

Rapport du Projet Tournoi d'Intelligence Artificielle Jeu Colonnes de trois en réseau

Partie IA

BARADI Farah, MONTEIRO Thiago

Master 1 Informatique

2021-2022

Nous souhaitons dans un premier temps remercier tous nos enseignants, qui nous ont permis de progresser dans ce projet tout en répondant à nos questions.

Introduction :

Ce projet a été effectué lors du deuxième semestre de Master 1 durant les mois de mars jusqu'à mai, dans le cadre de l'UE « Projet Tournoi d'Intelligence Artificielle » unissant les matières de "Systèmes Communicants et Synchronisés" et "Méthodes et Outils pour l'Intelligence Artificielle" encadré par BOUQUET FABRICE et FELEA VIOLETA

L'objectif de ce projet consistait à implanter le jeu "colonne de trois" avec des règles de jeu correspondant aux celles expliquées dans partie d'intelligence artificielle et d'implémenter la communication entre le client et le serveur ainsi avec le serveur d'ingénierie artificielle permettant de jouer en réseau

Le joueur (le client) et le serveur est implémentés en langage C et le serveur IA est développé en Java.

La communication au départ consiste de deux parties : une demande de participation envoyer par le joueur au serveur, dès que c'est accepté, on passe à la deuxième partie qui est le jeu de coup par les deux joueurs.

Chaque joueur fait appel au moteur IA pour connaître le bon coup à jouer dès que le jeu est toujours possible selon les règles de jeu et l'informe aussi de coup joué par son adversaire.

Les coups envoyés par les joueurs sont vérifiés et administrés par un processus serveur, arbitre du jeu, écrit en langage C.

Le jeu commence par une demande de participation envoyer de la part de client au serveur C, dès que cette demande est acceptée, chaque joueur va envoyer une demande à son serveur IA pour savoir le meilleur coup à jouer, et il l'envoie après au serveur C pour la valider.

Le jeu de coup est en deux parties, un placement des pions sur le tableau de jeu, ensuite un déplacement des pions selon les règles de jeu. Les joueurs jouent ces coups un par un

Déroulement :

Quand le joueur envoie une demande d'un coup à son serveur IA, le serveur fait appel à la fonction principale getNextMove qui regarde si tous les pions des joueurs sont placés sur le tableau de jeu ou pas, et ça pour savoir si c'est un coup de placement ou un coup de déplacement.

Ensuite pour chaque type de coup, elle appelle la fonction decideMove pour trouver le meilleur coup en utilisant l'algorithme alpha-beta qui sera expliqué dans une autre section.

Elle traite le cas d'un coup de type Passe (alors on ne change rien dans le tableau actuel), sinon soit on fait le placement en ajoutant un coup sur un carré de tableau, soit on déplace un pion en le supprimant du top de la liste et l'ajoute sur un autre.

On fait aussi attention au numéro du match actuel, car si c'est le premier c'est le joueur avec les pions blancs qui va commencer le match, et si c'est le deuxième c'est à celui avec des pions noirs, tout en sachant que le joueur ne change pas la couleur de leurs pions, c'est juste le tour qui change entre les deux joueurs.

- **Fonctions :**

Ici on vous explique quelques fonctions pour une meilleure compréhension du déroulement.

On utilise aussi la fonction saveOpponentMove pour modifier le tableau après avoir reçu les informations concernant le coup joué par l'adversaire (les joueurs) et pour garder le jeu à jour à chaque coup joué.

La fonction reset pour initialiser le tableau de jeu (avec aucun pion dessus) au début de chaque match.

Pour chaque joueur, on a créé une liste des pions, cette liste est une liste des pions dans la main du joueur (toujours pas mis sur le tableau de jeu), à chaque placement d'un coup sur le tableau de jeu, un pion est pris de sa liste de pions. Cette liste est initialisée (en la remplissant avec le maximum nombre des pions '8') au début de chaque match.

- **Classes :**

Ici on vous explique deux classes pour une meilleur compréhension de déroulement aussi.

Pour la mettre en place des règles de joue, on a créé la class pion, qui va nous idée à changer l'état de pion (lock/unlock) selon son ordre sur le carrée (le top est à la statue unlock et tout ce qui au-dessous et lock). Ce qu'on a bien utilisé dans un autre class sui s'appelle square dans les fonctions d'ajoute et de supprime d'un pion dans un carrée. Dans la classe square on a gérer le nombre maximum des pions sur un carrée et aussi la règle de gagne la joue

Alpha-Beta:

Le rôle principal de moteur IA et de calculer et donnée le joueur son meilleur coup à jouer pour qu'il gagne le match courant. Tout en tenant compte des règles de joue.

Alors on a une fonction principale qui est l'alpha-beta. Qui utilise deux sous-fonction une maximizer qui est considéré comme le joueur actuelle et la fonction minimizer pour son adversaire.

Cette fonctionne cherche le meilleur coup selon le type de coup cherché actuellement. Et pour chaque coup autorisé, elle cherche à savoir s'elle est le meilleur coup ou pas en continuation le joue avec ce coup pour le joueur. Alors c'est comme si on bascule la joue à partir de ce coup entre les deux joueurs. Et ainsi de suite pour tous les coups autorisés mais avec une limite maximale pour chaque coup défini par la profondeur.

- **Avantage et choix :**

Un grand avantage de cette fonction qu'elle n'est pas obligée à continuer la recherche d'un meilleur coup jusqu'à arriver au profondeur max, s'elle le trouve avant elle arrête. Ça nous aide à gagner de temps, et c'est la différence entre elle et le minimax algorithm. Et pour cela on a choisi cet algorithm et pas le minimax algorithm.

- **Recherche :**

Après plain de débogage, on a découvert des problèmes liés à l'espace mémoire, en fait le serveur java ne copiait pas dans la mémoire les informations liés à chaque carré des neuf carrées de tableau de jeu. Mais il a fait des pointers. Ce que nous a donnée des mauvais réponses (comme une carrée plain de 3 pions, disant qu'il est vide, ce qui est complètement l'inverse de la réalité)

Alors on voulait que ces informations liées à chaque carré de soient copiés dans la mémoire, Après plusieurs recherches, on a choisi d'utilisé un dépendance Library JSON crée par google qui a la méthode utilisée pour les copier comme ce qu'on le veut.

On estime que la raison d'un tel problème est la récursivité des fonctions qui peuvent arriver aux milliers des fois est de même avoir un meilleur coup dans un match qui n'est toujours pas commencé donc pas dans la liste de possible autorisés coup actuelle.

Donc on a fait une copie du meilleur coup obtenue actuellement et de la liste des coup autorisées.