pion peut prendre (ou "manger") un pion adverse en sautant par-dessus lui diagonalement, vers une case libre derrière le pion adverse. La prise multiple est possible.

Attention : La prise est obligatoire. Cela signifie que le joueur qui joue à l'obligation de prendre si cela est possible.

2.3. Promotion Dame

Si un pion atteint la dernière rangée de l'adversaire, il est promu en dame. Les dames ont le droit de se déplacer en diagonales dans toutes les directions (d'une case maximum), nous n'avons pas implémenté la possibilité de rendre la dame "volante", c'est-à-dire qu'elle ne peut pas sauter par-dessus plusieurs cases à la fois.

2.4. Match Nul

Nous avons imposé une limite de coups aux IAs de sorte que les parties ne soient pas interminables. Si la limite est dépassée et qu'il n'y a eu aucune prise, la partie est déclarée nulle (arrive relativement rarement). Sinon, à chaque nouvelle prise le compteur se réinitialise.

2.5. Gagner une partie

Dans notre jeu, il existe deux manières de gagner une partie. La première façon est de capturer tous les pions adverses. La seconde manière de gagner est indirecte : Si l'adversaire ne peut plus jouer de coup légal, il perd.

3. *Implémentation des IAs*

L'une des principales difficultés de ce projet était tout d'abord d'implémenter une intelligence artificielle capable de prendre une décision afin de déterminer le meilleur prochain coup à jouer en fonction des informations disponibles et de sa capacité de calcul. Une fois cela fait, il fallait ensuite créer des IAs de difficulté différente, en faisant appel à des notions de stratégies afin de renforcer leurs compétences. En effet, les IAs les plus avancées peuvent utiliser des stratégies plus complexes, ce qui leur donne un avantage sur les IAs plus faibles et sur les joueurs humains moins expérimentés. L'implémentation des difficultés implique de bien comprendre les règles et les mécanismes du jeu afin d'analyser et de déterminer le meilleur état possible en fonction de différents facteurs et situations à prendre en compte.

3.1. *Fonction d'évaluation*

En intelligence artificielle, une fonction d'évaluation est une méthode utilisée pour évaluer un état ou une configuration donnée dans un problème donné. Dans notre cas, elle est utilisée dans l'algorithme minimax que nous avons implémenté.

L'objectif de la fonction d'évaluation est de donner une valeur numérique à un état, qui peut être utilisée pour comparer cet état avec d'autres états possibles et déterminer quelle est la meilleure action à effectuer à partir de cette situation.

Globalement, plus la fonction d'évaluation prend de facteurs en compte, plus l'IA sera apte à prendre la meilleure décision possible. Dans notre cas, elle doit être capable de fournir des informations précises et utiles sur la position des pièces sur le plateau, afin que l'IA puisse prendre des décisions éclairées sur les mouvements à effectuer. De plus, il peut être intéressant d'ajouter des coefficients à nos facteurs afin de leur donner un poids plus ou moins important. Une fonction d'évaluation plus sophistiquée permettra à l'IA de mieux comprendre les configurations complexes et les interactions entre les pièces sur le plateau, et donc de prendre des décisions plus avancées.

Voici nos choix de fonction d'évaluation :

- IA facile : Prend en compte la différence du nombre de pion de chaque joueur sur le damier.
- IA intermédiaire : Idem que l'IA facile, prend aussi en compte la différence du nombre de dames sur le damier.
- IA difficile : Idem que l'IA intermédiaire, prend aussi en compte la position des pions (proche de promotion dame, pion bloqués, pion attaquants).

3.2. *Minimax et profondeur de recherche*

L'algorithme Minimax est un algorithme de recherche utilisé en intelligence artificielle pour la prise de décision dans des jeux à deux joueurs à somme nulle, tels que les échecs ou le tic-tac-toe. Son objectif est de trouver le meilleur coup possible pour un joueur à partir d'un état donné. Il suppose que chaque joueur cherche à maximiser son propre gain et à minimiser le gain de l'adversaire.

L'algorithme minimax utilise une profondeur maximale à ne pas dépasser. En effet, en limitant la profondeur de recherche, l'algorithme Minimax peut explorer l'arbre de recherche jusqu'à une certaine profondeur et évaluer les positions atteintes, ce qui permet de déterminer le meilleur coup à jouer dans cette situation. La profondeur de recherche est donc un compromis entre la qualité de la décision prise et le temps de calcul nécessaire pour l'obtenir.

Il est important de noter que plus la profondeur de recherche est élevée, plus l'algorithme Minimax est susceptible de trouver la meilleure décision possible, mais plus le temps de calcul est important. À l'inverse, une faible profondeur de recherche peut entraîner des décisions moins précises mais plus rapidement obtenues. En outre, on peut facilement remarquer que changer la profondeur de recherche aura un impact significatif sur la performance de l'IA.

Voici nos choix de profondeur :

- IA facile : profondeur 1.
- IA intermédiaire : profondeur 3.
- IA difficile : profondeur 5.

3.3. *Élagage Alpha-Beta*

L'élagage alpha-beta est une technique utilisée dans l'algorithme Minimax pour réduire le nombre de nœuds de l'arbre de recherche à explorer, afin d'optimiser les performances de l'IA. L'élagage consiste à éliminer certaines branches de l'arbre de recherche qui sont susceptibles de ne pas mener à une solution optimale, ce qui permet d'économiser du temps de calcul.

Cette technique permet d'éliminer des branches de l'arbre de recherche qui ne sont pas pertinentes en fonction des valeurs minimales et maximales déjà calculées pour les nœuds précédents. Elle fonctionne en comparant les valeurs minimales et maximales calculées pour les nœuds précédents avec celles des nœuds suivants. Si une branche a une valeur minimale ou maximale qui ne peut pas améliorer le résultat déjà obtenu, elle peut être élaguée sans affecter le résultat final.

Dans le cadre de notre projet, nous avons mis en place cette technique afin de minimiser la complexité en temps de notre algorithme de recherche.

3.4. *Difficultés : Simple, Intermédiaire, Difficile*

Il nous était demandé d'implémenter trois difficultés d'IAs différentes, allant de la plus simple à la plus difficile. Afin d'atteindre cet objectif, nous avons donné à chaque IA une fonction d'évaluation et une profondeur de recherche différente. En effet, la complexité de la fonction d'évaluation et la profondeur de recherche sont deux facteurs clés pour déterminer la difficulté de l'IA dans le contexte de notre jeu.

En d'autres termes, la fonction d'évaluation donnée est plus ou moins complexe en fonction de la difficulté de l'IA souhaitée et la profondeur de recherche associée est plus ou moins importante afin de creuser l'écart de difficulté entre nos IAs.

4. Résultats statistiques

Afin de comparer nos IAs et ainsi avoir une meilleure idée de leurs performances afin de les optimisées, nous les avons fait jouer l'une contre l'autre pendant un total de 1000 parties d'affilée. Après avoir modifié certains paramètres afin d'obtenir des résultats bien distincts et cohérents, voici ce que nous avons obtenu :

4.1. IA facile vs IA intermédiaire

```
***** Pions Blancs : IA Intermédiaire VS Pions Noirs : IA Facile *****
Total victoires des blancs: 984
Total victoires des noirs: 16
Nuls par limite de coups : 0
```

Figure 1 - IA facile vs IA intermédiaire

Taux de victoire :

- IA facile : 1,6%
- IA intermédiaire : 98,4%

Matches Nuls :

- Aucun

Note : Les matches nuls sont plutôt rares. Généralement, l'une des IA finira toujours par l'emporter.

4.2. IA facile vs IA difficile

```
***** Pions Blancs : IA Difficile VS Pions Noirs : IA Facile *****
Total victoires des blancs: 997
Total victoires des noirs: 3
Nuls par limite de coups : 0
PS D:\Users\nassim\Desktop\Projets\ProjetIA>
```

Figure 2 - IA facile vs IA difficile

Taux de victoire :

- IA facile : 0,3%
- IA difficile : 99,7%

Matches Nuls :

- Aucun

4.3. IA intermédiaire vs IA difficile

Pour ce cas, nous avons chacun de notre côté lancé 500 parties afin d'optimiser notre temps et ne pas laisser tourner le code trop longtemps. Voici les résultats obtenus :

```
***** Pions Blancs : IA Difficile VS Pions Noirs : IA Intermédiaire *****
Total victoires des blancs: 402
Total victoires des noirs: 98
Nuls par limite de coups : 0
```

```
***** Pions Blancs : IA Difficile VS Pions Noirs : IA Intermédiaire *****
Total victoires des blancs: 410
Total victoires des noirs: 87
Nuls par limite de coups : 3
PS D:\Users\nassim\Desktop\Projets\ProjetIA> █
```

Figure 3 - IA intermédiaire vs IA difficile

Taux de victoire :

- IA intermédiaire : 18,5%
- IA difficile : 81,2%

Matches Nuls :

- 3 nuls : 0,3%

4.4. Analyse des résultats

On remarque que l'IA difficile l'emporte haut la main contre chacune des difficultés. De plus, l'IA facile quant à elle gagne très rarement, ce qui est plutôt cohérent. Les paramètres attribués qui nous ont permis d'arriver à ce bilan ont donc été conservés.

5. Conclusion

Finale^{ment}, les résultats observés sont plus que satisfaisants vis à vis de nos attentes. D'autre part, ce projet nous a permis de développer et mettre en pratique nos compétences en Intelligence Artificielle. De plus, il nous a offert l'opportunité de travailler en équipe sur une problématique concrète. Nous sommes convaincus que les connaissances acquises nous seront précieuses pour nos futurs travaux en Intelligence Artificielle.

6. Sources

Ci-dessous, un lien GitHub vers un projet similaire qui nous a aidé à coder la base du jeu sans IA : <https://github.com/techwithtim/Python-Checkers>

Nous nous sommes inspirés des fonctions et des classes afin d'avoir une base de jeu fonctionnelle que nous avons par la suite optimisé et amélioré afin qu'il ressemble plus à ce que nous voulions faire. Par exemple, nous avons ajouté différentes fonctions afin de modifier certaines règles ainsi que les modes de jeu. Nous avons notamment implémenté la prise obligatoire, le match nul par limite de coup, les méthodes faisant jouer les IAs entres-elles, une méthode pour faire jouer un humain contre une IA en choisissant la difficulté souhaitée, un menu permettant de lancer la partie de notre choix (joueur vs joueur, joueur vs IA, IA vs IA) ainsi que les différentes méthodes nous permettant d'implémenter les IAs.