

MCTS

PUISSANCE 4



Table des matières

INTRODUCTION	3
MODELISATION & MISE EN PLACE	4
FONCTION D’EVALUATION	4
RESULTATS.....	5

INTRODUCTION

Pour ce TP, nous avons développé le jeu du Puissance 4, un grand classique des jeux de société. Cependant, nous avons ajouté une dimension supplémentaire à ce projet en intégrant trois types d'Intelligence Artificielle (IA) populaires : Minimax, Alpha-Beta et MCTS (Monte Carlo Tree Search). Notre objectif principal était de comparer ces différentes approches d'IA et d'évaluer leur performance respective dans le contexte spécifique du Puissance 4. Nous avons aussi implémenté la possibilité pour un humain de jouer, que ce soit contre une des 3 IA ou contre un autre humain éventuellement.

Quelques présentations tout d'abord, le Puissance 4 est un jeu de stratégie abstrait dans lequel deux joueurs s'affrontent en déposant à tour de rôle des pions dans une grille verticale de 6 lignes sur 7 colonnes. Les pions viennent d'empiler les uns sur les autres. Le but est de former une ligne de quatre pions de sa couleur, que ce soit horizontalement, verticalement ou en diagonale, avant l'adversaire. Le jeu possède des règles basiques, mais une complexité croissante au fur et à mesure que la grille se remplit. Chaque coup va demander de plus en plus de réflexion pour un humain, puisque de nouvelles combinaisons vont se créer à chaque étape.

Le développement d'une IA capable de jouer au Puissance 4 présente des aspects intéressants et challengeant, car le jeu offre de nombreuses combinaisons possibles, rendant l'exploration exhaustive de l'arbre de jeu impossible dans un temps raisonnable. C'est pourquoi nous avons exploré trois approches différentes d'IA pour trouver la meilleure stratégie possible tout en optimisant le temps de calcul : le Minimax, l'Alpha-Beta et le MCTS. Nous allons nous efforcer à comparer ces 3 modèles dans ce rapport. Dans cette même idée, nous avons en place plusieurs stratégies pour Alpha-Beta au travers de nos fonctions d'évaluations.

Nous présenterons les différentes étapes de développement du jeu de Puissance 4, ainsi que les spécificités de chaque algorithme d'IA que nous avons implémenté. Nous analyserons également les résultats obtenus en termes de performance et d'efficacité des différentes approches, et nous discuterons des avantages et des limitations de chacune d'entre elles. Nous avons aussi créé des parties entre les modèles, afin d'analyser leurs comportements.

MODELISATION & MISE EN PLACE

Pour mettre en place notre projet de Puissance 4, nous avons choisi le langage Python, qui a l'avantage d'être souple et simple en termes d'algorithmie.

Notre code se compose de 5 fichiers, présenté dans la Figure 1 :

- ✚ bcolors.py, qui est un simple fichier de configuration pour les couleurs de notre interface graphique.
- ✚ evaluate_functions.py, qui contient nos 3 fonctions d'évaluations (Modérée, Agressive, Défensive).
- ✚ main.py, qui contient l'architecture de la grille, la logique du jeu ainsi que les différents algorithmes. C'est ici que nous lançons le jeu.
- ✚ mcts_board.py, qui est une version du plateau de jeu adaptée pour correspondre aux besoins de l'algorithme MCTS
- ✚ node.py, qui est une classe personnalisée qui nous permet de gérer plus facilement les nœuds dans nos algorithmes.

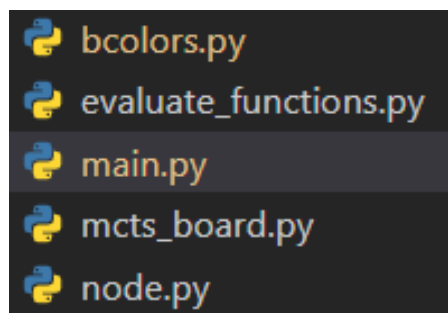


Figure 1 : Architecture du projet

FONCTION D'ÉVALUATION

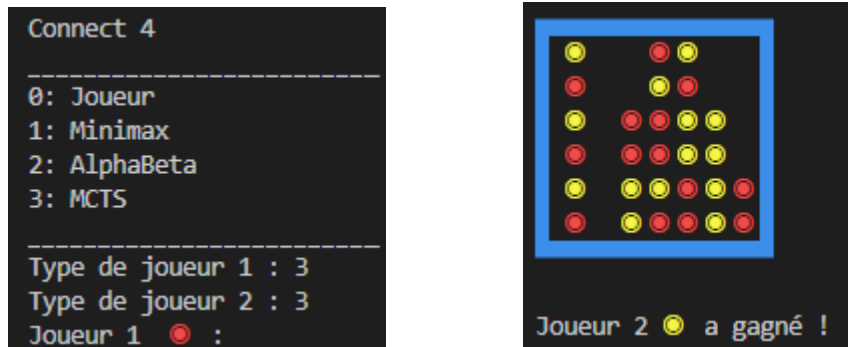
Pour notre fonction d'évaluation, qui permet d'attribuer des points à chaque joueur en fonction des combinaisons de 4 cases possibles. Le point majeur est que nous faisons la différence entre les points des deux joueurs pour définir quelle sera le meilleur mouvement au prochain tour. Notre fonction de base donne des points selon le modèle suivant :

- ❖ 1000 points si 4 pions de la même couleur,
- ❖ 50 points si 3 pions de la même couleur et un vide,
- ❖ 5 points si 2 pions de la même couleur et deux vides,
- ❖ 1 point si 1 pion et 3 vides.

RESULTATS

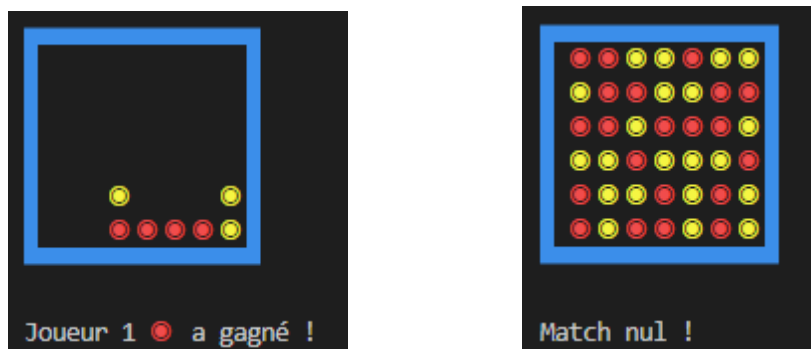
Nous avons choisi de présenter nos résultats par le biais de 3 matchs de puissance 4 entre deux joueurs contrôlés par nos IA et nos algorithmes MCTS. A chaque match, nous avons changé certains paramètres afin de modifier le comportement de nos joueurs.

1^{er} match (3000 de budget) :



Pour cette première partie, nous avons pris un budget assez grand, dans la moyenne de ce que peut supporter notre machine. On voit ici une partie cohérente, qui se termine dans un nombre de coups qui nous semble assez raisonnable.

2^{ème} match (100 de budget) :



Pour cette seconde partie, nous avons pris un budget très petit, afin de vérifier que nos algorithmes ne suffisent pas avec si peu. On remarque ici assez aisément que les coups joués se rapprochent de l'aléatoire. On peut ainsi se retrouver avec des parties qui se terminent en seulement quelques coups, comme certaines qui se terminent par une égalité.

3^{ème} match (10000 de budget) :



Pour cette troisième et dernière partie, nous avons pris un budget bien plus conséquent. On remarque ici que les coups ont encore plus de sens. On arrive ici à des parties qui se terminent quasiment à chaque fois par une victoire de l'un des deux joueurs, mais en jouant généralement plus de coups que les autres matches.