

# Compte rendu de projet

*Agents intelligents au jeu de dames*

Membres du groupe : Hugo Demaret

Cours : Intelligence Artificielle (L3)



**Université Paris Cité**  
UFR de Mathématiques et Informatique

# Sommaire

<b>1</b>	<b>Jeu de dames</b>	<b>3</b>
1.1	Présentation . . . . .	3
1.2	Quelques exemples . . . . .	4
1.3	Historique et faits . . . . .	4
<b>2</b>	<b>Problèmes rencontrés</b>	<b>5</b>
2.1	Agents . . . . .	5
<b>3</b>	<b>Agents</b>	<b>5</b>
3.1	Cas de l'agent aléatoire . . . . .	5
3.2	Agents intelligents . . . . .	5
3.2.1	Agent glouton . . . . .	5
3.2.2	Agent Alpha-Beta bête . . . . .	5
3.2.3	Agent Alpha-Beta plus évolué . . . . .	5
3.3	Résultats . . . . .	6
<b>4</b>	<b>Améliorations</b>	<b>6</b>
4.1	Utilisation de bit-board . . . . .	6
4.2	Fonctions d'évaluation statiques . . . . .	6
4.3	Fonctions d'évaluation dynamiques . . . . .	7
4.4	Algorithme génétique . . . . .	7
4.5	Ensemble d'ouvertures . . . . .	7
4.6	Fin de partie . . . . .	7
<b>5</b>	<b>Bilan</b>	<b>7</b>

## Préambule

Le programme consiste en deux fichiers python. Le fichier à exécuter est **draughts.py**. Une fois lancé, celui-ci propose soit de voir des agents s'affronter entre eux, soit de jouer. La première option demande ensuite de choisir le niveau des agents, puis la partie s'en suit. La deuxième option demande au joueur s'il souhaite jouer contre un humain ou un agent. Dans le premier cas, le jeu commence et les joueurs peuvent jouer. Les coups valides sont proposés, et le joueur doit juste choisir celui qu'il souhaite faire. Dans le deuxième cas, le jeu demande le niveau de l'adversaire, puis le jeu commence. Le joueur humain commence, et saisi à chaque coup parmi les coups valides le coup qu'il souhaite effectuer. Une fois la partie terminée, le gagnant est affiché. S'il n'y en a pas, le jeu affiche qu'il y a égalité.

# 1 Jeu de dames

## 1.1 Présentation

Le jeu de dames est un jeu de société qui se joue à deux joueurs sur un plateau de 64 cases, alternativement claires et foncées. Chaque joueur possède 12 pions qui sont placés sur les cases noires de la rangée la plus proche de lui. Le but du jeu est de capturer tous les pions de l'adversaire ou de l'empêcher de faire un mouvement valide.

Les pions peuvent se déplacer diagonalement d'une case à la fois, vers l'avant ou vers l'arrière. Les pions peuvent également capturer les pions adverses en sautant par-dessus eux, à condition qu'il y ait une case vide derrière le pion capturé. Si un pion atteint la rangée la plus éloignée du plateau, il est promu en une dame qui peut se déplacer et capturer dans toutes les directions.

Le jeu de dames est un jeu de stratégie passionnant qui peut être joué par des joueurs de tous niveaux. Les règles du jeu sont les suivantes :

- Le jeu se joue à deux joueurs sur un plateau de 8x8 cases, alternativement claires et foncées.
- Chaque joueur possède 12 pions qui sont placés sur les cases noires de la rangée la plus proche de lui.
- Les pions peuvent se déplacer diagonalement d'une case à la fois, vers l'avant ou vers l'arrière.
- Les pions peuvent capturer les pions adverses en sautant par-dessus eux, à condition qu'il y ait une case vide derrière le pion capturé.
- Si un pion atteint la rangée la plus éloignée du plateau, il est promu en une dame qui peut se déplacer et capturer dans toutes les directions.
- Le joueur qui capture tous les pions de l'adversaire ou qui l'empêche de faire un mouvement valide remporte la partie.
- La règle de la capture obligatoire stipule qu'un joueur doit capturer un pion adverse s'il en a l'opportunité. Si plusieurs captures sont possibles, le joueur doit choisir celle qui capture le plus grand nombre de pions. Si un joueur ne capture pas de pion alors qu'il en avait la possibilité, son adversaire peut exiger qu'il effectue la capture manquée ou faire rejouer son tour (dans notre implémentation, on force cette règle).

En résumé, le jeu de dames est un jeu de société classique qui se joue à deux joueurs sur un plateau de 64 cases. Les joueurs déplacent leurs pions diagonalement pour capturer les pions adverses et essayer de promouvoir leurs pions en dames. Le joueur qui capture tous les pions de l'adversaire ou qui l'empêche de faire un mouvement valide remporte la partie.

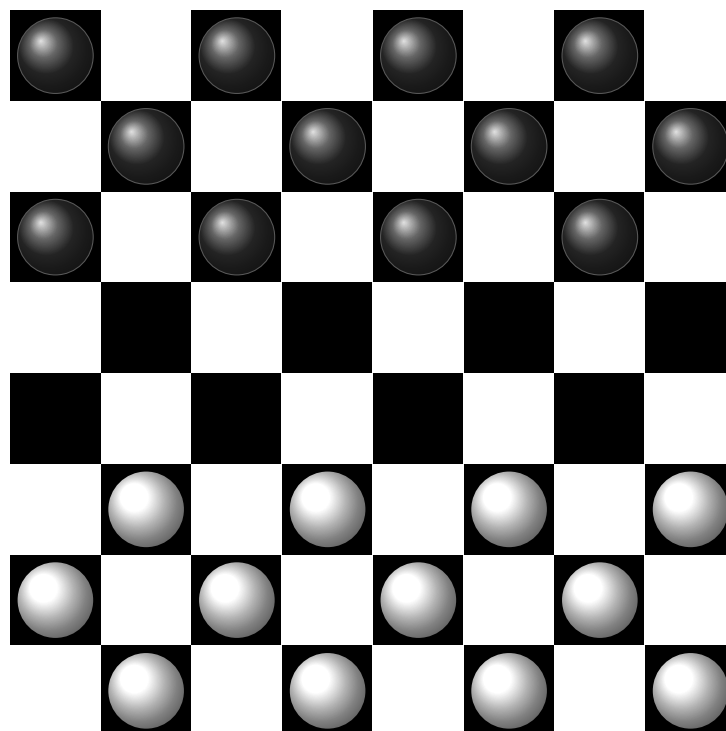


Figure 1: Quelques coups au jeu de dames (plus ou moins intelligents)

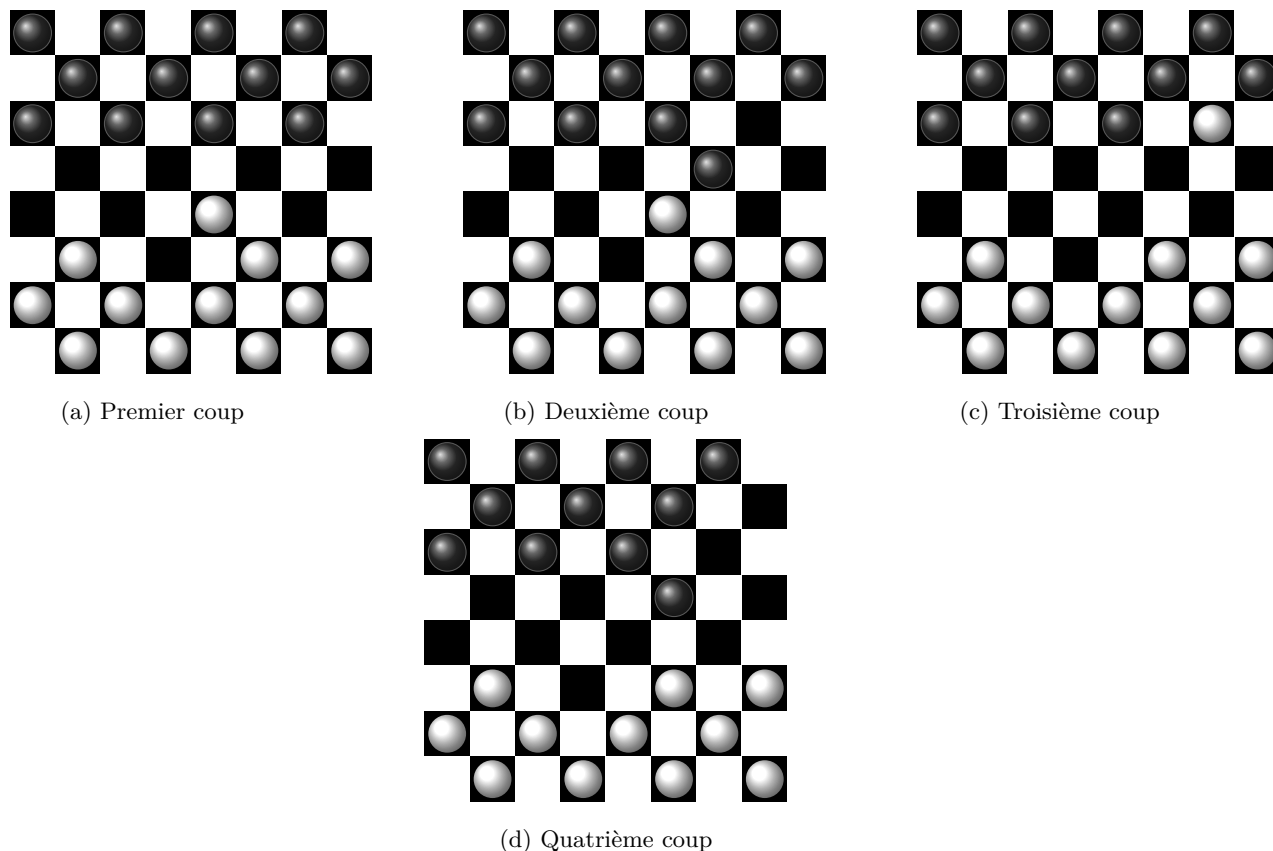


Figure 2: Création d'un roi et ses mouvements

## 1.2 Quelques exemples

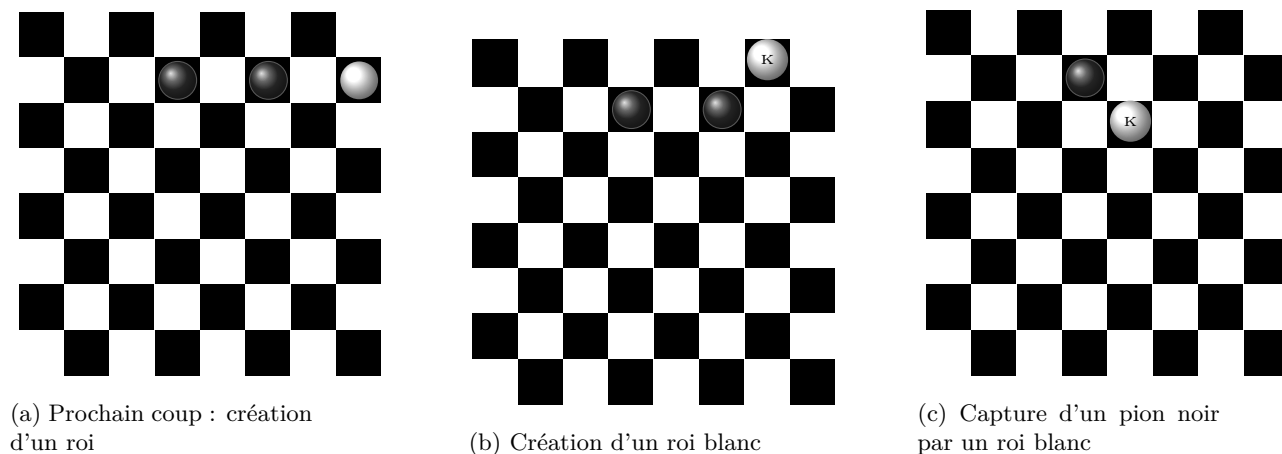


Figure 3: Configuration initiale du jeu de dames

## 1.3 Historique et faits

Le jeu de dames est issu du transfert, au XIV<sup>e</sup> siècle en Europe, de l'alquerque de douze sur un échiquier de 64 cases. Ce sont les Arabes, au Moyen Âge, qui introduisirent l'alquerque dans la Péninsule ibérique. Ce jeu, appelé aussi « marelle de douze », aux règles très proches de celles du jeu de dames, était déjà joué dans l'Égypte antique, vers 1500 av. J.-C.1.

Une fois établi sur un damier à soixante-quatre cases, le jeu diffusa en France, en Angleterre, en Pologne, etc. Une littérature apparaît en Espagne aux XVI<sup>e</sup> siècle et XVII<sup>e</sup> siècle chez des auteurs comme Lorenzo Valls, Juan de Timoneda, Juan García Canalejas et Antonio de Torquemada2.

Les dames ont permis de nombreux avancements dans les intelligences artificielles appliquées aux jeux. En 1951, Christopher Strachey a programmé le premier jeu vidéo de dames. Dans les années 80, et jusqu'à 2007, un programme nommé Chinook était développé afin de jouer de manière efficace aux dames, avec l'objectif de battre l'humain. En 1994, il est devenu champion du monde (du concours Humain-Machine)

aux dames. Après 1995, l'objectif a été de résoudre le jeu de dames, ce qui a été fait en 2007 [1]. Ce papier démontre que si les deux adversaires jouent parfaitement, alors la partie finira toujours par un match nul.

## 2 Problèmes rencontrés

### 2.1 Agents

Dans certains cas, probablement dûs aux fonctions d'évaluations, les agents se mettent à tourner en rond ou à rentrer dans des cycles infinies. Cela est dû au fait que la fonction d'évaluation renvoie un meilleur coup qui fait aller dans un état A, puis dans l'état A la fonction renvoie dans l'état B, et ainsi de suite. Les agents étant déterministes, ils resteront bloqués. Pour cette raison, le jeu s'arrête au bout de 500 parties par une égalité. Toutefois, cela pourrait être corrigé de différentes manières :

- Changer de fonction d'évaluation si on détecte un cycle
- Bloquer temporairement la case si on détecte un cycle (en renvoyant une valeur absurde)
- Ajouter un peu d'aléatoire

Cette liste n'est pas exhaustive.

## 3 Agents

### 3.1 Cas de l'agent aléatoire

Nous avons implémenté un joueur aléatoire, et ce pour deux raisons :

- Tout d'abord, un joueur aléatoire peut être considéré comme le niveau 0 d'adversaire, n'étant pas intelligent, mais n'étant pas bête non plus.
- Cet agent aléatoire servira de mesure pour les autres agents. Si un agent joue mieux que l'aléatoire, alors il est a priori intelligent, tandis que s'il joue moins bien que l'aléatoire, il est bête.

### 3.2 Agents intelligents

Dans cette implémentation du jeu de dames, il y a trois niveaux d'agents intelligents.

#### 3.2.1 Agent glouton

Le premier niveau est un algorithme glouton, qui à chaque coup prend le meilleur possible selon une certaine fonction d'évaluation que voici :

$$\text{Eval} = \# \text{pièces joueur} - \# \text{pièces adversaire}$$

C'est l'agent intelligent le plus basique, mais nous verrons que celui-ci a toutefois de bons résultats.

#### 3.2.2 Agent Alpha-Beta bête

Cet agent utilise l'algorithme MiniMax avec un élagage  $\alpha - \beta$  avec une profondeur de 3. Cette profondeur permet d'avoir quelques coups d'avance. La fonction d'évaluation utilisée est assez simple : c'est la même que pour l'agent glouton c'est à dire :

$$\text{Eval} = \# \text{pièces joueur} - \# \text{pièces adversaire}$$

La profondeur de 3 est suffisante, étant donné la simplicité de la fonction d'évaluation. En effet, après avoir testé différentes profondeurs, on peut voir que 3 est un bon compromis.

#### 3.2.3 Agent Alpha-Beta plus évolué

Cet agent utilise aussi l'algorithme MiniMax avec élagage  $\alpha - \beta$ , cette fois-ci avec une profondeur de 5. Cette profondeur permet de voir plus loin que la moyenne des joueurs humains. De plus, cet agent a une fonction d'évaluation plus sophistiquée que voici :

$$\begin{aligned}
\text{Eval} = & 7 \times (\# \text{pièces joueur} - \# \text{pièces adversaire}) \\
& + 1 \times (\# \text{rois joueur} - \# \text{rois adversaire}) \\
& + \frac{1}{2} \times (\# \text{pièces joueur centre} - \# \text{pièces adversaire centre}) \\
& + 1 \times (\# \text{création rois possible}) \\
& + \frac{1}{2} \times (\# \text{rois à risque})
\end{aligned}$$

### 3.3 Résultats

Joueur b vs. Joueur w	Win b	Lose b	Draw	Matches	Winrate b
Random vs. Random	45	55	0	100	45%
Random vs. AI 1 w	29	65	6	100	29%
Random vs. AI 2	0	100	0	100	0%
Random vs. AI 3	0	10	0	10	0%
AI 1 vs. Random	93	4	3	100	93%
AI 2 vs. Random	70	8	22	100	70%
AI 3 vs. Random	10	0	0	10	100%

Table 1: Résultats Aléatoire IA

On observe que les agents intelligents battent les agent aléatoires dans la plupart des cas. Un comparatif entre les agents intelligents est, dans notre cas, assez inutile : tous sont déterministe et donc les matchs seront toujours les mêmes. Afin de les comparer, il pourrait être utile de prendre différentes parties à différents stades, et de voir comment les agents intelligent jouent sur cette partie déjà commencée. Toutefois, cela n'a pas été fait dans le cadre de ce projet.

Joueur b vs. Joueur w	Winner
AI 1 vs. AI 1 w	b
A2 vs. AI 2	w
A3 vs. AI 3	w
AI 1 vs. AI 2	b (AI 2)
AI 1 vs. AI 3	w (AI 3)
AI 2 vs. AI 1	b (AI 2)
AI 2 vs. AI 3	boucle
AI 3 vs. AI 1	b (AI 3)
AI 3 vs. AI 2	boucle

Table 2: Résultats IA contre IA

## 4 Améliorations

### 4.1 Utilisation de bit-board

L'implémentation actuelle est assez lente, d'une part car elle est en Python, d'autre part car elle est assez mal réfléchi pour la performance. Une implémentation en utilisant un système de bit-board permettrait d'aller beaucoup plus vite – et donc beaucoup plus loin – notamment en profondeur de MiniMax. L'idée est de représenter chaque case par un bit, 0 si elle est inoccupée, 1 sinon. Ensuite, avec un tableau associatif on peut savoir la pièce qui est localisé aux coordonnées. Cela permet de prendre beaucoup moins de place en mémoire. On pourrait aussi stocker certaines choses, sous forme mémoiser, afin d'éviter de recalculer quelque chose déjà calculé. Enfin, on pourrait utiliser la programmation dynamique pour calculer le nombre minimal de coup nécessaire à, par exemple, capturer une pièce adverse.

### 4.2 Fonctions d'évaluation statiques

La fonction actuellement utilisée n'est pas très recherchée. En y passant plus de temps, il aurait été possible de trouver des poids plus efficaces avec les mêmes paramètres. De plus, il est totalement possible d'introduire de nouveaux paramètres, notamment la capture des coins. On pourrait aussi augmenter l'agressivité d'un agent en donnant plus de poids à un mouvement rapprochant de l'adversaire.

### 4.3 Fonctions d'évaluation dynamiques

Parfois, une fonction d'évaluation efficace en début de partie ne l'est plus en milieu ou fin de partie. Il peut être très utile de disposer de plusieurs fonctions d'évaluation, et selon une certaine heuristique (par exemple, le nombre de pièces restantes...), de changer de fonction d'évaluation.

### 4.4 Algorithme génétique

Afin de trouver des poids pour les différents paramètres, il peut être utile de pré-entraîner à l'aide d'un algorithme génétique, les fonctions d'évaluation. Il est également possible de trouver plusieurs fonctions d'évaluations pour différents stades de la partie, afin d'appliquer des fonctions dynamiques.

### 4.5 Ensemble d'ouvertures

Certaines ouvertures peuvent être beaucoup mieux que d'autres, et certaines ne sont pas à faire. La plupart des programmes intelligents, aussi bien aux échecs, Go, et bien entendu aux dames, utilisent un ensemble d'ouverture. Cela permet de bien commencer le jeu, mais aussi d'éviter certains pièges pouvant être tendu par l'adversaire en début de partie.

### 4.6 Fin de partie

Pour beaucoup de jeux de sociétés, il peut être utile d'implémenter un ensemble de mouvements afin d'accélérer les fins de partie. Souvent, au jeu de dames, cela est fait quand il ne reste que 8 pièces.

## 5 Bilan

Les résultats des agents contre l'aléatoire sont assez surprenant. Je regrette en effet de ne pas avoir pu plus m'investir dans ce projet. Je comptais à la base prendre le jeu de Hex, mais pour différentes raisons personnelles, j'ai préféré prendre le jeu de dames (il est en effet plus simple).

Toutefois, modifier les fonctions d'évaluations, et voir sous mes yeux les résultats changer et les agents devenir plus où moins fort, est quelque chose de très intéressant. J'aurais bien aimé tester et implémenter les différentes méthodes abordées dans la section Améliorations, et je pense que ces ajouts auraient été efficaces. Toutefois, le jeu de dames étant résolu, il aurait été plus judicieux, selon moi, de les tester dans un jeu non résolu, comme le Hex.



## References

- [1] Jonathan Schaeffer, Neil Burch, Yngvi Björnsson, Akihiro Kishimoto, Martin Müller, Robert Lake, Paul Lu, and Steve Sutphen. “Checkers Is Solved”. In: *Science* 317 (2007), pp. 1518–1522.