

BUT Informatique

SAÉ 1.02

Approche Algorithmique

Besjan, Anthony, Matthieu

Table des matières

Partie 1 : Introduction	2
En quoi consiste le jeu ?	2
En quoi consiste le projet ?	2
Répartition des tâches	2
Partie 2 : IAs primitives	3
En quoi consistent-elles ?	3
Point de vue critique	3
Partie 3 : L'approche de l'arbre	4
En quoi consiste cette approche ?	4
Choix réalisés lors de sa création	4
Ses avantages	5
Ses inconvénients	5
Qu'en tirer ?	5
Partie 4 : L'approche du score par case	6
En quoi consiste cette approche ?	6
Choix réalisés lors de sa création	7
Ses avantages	7
Ses inconvénients	7
Qu'en tirer ?	8
Partie 5 : L'approche du score par pion	9
En quoi consiste cette approche ?	9
Choix réalisés lors de sa création	11
Ses avantages	11
Ses inconvénients	11
Qu'en tirer ?	11
Partie 6 : Conclusion	12
Que peut-on retenir de ce projet ?	12
Quelle(s) solution(x) retenons-nous ?	12
Merci d'avoir lu !	12

Partie 1 : Introduction

En quoi consiste le jeu ?

Pour cette SAÉ nous avons dû créer un jeu qui mixe le jeu de dames avec celui du jeu d'échecs. En effet, ce jeu se joue sur un plateau de dimensions $n \times m$. Au départ, chaque joueur dispose de sa rangée de pions de son côté.

Par défaut, les pions ne peuvent qu'avancer, mais comme aux échecs, ils peuvent manger un adversaire en diagonale. Il n'y a cependant pas de possibilités d'avancer de deux cases sur le premier coup.

Un joueur gagne lorsqu'il atteint la dernière rangée avec un pion ou lorsque l'adversaire ne peut plus jouer.

En quoi consiste le projet ?

Le projet consiste à créer des intelligences artificielles pouvant jouer sans aucune aide de la part d'un utilisateur. On nous a imposé deux petites IAs, et nous avons réfléchi, développé et implémenté trois grandes IAs.

Petites IAs :

- Une IA qui avance le plus possible : On fait en sorte que l'ordinateur avance toujours son pion le plus avancé.
- Une IA qui empêche la progression de l'adversaire : l'ordinateur doit bloquer le plus possible les pions adverses.

Grandes IA :

- L'approche de l'arbre : Cette IA est la plus complexe car elle consiste en le calcul de toutes les possibilités de jeu afin de choisir le chemin le plus avantageux
- L'approche du score par case : Cette IA consiste en l'attribution de points aux cases dans lesquelles des coups positifs ont été commis, et de privilégier le déplacement vers celles-ci
- L'approche du score par pions : Cette IA consiste en l'attribution de points aux pions qui commettent des actions positives, afin de privilégier leur jeu

Répartition des tâches

Besjan : Approche du score par pions + Rédaction du rapport

Anthony : L'approche du score par cases + IA qui avance le plus possible

Matthieu : Jeu de base + Approche avec l'arbre + Correction des codes + Aide à la rédaction, présentation & correction du rapport

Partie 2 : IAs primitives

En quoi consistent-elles ?

- Une IA qui avance le plus possible consiste à toujours faire avancer le pion le plus avancé sur le plateau pour qu'il ait une petite chance de gagner.
- Une IA qui empêche la progression des pions adverses consiste à essayer de bloquer le plus possible les pions adverses en opposant toujours un de ses pions en face.

Point de vue critique

IA qui avance le plus possible :

Si on parle d'un algorithme qui fait avancer le plus possible un pion sur un jeu, le gros problème de cet algorithme serait qu'il ne peut pas se défendre contre les stratégies adverses. Cela signifie que bien qu'il puisse manger les pions, il peut très bien foncer droit dans un mur sans s'en rendre compte. Cela peut limiter considérablement ses chances de succès dans le jeu, car la stratégie est la clé de la victoire dans ce jeu.

Pour résoudre ce problème, il pourrait être utile de doter l'algorithme de stratégies de défense ou de contre-attaque, afin qu'il puisse réagir aux menaces et protéger ses positions. Cela pourrait être réalisé en utilisant des techniques d'apprentissage automatique ou en programmant des règles de défense précises dans l'algorithme. En agissant de cette manière, l'algorithme serait mieux équipé pour affronter les défis et les obstacles qui se présentent dans le jeu, et il aurait une meilleure chance de réussir.

Cependant, le but de ces IAs est de rester simples et faciles d'implémentation. On ne se penchera donc pas sur une optimisation de celles-ci.

IA qui empêche la progression :

le gros problème de cette IA serait qu'elle n'a pas pour vocation de gagner, mais de créer des situations de match nul. Cela signifie qu'elle ne peut pas remporter la victoire en attaquant ou en prenant l'initiative, et qu'elle doit se contenter de défendre ses positions et d'empêcher l'adversaire de progresser.

Cela peut être un problème, car cela peut rendre le jeu moins intéressant et moins stimulant pour les joueurs. En effet, si l'IA ne peut pas gagner et ne peut créer que des situations de match nul, les joueurs risquent de perdre l'intérêt pour le jeu.

Pour résoudre ce problème, il pourrait être utile de doter l'IA de stratégies d'attaque ou de prise d'initiative, afin qu'elle puisse essayer de remporter la victoire et de rendre le jeu plus stimulant pour les joueurs. Cela pourrait être réalisé en utilisant des techniques d'apprentissage automatique ou en assemblant la première IA avec celle-ci pour que l'IA puisse se défendre et avancer quand elle en a l'opportunité pour gagner.

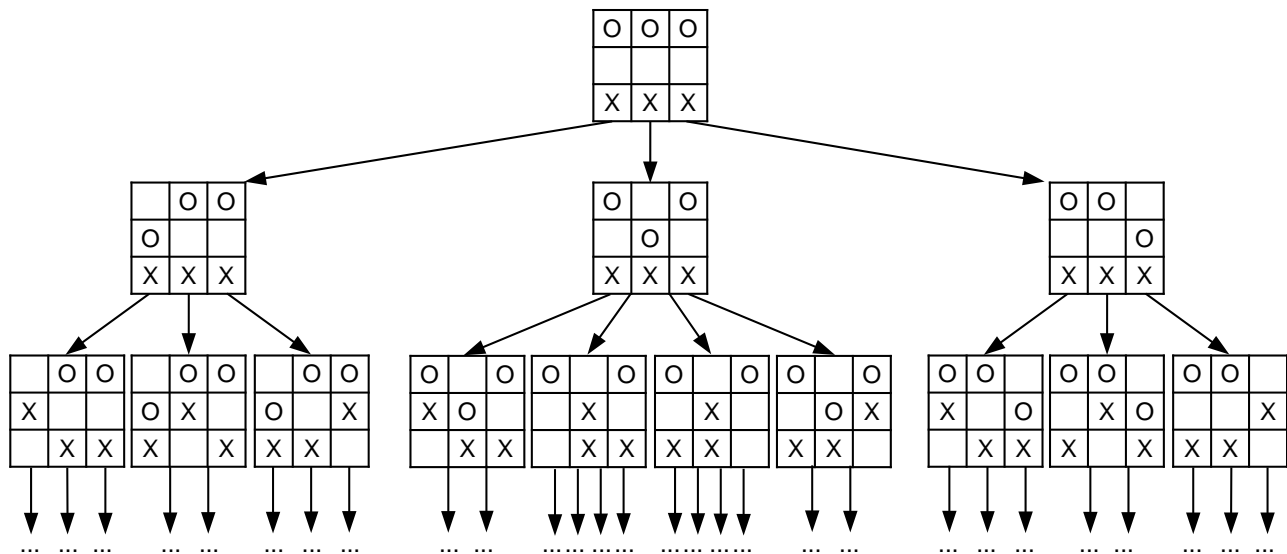
Mais encore une fois, ce n'est pas le but de ces IAs donc nous ne nous pencherons pas sur ces idées.

Retenons tout de même que ces IAs, bien que primitives, sont faciles à programmer et peuvent incarner une difficulté "facile" contre l'ordinateur, à défaut d'être optimales.

Partie 3 : L'approche de l'arbre

En quoi consiste cette approche ?

Celle-ci consiste en le calcul de chaque possibilité sous la forme d'un arbre, de la manière suivante :



Chaque branche de l'arbre possède une valeur qui est égale à la somme des valeurs de tous ses sous-branches.

Les feuilles ont pour valeurs :

- Si le plateau est une victoire, 2
- Si le plateau est une défaite, -2
- Si le plateau est une égalité, 1

Ce système permet à l'IA de savoir si une branche doit être privilégiée par rapport à une autre : une branche menant à plus de victoires/égalités que de défaites aura une valeur positive, sinon négative.

Choix réalisés lors de sa création

On aurait pu faire s'entraîner l'IA en la faisant reconnaître les possibilités à travers des parties simulées, mais ce n'est pas ce qui a été ici choisi.

L'option privilégiée a été de calculer toutes les possibilités en avance, sans passer par un entraînement donc

Un autre choix a été de valoriser l'égalité, puisqu'elle permet souvent d'éviter une défaite. On aurait pu l'ignorer en la mettant à 0 mais cela peut causer des cas où une branche moins bonne peut être choisie.

Après le calcul de toutes les branches, on repasse sur les feuilles pour changer leur valeur, afin de donner aux victoires le score maximum possible, et aux défaites le score minimum possible.

La valeur des branches ayant déjà été calculé à partir des valeurs citées précédemment, ce changement ne les affecte pas.

Ainsi, si une l'IA hésite entre gagner ou aller vers une branche menant vers beaucoup de victoires, elle ira gagner directement. À l'inverse, si elle hésite entre perdre ou aller vers une branche menant vers plein de défaites, elle évitera de perdre. Son comportement en bénéficie donc grandement.

Ses avantages

Cette IA a ses mérites :

- Presque optimale, puisqu'elle connaît toutes les possibilités de jeu
- Jeu rapide
- Implémentation peu difficile

Ses inconvénients

Cependant, elle a des défauts importants :

- Très lourde : plus de 240 Mo de RAM consommé pour une table 4×4 , calcul impossible pour une 5×5
- Calcul des possibilités long (4s pour une 4×4 , inestimable pour une 5×5)

Qu'en tirer ?

Cette IA est très utile pour les tables jusqu'en 4×4 , puisque c'est la limite à laquelle le programme peut calculer sans tomber à court de mémoire ou mettre trop de temps à calculer.

Son jeu étant quasi-optimal, elle est difficile à battre, mais tout de même possible à coincer si on sait s'y prendre.

Mais on ne peut pas l'utiliser pour des grandes tables, il faudra donc en utiliser une autre si l'on veut jouer sur des plateaux plus volumineux.

Partie 4 : L'approche du score par case

En quoi consiste cette approche ?

Cette IA consiste en la création de deux tableaux : l'un pour les rounds, l'autre pour les croix.

Chaque tableau est initialisé à 0. En fonction du joueur qui joue, un point est apporté à une case de son tableau s'il effectue un coup positif (manger un adversaire) à l'endroit où il l'a effectué.

Exemple avec une grille en 3×3 :

1	1	1
1	1	1
1	1	1

O		O
	O	
X	X	X

Deux tableaux de la même taille que celui de jeu ont été instanciés à 1. le joueur O a ensuite joué Un coup neutre.

1	1	1
1	1	1
1	1	1

1	1	1
1	1	1
1	1	1

O		O
	X	
	X	X

Un X a mangé un O sur la case centrale. On lui ajoute donc un point dans son tableau, à la même case.

1	1	1
1	2	1
1	1	1

Avant de jouer, l'IA tire un chiffre aléatoire entre 0 et 100. Grâce à celui-ci, on peut déterminer quel pion jouer selon les scores des différentes cases disponibles.

Exemple avec une grille en 4×4 :

1	1	1	1
1	1	1	1
1	1	1	1

O	O	O	O
●	●	●	●
X	X	X	X

On instancie un jeu en 4×4. les O commencent. Leurs possibilités sont modélisées par les lignes vertes.

Puisque les cases pointées par chaque possibilité ont le même score, elles doivent avoir la même probabilité d'être jouées : 25 %

1	1	1	1
1	1	1	1
1	1	1	1

1	1	1	1
2	1	1	1
1	1	1	1

	O	O	O
O			
X	X	X	X

1	1	1	1
3	2	1	1
1	1	1	1

Après quelques jeux, un nouvel essai commence avec l'état et les scores suivants.

Maintenant, les quatre possibilités n'ont pas la même valeur. alors, leur probabilité est déterminée par la formule :
 $\text{valeur} / (\text{valeurTotale})$

Les probabilités sont donc :

- Possibilité 1 : 42,86 %
- Possibilité 2 : 28,57 %
- Possibilité 3 : 14,29 %
- Possibilité 4 : 14,29 %

Pour déterminer quelle possibilité jouer, il suffit de regarder l'intervalle dans lequel le nombre généré aléatoirement est inclus :

- S'il est entre 0 (inclus) et 42,86 (exclu) on jouera la possibilité 1
- S'il est entre 42,86 (inclus) et 71,43 (exclu) on jouera la possibilité 2
- S'il est entre 71,43 (inclus) et 85,72 (exclu) on jouera la possibilité 3
- S'il est entre 85,72 (inclus) et 100 (exclu) on jouera la possibilité 4
- Si 100 tout pile est tiré, le dernier pion regardé est utilisé (ici la possibilité 4)
- Aucune possibilité n'est ignorée, puisque le score minimum est 1

Déterminer le coup à jouer par un nombre aléatoire permet d'éviter que la même case soit tout le temps jouée en début de jeu, et permet également de changer la façon de jouer à chaque nouvelle partie.

Choix réalisés lors de sa création

On a choisi d'utiliser des probabilités pondérées selon le score des cases, mais on aurait pu le faire différemment :

- On aurait pu rendre les probabilités exponentiellement plus grandes selon le score, plutôt que proportionnellement
- On aurait pu ajouter un biais à certaines cases dont on connaît d'avance l'avantage ou le désavantage

Ses avantages

- Le but de cette IA n'est pas d'être optimale et de réfléchir à la meilleure stratégie, mais de pouvoir jouer en fonction des cases les plus avantageuses durant les anciennes parties. Le fait de pouvoir jouer avec une IA qui retient les meilleurs coups peut être quitte ou double : l'utilisateur ne sait pas ce que l'IA a retenu et peut donc commettre des erreurs, mais l'IA aussi
- Contrairement à l'approche de l'arbre, cette IA met bien moins de temps à précalculer ce dont elle a besoin : 25000 parties sur un tableau 5x5 se simulent en à peine quelques secondes

Ses inconvénients

- Puisqu'elle n'est pas optimale, elle se retrouve parfois occupée à manger des pions adverses alors qu'avancer la ferait gagner car elle privilégie les cases avec un score plus élevé. Ce petit problème laisse des ouvertures pour le joueur, qui peut donc parfois gagner facilement.

Qu'en tirer ?

Cette IA laisse la possibilité à une personne débutante dans le jeu de jouer contre un ordinateur sans trop de difficulté.

Pour qu'elle soit plus optimale, il faudrait lui donner plus de conditions, comme prévoir les coups d'avance sur son adversaire ou se poser des questions du type « est-ce que si je mange ce pion, il pourra me manger ensuite ? ».

Elle est donc largement améliorable avec un peu plus de temps afin d'en tirer de meilleurs résultats et une partie plus complexe, laissant moins de mouvement au joueur et moins de failles qui pourraient lui assurer la victoire.

Partie 5 : L'approche du score par pion

En quoi consiste cette approche ?

Cette IA attribue un score à chaque pion selon les actions qu'il commet afin de le faire jouer plus ou moins souvent.

Pour cela, elle utilise des structures de données pour stocker l'information de chaque pion, comme leurs scores, leurs coordonnées et "morts". Elle utilise également un algorithme pour calculer la probabilité pour chaque pion d'être joué en utilisant leurs scores.

Cette approche permet à l'IA de jouer de manière aléatoire en fonction de l'utilité de chaque pion, ce qui peut simuler une certaine forme de réflexion ou d'intelligence.

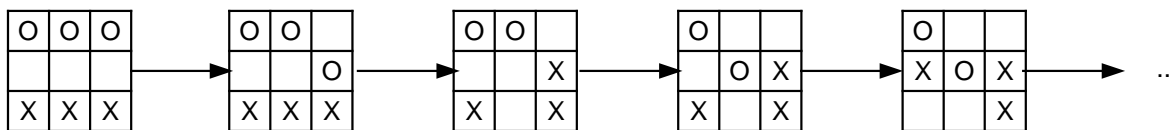
Comme pour l'approche du score par case, chaque pion commence avec un score de 1 et la possibilité à jouer est calculée à l'aide de l'intervalle dans lequel il se trouve par rapport à chaque probabilité.

Puis, à chaque fois qu'un pion en mange un autre, son score augmente.

(Schémas explicatifs en page suivante)

IA sans entraînement

Puisque les pions ne possèdent qu'un point chacun, ils jouent de manière aléatoire



Début de jeu, l'IA possède les pions X et le joueur les pions O

Le pion du milieu a été choisi (au hasard donc) et a mangé un pion adverse. On ajoute donc 1 à son score.

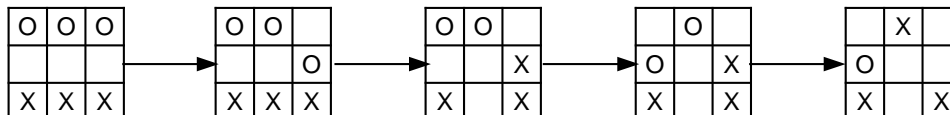
IA avec entraînement

Priorité des pions

Début de jeu, l'IA possède les pions X et le joueur les pions O

Le pion du milieu a un score plus haut et est donc priorisé, et mange. Cet écart n'est pas plus par le hasard total mais par un hasard biaisé

Ce même pion est toujours priorisé et aura donc de grandes chances d'être sélectionné pour la victoire au tour suivant

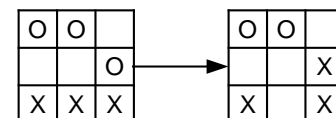


L'IA a effectué 9000 parties à l'avance contre elle-même.

Calcul de probabilité

Le programme choisit aléatoirement un pion selon son score avec la formule: $\text{score} / \text{scoreTotal}$

Par exemple, le premier pion aura $1 / (1+8) = 0,083 = 11,11\%$ de chance d'être joué



Notez que le troisième pion est ignoré dans le calcul car il est incapable de jouer

Pour retenir ces scores, les pions sont stockés dans un tableau de pointeurs vers une structure de données

L'IA choisit donc toujours un pion aléatoirement, mais en donnant plus de chances à certains pions

Joueur	Mort ?	Score	Probabilité de jeu	Coordonnées
X	Non	1	11,11%	0;2
X	Non	8	88,89%	1;2
X	Non	3	Ignorée	3;2

Choix réalisés lors de sa création

- On a choisi de prioriser les mouvements offensifs sur ceux neutres : si le pion qui joue peut manger, alors il mangera

Ses avantages

Flexibilité : En utilisant un nombre aléatoire pour décider quel pion jouer en fonction des chances calculées, l'IA peut jouer de manière différente à chaque partie, ce qui rend le jeu plus intéressant pour les joueurs.

Simulation de la réflexion : En utilisant des scores pour les pions, l'IA peut simuler une certaine forme de réflexion ou d'intelligence en jouant des coups qui ont un score plus élevé, ce qui peut être plus difficile à battre pour les joueurs.

Simplicité : La méthode utilisée par l'IA est relativement simple, ce qui facilite sa mise en place et son développement.

Adaptabilité : L'IA peut être facilement adaptée pour utiliser d'autres informations ou des algorithmes plus avancés pour prendre des décisions de jeu.

Personnalisation : l'IA peut être personnalisée en utilisant différents scores pour les pions pour créer une IA plus agressive ou plus défensive, selon les besoins du jeu.

Simplicité de maintenance : cette méthode d'IA n'utilise pas d'apprentissage automatique ni de réseaux de neurones, ce qui la rend plus facile à maintenir et à comprendre pour les développeurs.

Ses inconvénients

Réflexion présente mais primitive : En utilisant des scores pour les pions, l'IA ne tient pas compte des contextes de jeu, des différentes stratégies ou des positions des autres pions, ce qui peut la rendre moins efficace contre des joueurs expérimentés.

Simplicité : La méthode utilisée par l'IA peut être trop simple pour des joueurs expérimentés et ne pas offrir suffisamment de défi et donc rendre le jeu plus facile à appréhender.

Dépendance des scores : l'IA est dépendante de la qualité des scores attribués aux pions, si ces scores ne sont pas assez pertinents, l'IA ne sera pas efficace.

Peu de capacité d'analyse : cette méthode d'IA ne permet pas une analyse profonde des situations de jeu et des mouvements possibles, ce qui peut la rendre moins efficace pour prendre des décisions de jeu de haut niveau.

Limite de l'adaptabilité : cette méthode d'IA ne peut pas s'adapter aux situations de jeu imprévues ou aux erreurs humaines, ce qui peut la rendre vulnérable aux joueurs expérimentés.

Qu'en tirer ?

Cette méthode d'IA est relativement simple à mettre en place et à comprendre, ce qui peut être avantageux pour les développeurs qui ne veulent pas faire de système trop complexe.

Elle peut simuler une certaine forme de réflexion ou d'intelligence en jouant des coups qui ont un score plus élevé, ce qui peut être plus difficile à battre pour les joueurs qu'un simple programme aléatoire.

Mais elle est dépendante de la qualité des scores attribués aux pions, il est important de les définir de manière adéquate pour que l'IA soit efficace.

Enfin, elle ne permet pas une analyse profonde des situations de jeu et des mouvements possibles, il est donc important de ne pas la considérer comme étant la seule solution pour notre jeu.

Partie 6 : Conclusion

Que peut-on retenir de ce projet ?

Le point le plus important ici est le fait qu'une seule approche n'est pas suffisante si l'on veut vraiment étudier le sujet de l'intelligence artificielle en profondeur.

Il faut bien garder en tête que certaines IAs sont plus douées dans certains domaines que d'autres, un peu comme nous les humains.

Il n'y a pas de réelle « meilleure approche » au final puisqu'elles ont toutes leurs forces et faiblesses.

Quelle(s) solution(x) retenons-nous ?

Les solutions à retenir comme réellement pertinentes ici sont les trois grandes IAs (arbre, score par case et score par pion) puisque celles-ci présentent une réelle simulation de réflexion de l'ordinateur, contrairement aux petites IAs (pion le plus avancé et bloqueur de progression) qui se contentent d'appliquer une stratégie bêtement et de manière très prévisible.

Merci d'avoir lu !

Nous espérons que notre étude vous aura intéressé, vous trouverez les fichiers sous les noms suivants :

Jeu sans modifications : jeu.cpp

Approche de l'arbre : arbre.cpp

Approche du score par case : scorecase.cpp

Approche du score par pion : scorepion.cpp

Avance du pion le plus avancé : leplusavance.cpp

Bloqueur de progression adverse : Indisponible, faute de temps :/