HONORÉ Alexandre

RIGONNET Arthur

BLANCHÉ Thomas



INFO 3 IA pour jeux à 2 joueurs

Table des matières

[Introduction 2](#_Toc134283640)

[I ) Recherches et démarrage du projet 2](#_Toc134283641)

[1. Recherches préliminaires 2](#_Toc134283642)

[Résultats 3](#_Toc134283643)

[Discussion 3](#_Toc134283644)

[Conclusion 4](#_Toc134283645)

[Annexes 4](#_Toc134283646)

# Introduction

Dans le cadre des mini-projets de PEIP2, nous avons décidé de nous concentrer sur la mise au point de différentes formes d'intelligence artificielle appliquées au célèbre jeu de société "Puissance 4". Le concept est simple, aligner 4 pions de sa couleur pour gagner. Les pions peuvent être alignés horizontalement, verticalement ou diagonalement. Pour cela, les joueurs laissent tomber chacun à leur tour un jeton de leur couleur dans la colonne de leur choix. Malgré des règles aussi simples, les stratégies pour gagner sont nombreuses. Ainsi nous pouvons trouver un intérêt particulier au développement d’intelligences artificielles. En effet, il existe un nombre colossal de parties possibles. Soit environ 1,6x1013 parties et seulement 69 sont gagnantes. Calculer toutes les parties afin de gagner à coup sûr est donc impossible. C'est pourquoi nous avons recours aux intelligences artificielles pour développer des algorithmes de jeu capables d'apprendre et d'adapter leur stratégie en fonction des mouvements de l'adversaire.

Jeu de puissance 4 classique

L'objectif principal de ce projet était de nous immerger pour la première fois dans l'univers de l'intelligence artificielle afin d'acquérir une compréhension plus approfondie de son fonctionnement, de ses capacités et de ses limites. Nous avons ainsi développé trois programmes distincts : Minimax, Alpha-Bêta et Monte Carlo Tree Search, dans le but de les faire s'affronter les uns contre les autres et de recueillir et d’analyser différentes statistiques comme le temps d’exécution, le pourcentage de victoire etc…

Dans ce rapport final, nous allons exposer en détail le processus de développement de ce projet. Nous commencerons par présenter les recherches préliminaires que nous avons menées, ainsi que l'organisation que nous avons adoptée pour assurer une collaboration efficace. Nous aborderons ensuite les différentes étapes de création du jeu, à la fois sous sa forme textuelle et graphique, et nous expliquerons les méthodes que nous avons employées pour intégrer les trois types d'intelligence artificielle dans le jeu. Enfin, nous procéderons à une analyse détaillée des statistiques recueillies lors des différents matchs qui ont opposé les programmes entre eux.

# I ) Recherches et démarrage du projet

## Recherches préliminaires

Lorsque nous avons commencé à travailler sur ce projet, notre première étape a été de nous informer sur les différentes intelligences artificielles que nous allions développer. Pour ce faire, nous avons exploré diverses ressources et nous allons maintenant détailler comment chacune d'entre elles nous a été utile.

Pour les algorithmes "minimax" et "alpha-bêta", nous avons trouvé des vidéos très instructives d'Hugo Larochelle sur YouTube. Dans ces vidéos, il explique en détail comment fonctionnent ces deux types d'intelligences artificielles, en soulignant leurs avantages et leurs inconvénients. En outre, il nous a fourni des informations sur les propriétés de ces deux algorithmes, tels que leur complexité en temps et en mémoire.

Pour comprendre le fonctionnement des Monte Carlo Tree Search, nous avons exploré différentes ressources, et en particulier, nous avons trouvé une vidéo très instructive de Random Coder qui explique en détail le fonctionnement de cet algorithme. Dans cette vidéo, il fournit une description complète de l'algorithme, ainsi qu'une implémentation en python de l'intelligence artificielle pour le jeu du puissance 4. Nous avons trouvé cette vidéo très utile pour comprendre les concepts clés des Monte Carlo Tree Search et leur application pratique. En outre, Random Coder a également réalisé un tutoriel écrit qui nous a aidé à mieux comprendre les concepts expliqués dans la vidéo. Ce tutoriel écrit nous a permis de revenir sur les points clés et de mieux comprendre certains aspects plus complexes de l'algorithme. Nous avons également trouvé que la combinaison de ces deux ressources nous a permis de bien saisir les différents concepts liés aux Monte Carlo Tree Search et leur application au jeu du puissance 4.

## Création git, cahier des charges, répartition des tâches

Lors de la réalisation de ce projet, nous étions trois et nous avons dû nous organiser efficacement pour travailler ensemble. L'un des premiers défis auxquels nous avons été confrontés a été de trouver un moyen de travailler simultanément sur le projet. Heureusement, grâce à nos cours d'informatique du semestre quatre, nous avons appris à utiliser la plateforme GitHub, qui s'est révélée être une solution idéale. GitHub est une plateforme de développement collaborative en ligne qui permet à plusieurs développeurs de travailler simultanément sur le même projet. Elle fournit également des outils de gestion de versions et de suivi de bugs pour faciliter la collaboration et la coordination. En outre, GitHub est une solution totalement gratuite, ce qui était un avantage considérable pour notre projet. En utilisant GitHub, nous avons ainsi pu créer un dépôt de code commun pour le projet, où nous avons pu partager et collaborer sur les différents éléments du code. Grâce à cette solution, nous avons pu travailler efficacement en équipe et gérer le développement du projet de manière collaborative.

Une fois ce problème résolu, nous nous sommes concentrés sur la réalisation d'un cahier des charges pour nous donner des objectifs clairs sur ce que nous allions réaliser. Le cahier des charges a été une étape essentielle pour la planification et l'organisation du projet, car il a défini les objectifs, les moyens, les délais, les fonctionnalités, les exigences et les contraintes de l'application que nous devions développer. Le cahier des charges a également été utile pour nous permettre de suivre notre avancement tout au long du projet. Nous avons pu nous référer à ce document à tout moment pour nous assurer que nous étions sur la bonne voie et que nous respectons les délais et les spécifications du projet. Le document est par ailleurs disponible en annexe de ce rapport.

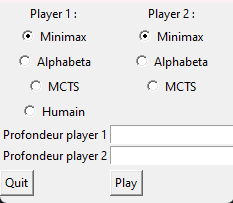
# II) Développement du puissance 4

## Version textuelle

Pour commencer la réalisation de ce projet, nous avons décidé d'utiliser le langage de programmation Python, qui est largement utilisé dans le domaine de l'intelligence artificielle et que nous voulions approfondir. Avant de pouvoir développer des intelligences artificielles pour le jeu de puissance 4, nous avons dû créer une version textuelle du jeu. Cette version fonctionne uniquement dans la console et dispose d'un affichage rudimentaire mais suffisant pour jouer. Afin de créer ce jeu, nous avons dû mettre en place plusieurs éléments clés. Tout d'abord, il nous a fallu un plateau de jeu qui est représenté par une liste contenant six autres listes de sept éléments chacune. Chaque liste représente une colonne du plateau. Nous avons également créé une fonction qui permet à un joueur de laisser tomber un jeton de sa couleur dans une colonne donnée, mais qui vérifie auparavant si la colonne n'est pas déjà remplie. Cette fonction est utilisable aussi bien par le joueur humain que par les intelligences artificielles. Afin d'assurer un affichage du plateau dans la console, nous avons créé une autre fonction qui parcourt le plateau et affiche, pour chaque case, un caractère différent en fonction du type de jeton (par exemple, '1' ou '2'). Enfin, la dernière fonction que nous avons créée est celle qui teste si un des joueurs a gagné. Cependant, cette fonction n'était pas si simple à implémenter car il fallait tester toutes les combinaisons possibles pour déterminer si quatre jetons étaient alignés. Ainsi, il fallait examiner chacune des colonnes, chacune des lignes et chacune des diagonales possibles. Nous avons aussi dû apprendre à utiliser des arguments lors de l'exécution de notre programme afin de choisir les deux joueurs qui s’affronteront, ainsi que la profondeur des algorithmes. Nous nous sommes servis du module python argparse.

## Version graphique

Le cahier des charges établi pour le projet inclut la réalisation d'une version graphique du jeu de Puissance 4. Pour cela,nous avons choisi d'utiliser la librairie graphique Tkinter, car certains d'entre nous l'avaient déjà utilisée lors de projets de spécialité NSI au lycée et en avaient donc une certaine familiarité. Nous avons ensuite développé une fonction spécifique chargée de l'affichage du plateau de jeu. Cette fonction commence par dessiner les cases vides lors de l'initialisation du programme. Ensuite, après chaque action d'un joueur ou d'une intelligence artificielle, la fonction dessine le jeton qui vient d'être déposé. L'implémentation de cette fonction a été essentielle pour la réalisation de la version graphique du jeu de Puissance 4, car elle permet aux joueurs de visualiser le plateau de jeu en temps réel et de suivre l'évolution de la partie.

De plus, comme stipulé dans le cahier des charges, nous avons également travaillé sur le développement d'un menu pour permettre aux utilisateurs de choisir les deux joueurs pour la prochaine partie, ainsi que la profondeur des intelligences artificielles si celles-ci sont choisies. Dans ce but, nous nous sommes appuyés sur la documentation française de Tkinter pour concevoir un menu fonctionnel et ergonomique pour les utilisateurs. La mise en place de ce menu a été une étape importante pour permettre une utilisation simplifiée de l'application, car elle permet aux utilisateurs de personnaliser leurs parties selon leurs préférences.

Menu graphique

Une image contenant graphique

Description générée automatiquement

Affichage avec Tkinter