

# Rapport IA

Ce rapport retrace le processus de développement et d'optimisation d'un agent intelligent conçu pour maîtriser une variante complexe du jeu d'Awale à 16 trous. La trame de ce document suit l'évolution chronologique et technique de notre approche : nous partons des fondamentaux de la théorie des jeux avec l'implémentation d'un moteur Minimax classique, pour progressivement lever les verrous technologiques liés à l'explosion combinatoire et aux contraintes du temps réel. À travers une analyse détaillée, nous exposons comment l'intégration de l'élagage Alpha-Bêta, couplée à une gestion fine de la mémoire via les Tables de Transposition et le hachage de Zobrist, a permis de transformer un algorithme théorique en un joueur compétitif. Enfin, au-delà de l'architecture logicielle, nous disséquons l'heuristique stratégique la "vision" de l'IA concernant les graines colorées et la mobilité avant de démontrer concrètement l'efficacité de ces solutions par une série de benchmarks comparatifs.

## Table des matières

I. Introduction .....	3
II. Le Moteur de Recherche .....	4
III. Optimisations Avancées .....	4
IV. La Fonction d'Évaluation .....	5
V. Benchmark et Résultats Expérimentaux .....	6
VI. Conclusion et Perspectives.....	6

## I. Introduction

L'objectif principal de ce projet est la conception et l'implémentation d'un agent intelligent capable de rivaliser à un niveau compétitif dans notre variante stratégique de l'Awale. La complexité majeure de cette tâche ne réside pas uniquement dans la qualité décisionnelle, mais dans l'adaptation aux mécaniques spécifiques de ce jeu (plateau de 16 trous, gestion de trois types de graines : Rouges, Bleues et Transparentes). L'agent doit être en mesure d'identifier la meilleure stratégie de semaille possible en respectant des contraintes temporelles strictes, ce qui impose une rigueur algorithmique particulière pour gérer les multiples variations de règles associées à chaque couleur. Face à la nature combinatoire du jeu, où le nombre de scénarios possibles à chaque tour est énorme, amplifié par le choix de la couleur à jouer qui augmente le facteur de branchement, une recherche exhaustive est impossible en temps réel.

La problématique centrale consiste donc à permettre à l'IA d'explorer l'arbre des possibilités le plus profondément possible pour anticiper les captures complexes et les famines, tout en filtrant efficacement les scénarios de semailles non pertinents. Il s'agit de trouver l'équilibre parfait entre la vitesse d'exécution du moteur et la précision de l'analyse stratégique sur un plateau en constante mutation.

Pour répondre à ce défi, nous avons adopté une approche de développement incrémentale. Le travail a débuté par la mise en place d'un moteur de recherche fondamental reposant sur l'algorithme Minimax, permettant une simulation fiable des mouvements de graines. Par la suite, ce moteur a été successivement amélioré par des techniques d'élagage (Alpha-Bêta) et, dans sa version finale, par des optimisations avancées de mémoire (Table de Transposition via hachage Zobrist) et de gestion du temps (Approfondissement Itératif), indispensables pour traiter la complexité additionnelle introduite par les graines spéciales.

## II. Le Moteur de Recherche

Le cœur décisionnel de notre agent repose sur l'algorithme Minimax, adapté ici à la complexité particulière de notre variante d'Awale à 16 trous. Contrairement aux jeux d'échecs où les pièces se déplacent, notre jeu implique une modification dynamique de l'état global du plateau via le mécanisme de semaille. L'arbre de recherche modélise chaque décision possible : le choix du trou (parmi les 8 contrôlés) et le choix de la couleur à égrener (Rouge, Bleu, ou Transparent avec ses variantes d'assignation). L'IA projette ces séquences de semaines dans le futur, cherchant à maximiser son score de graines capturées tout en minimisant celui de l'adversaire. La profondeur de l'arbre permet d'anticiper des pièges tactiques, tels que les famines ou les captures en chaîne ("Krou"), invisibles lors d'une analyse superficielle.

Cependant, avec trois types de graines et des règles de distribution conditionnelles, le facteur de branchement est élevé. Pour endiguer cette explosion combinatoire, nous avons intégré l'élagage Alpha-Bêta. Cette technique permet d'ignorer des branches entières de calcul dès lors qu'une réfutation est trouvée. Concrètement, si l'IA découvre qu'un coup spécifique permet à l'adversaire de capturer instantanément 10 graines, elle abandonne l'exploration des sous-variantes de ce coup, sachant qu'elle ne choisira jamais cette option si une alternative plus sûre existe déjà. Cela permet de concentrer la puissance de calcul sur les séquences de jeu les plus critiques et pertinentes.

## III. Optimisations Avancées

Pour dépasser les limites de la recherche brute et atteindre des profondeurs de jeu compétitives, nous avons implémenté une Table de Transposition reposant sur le hachage de Zobrist. Dans notre jeu, une même configuration de graines (répartition des Rouges, Bleus, Transparents) peut être atteinte par différentes séquences de coups. Pour exploiter cette redondance, nous attribuons une signature unique sur 64 bits à chaque configuration du plateau, calculée en XORant des clés aléatoires associées à chaque quantité de graine par trou et par couleur. Lorsqu'un état est analysé, son résultat (score et meilleur coup) est stocké en mémoire. Si l'IA rencontre à nouveau cette position via un autre chemin, elle récupère instantanément l'analyse précédente, économisant ainsi des milliers de cycles processeur.

Cette mémoire persistante est la clé de voûte de notre gestion du temps via l'Approfondissement Itératif.

Plutôt que de lancer une recherche à une profondeur fixe risquée, l'algorithme explore le jeu strate par strate (profondeur 1, puis 2, puis 3, etc.) jusqu'à l'épuisement du temps alloué (par exemple 100ms ou 3s selon le tournoi). Cette approche assure qu'une décision valide est toujours disponible, même en cas d'interruption brutale. De plus, elle alimente l'Ordonnancement des Coups : les meilleurs coups identifiés lors des itérations précédentes (les *Hash Moves*) sont testés en priorité lors de l'itération suivante. En examinant les coups forts tôt, l'élagage Alpha-Bêta devient drastiquement plus efficace, car il peut invalider les mauvaises branches beaucoup plus rapidement.

## IV. La Fonction d'Évaluation

Au-delà de la puissance de calcul brute fournie par l'algorithme de recherche, l'intelligence stratégique de notre agent réside dans sa fonction d'évaluation heuristique. Lorsque l'exploration de l'arbre atteint la limite de temps ou de profondeur, l'algorithme doit juger la "qualité" de l'état du plateau sans connaître l'issue finale de la partie.

Pour ce faire, nous avons défini une combinaison linéaire de critères stratégiques pondérés, permettant de traduire une configuration complexe de graines en un score numérique unique. Le critère le plus évident reste le différentiel de score courant (le nombre de graines capturées), mais se baser uniquement sur cet indicateur mène à un jeu à courte vue, incapable de construire des stratégies à long terme.

Pour affiner ce jugement, nous avons intégré des paramètres positionnels cruciaux pour cette variante de l'Awale. La notion de "Hoarding" (accumulation) a été valorisée : l'IA est récompensée si elle parvient à conserver un nombre important de graines dans ses propres trous, ce qui constitue une réserve défensive et un potentiel d'attaque futur.

De même, la "Mobilité" est prise en compte pour éviter les situations de famine où l'agent serait forcé de jouer un coup désavantageux faute d'alternatives. Enfin, une attention particulière a été portée aux graines "Transparentes". En raison de leur capacité à changer de valeur ou d'appartenance selon le contexte, leur présence dans notre camp se voit attribuer un poids heuristique spécifique, l'agent les considérant comme des actifs de haute valeur stratégique à préserver pour les moments critiques. L'équilibrage de ces différents facteurs (Score, Hoarding, Mobilité, Graines Spéciales) s'effectue via un système de coefficients ajustés manuellement.

Si le poids du score immédiat est trop élevé, l'IA devient opportuniste et tombe dans des pièges à retardement. À l'inverse, si l'accumulation est surévaluée, elle joue trop passivement. La version actuelle utilise un réglage hybride qui privilégie la sécurité positionnelle en début de partie et bascule vers une agressivité matérielle (capture pure) lorsque la fin de partie approche, permettant une adaptation dynamique au déroulement du jeu.

## V. Benchmark et Résultats Expérimentaux

Afin de valider les choix architecturaux, nous avons procédé à une analyse comparative entre le moteur de base et la version finale optimisée. Les tests ont mis en évidence l'impact critique de la Table de Transposition couplée au hachage de Zobrist dans un jeu comme l'Awale, où les boucles de jeu et les répétitions de positions sont fréquentes. Là où la version standard recalculait inutilement des séquences de semaines déjà rencontrées, la version optimisée parvient à récupérer les évaluations en mémoire quasi instantanément.

Cette économie de ressources s'est traduite par une augmentation significative de la profondeur de recherche moyenne, permettant à l'IA de voir plusieurs coups plus loin ("plies") dans le même laps de temps imparti.

L'efficacité de l'élagage Alpha-Bêta a également été décuplée par l'ordonnancement des coups. En priorisant l'analyse des coups qui capturent des graines ou qui ont été identifiés comme "meilleurs" lors de l'itération précédente, l'algorithme parvient à couper les branches inintéressantes beaucoup plus tôt. En pratique, cela signifie que notre agent peut anticiper des combinaisons complexes, comme des captures en chaîne déclenchées par des graines bleues, que la version naïve ne pouvait pas détecter faute de temps. La robustesse de l'IA s'en trouve renforcée : elle ne perd plus au temps ("timeout") et maintient un niveau de jeu constant, même dans les configurations de plateau chargées en graines qui génèrent habituellement une explosion du nombre de coups possibles.

## VI. Conclusion et Perspectives

Ce projet a permis d'aboutir à un agent intelligent fonctionnel et performant, capable de gérer la complexité combinatoire d'un Awale à 16 trous avec graines colorées. L'intégration successive de l'algorithme Minimax, de l'élagage Alpha-Bêta et surtout des optimisations de mémoire (Table de Transposition) et de temps (Iterative Deepening) a transformé un moteur de recherche théorique en un joueur compétitif. L'agent ne se contente plus de calculer mécaniquement ; il reconnaît des structures de jeu grâce au hachage et évalue la dangerosité des positions grâce à une heuristique prenant en compte les spécificités des graines transparentes et la mobilité.

Néanmoins, le système actuel possède encore une marge de progression, notamment au niveau de sa fonction d'évaluation. Les coefficients qui régissent le comportement de l'IA (importance de l'accumulation vs capture immédiate) sont actuellement statiques et définis empiriquement. Une évolution naturelle de ce travail consisterait à implémenter une méthode d'apprentissage automatique (telle qu'un algorithme génétique ou de la régression logistique) pour ajuster automatiquement ces poids en faisant jouer l'IA contre elle-même des milliers de fois. Cela permettrait d'affiner "l'intuition" du programme et de corriger les derniers biais de jugement humains restants.