Intelligence Artificielle

Rapport de projet

Jeu du puissance 4

L3 Informatique 2022-2023

AMOZIEG Shirel

OUALI Anaïs

SOMMAIRE

1. INTRODUCTION

Le présent rapport constitue un compte rendu détaillé du projet intitulé **"Puissance4"**. Il vise à fournir la description de son déroulement et de son organisation. Il comporte les éléments suivants :

- Une description du jeu programmé, et de ses règles
- Une description détaillée des IAs implémentées, et les explications des choix effectués
- Le résultat des tournois entre les IAs implémentées
- Un bilan du projet

Pour sa réalisation, nous avons formé un groupe de deux étudiantes :

- Anaïs OUALI
- Shirel AMOZIEG

Ce projet s'inscrit dans le cadre de l'UE Intelligence Artificielle du niveau Licence 3 informatique.

2. CONTEXTE

De nos jours, il est très courant que les jeux informatiques incluent des intelligences artificielles (IA) pour fournir aux joueurs une expérience de jeu plus riche et plus immersive, notamment les jeux de stratégie en temps réel (RTS). Ces derniers utilisent souvent des IAs pour simuler des comportements réalistes des unités et des ennemis contrôlés par l'ordinateur, rendant ainsi le jeu plus difficile et plus intéressant. Parmi ces jeux de stratégies nous pouvons citer le jeu du puissance4 (ou Connect4 en anglais).

Nous avons choisi de travailler sur le développement de ce jeu de stratégie car il s'agit d'un jeu que nous apprécions beaucoup toutes les deux et dont les règles nous sont très familières. Cet avantage nous a permis de ne pas s'attarder sur la compréhension du jeu et de nous concentrer rapidement sur son développement. De plus les erreurs de règles du jeu nous ont été très facile à repérer et à corriger compte tenu de notre connaissance de celles-ci.

Ainsi, nous vous présenterons le jeu du **Puissance4**.

3. DESCRIPTION DU JEU ET DE SES REGLES

Le jeu du Puissance 4 est un jeu de stratégie pour deux joueurs. Il consiste à remplir une grille avec des jetons de sa couleur et d'en aligner 4 avant l'autre joueur.

Voici les règles du jeu :

1. Le jeu se joue sur une grille verticale de 6 lignes et 7 colonnes.
2. Chaque joueur dispose d'un ensemble de jetons d'une couleur qui lui est attribuée (habituellement jaune et rouge).
3. Les joueurs jouent à tour de rôle en plaçant un jeton dans l'une des colonnes de la grille.
4. Le jeton tombe alors dans la ligne de niveau le plus bas disponible de cette colonne.
5. Le but du jeu est d'aligner quatre jetons de sa propre couleur horizontalement, verticalement ou en diagonale.
6. Le premier joueur qui réussit à aligner quatre jetons de sa propre couleur gagne la partie.
7. Si toute la grille est remplie sans qu'il y ait d'alignement de quatre jetons d'une même couleur, la partie est déclarée nulle.

4. STRUCTURE DU JEU

Notre code se devise en 4 parties :

- **Interface :**

Après avoir initialisé toutes les variables nécessaires au bon fonctionnement du jeu, nous avons implémenté les fonctions permettant de gérer les interfaces aussi bien celle graphique que celle dans le terminal. Elle comporte donc les fonctions de création du plateau de jeu, une grille de 6 rangées et 7 colonnes, dans laquelle les joueurs déposent leurs jetons à tour de rôle. Grâce à l'importation de la bibliothèque « pygame », nous avons pu rendre plus vivant le déplacement des jetons au-dessus des colonnes, ce qui est un plus pour améliorer l'expérience du joueur.

- **Vérification :**

Dans cette partie se trouvent toutes les fonctions vérifiant le bon fonctionnement du jeu dans le respect de ses règles. Elles permettent de vérifier si un joueur peut déposer son jeton à l'emplacement choisi, s'il a gagné ou si la partie est terminée.

Elle possède également des fonctions de calcul du son score des joueurs en fonction de leurs dépôts.

- **IA :**

Pour ce jeu, trois niveaux d'IA ont été implémentés.

  - **Niveau 1** : Pour ce niveau d'implémentation, il s'agit d'une méthode de calcul retentant le meilleur dépôt qu'il est possible d'effectuer par l'IA. Ce niveau n'utilise donc pas une stratégie de calcul en coups d'avance mais plutôt une stratégie de calcul de score qu'elle obtiendra selon son dépôt.

  - **Niveau 2** : Ce niveau utilise l'algorithme « minimax ». Il s'agit d'un algorithme utilisé dans la théorie des jeux pour déterminer le meilleur coup à jouer dans une situation de jeu à deux joueurs avec des résultats possibles. Il repose sur le principe de maximisation du gain pour l'un des joueurs et de minimisation de la perte pour l'autre. Son principe est d'explorer toutes les possibilités de coups dans l'arbre de recherche jusqu'à une certaine profondeur, en évaluant les gains ou les pertes possibles pour chaque joueur. Pour chaque nœud de l'arbre, il choisit le coup qui maximise le gain du joueur actuel si c'est son tour de jouer, et minimise la perte si c'est le tour de son adversaire. En remontant l'arbre de recherche, l'algorithme Minimax détermine finalement le meilleur coup à jouer à chaque niveau de profondeur.

  - **Niveau 3** : Ce niveau utilise la méthode d'élagage alpha-bêta qui est une amélioration de l'algorithme minimax. Elle permet d'accélérer la recherche en éliminant les branches de l'arbre de recherche qui ne sont pas utiles pour déterminer le meilleur coup à jouer. Son fonctionnement est le suivant : elle garde une paire de valeurs appelées alpha et bêta à chaque niveau de l'arbre de recherche. La valeur alpha représente la meilleure valeur connue pour le joueur maximisant, tandis que la valeur bêta représente la meilleure valeur connue pour le joueur minimisant. Lors de l'exploration de l'arbre de recherche, si la valeur évaluée pour un nœud dépasse la valeur bêta actuelle, le reste de la branche peut être ignoré, car le joueur minimisant ne choisira jamais cette branche. De même, si la valeur évaluée pour un nœud est inférieure à la valeur alpha actuelle, le reste de la branche peut être ignoré, car le joueur maximisant ne choisira jamais cette branche. Ainsi, cette méthode permet de réduire considérablement le temps de calcul nécessaire pour trouver le meilleur coup à jouer, et il est donc plus facile d'avoir une IA procédant à une recherche de profondeur plus élevée.

  - **Niveaux plus élevés :** Les niveaux d'IA peuvent être améliorés en modifiant la profondeur de recherche de l'algorithme. Plus la profondeur est élevée, plus son niveau de recherche augmente.

- **Game :** cette partie est responsable de lancer le programme du jeu et d'interagir avec l'utilisateur. Avant le début de chaque partie, elle invite l'utilisateur à choisir le mode de jeu auquel il souhaite jouer :
  1- deux joueurs humains,
  2- joueur contre IA
  3- IA contre IA

  S'il choisit le mode 1, chaque joueur dépose ses jetons aux emplacements qu'il souhaite et le programme ne s'occupe que de vérifier si un des joueurs a gagné ou si la partie est terminée.

  S'il choisit les modes 2 ou 3, il devra renseigner au programme le niveau de difficulté de ou des IAs qu'il souhaite voir jouer pour la partie.
  Notre programme fonctionne de la manière suivante : le premier joueur dépose son premier jeton. S'il a sélectionné le mode **1-** ou **2-**, il y a au moins un joueur humain, le déplacement du jeton au-dessus des colonnes lorsqu'il choisit son dépôt est activé. Les fonctions de vérification sont appelées.
  Dans le cas où une IA est un des joueurs, la fonction correspondant à son niveau de difficulté est appelée pour définir la colonne de dépôt appropriée. Ensuite les fonctions de vérification sont appelées comme pour le mode 1-.

5. RESULTATS DES TOURNOIS ENTRE IAS

En lançant notre programme, il nous est demandé de choisir le mode de jeu de la partie. Le mode **3-** est un mode permettant de lancer un tournoi être 2 IA en choisissant le niveau de chacune d'elles. Chaque IA appelle la fonction correspondante à son niveau à chaque dépôt de jeton.

- **IA niveau 1 vs IA niveau 2 :** L'IA de niveau 1 calcule le meilleur dépôt en temps réel, c'est-à-dire qu'elle ne va chercher à savoir ce qu'il va se passer selon les différents dépôts qu'elle fera. Elle se contente de faire augmenter son score actuel en se basant uniquement sur le passé. L'IA de niveau 2 au contraire calcule le meilleur score qu'elle pourrait obtenir en se basant sur l'analyse des prochains dépôts possibles jusqu'à une certaine profondeur. De ce fait l'IA de niveau 2 a plus de visibilité sur le jeu et donc peut mieux évaluer et prévoir les coups gagnants. Le vainqueur de ce tournoi est donc l'**IA niveau 2.**

- **IA niveau 1 vs IA niveau 3 :** L'IA de niveau 1 fait ses calculs comme au tournoi détaillé ci-dessus. L'IA de niveau 3 utilise quant à elle l'algorithme « minimax »

comme celle du niveau 2 mais en améliorant sa recherche. En effet, au lieu de calculer le meilleur score en parcourant tout l'arbre de recherche, elle minimise son analyse en enregistrant et en mettant à jour des valeurs « alpha » et « bêta » s'autorisant ainsi à ne parcourir que les branches nécessaires optimisant ainsi son temps de calcul. Le vainqueur de ce tournoi est donc l'**IA niveau 3.**

- **IA niveau 2 vs IA niveau 3 :** Comme expliqué plus haut, ces 2 IA utilisent l'algorithme minimax pour déduire le coup qui leur rapportera le meilleur score. Mais l'IA de niveau 3 utilise en plus l'élagage « alpha-bêta », ce qui lui permet de parcourir moins de nœuds de l'arbre de recherche et donc utiliser ce temps gagné pour augmenter la profondeur de recherche. Ainsi, elle peut évaluer ses coups plus loin dans le futur et cette vision lui permet de mieux jouer face à l'IA de niveau 2. Le vainqueur de ce tournoi est donc l'**IA niveau 3.**

6. BILAN

Ce projet consistait à développer une intelligence artificielle capable de jouer au Puissance 4. Pour ce faire, nous avons utilisé l'algorithme Minimax avec élagage alpha-bêta pour évaluer les coups possibles et déterminer le meilleur coup à jouer.

Nous avons commencé par concevoir la représentation de l'état du jeu sous forme de matrice, qui nous permet de stocker la position des jetons et de mettre à jour l'état du jeu à chaque coup. Nous avons ensuite implémenté l'algorithme Minimax et rajouté l'élagage alpha-bêta en utilisant une fonction d'évaluation pour attribuer un score à chaque état du jeu.

Pour tester notre IA, nous avons mis en place un environnement de simulation qui permet de faire jouer une IA contre un joueur humain ou de faire jouer deux IA l'une contre l'autre.

Les résultats de nos tests montrent que notre IA est capable de jouer au Puissance 4 à un niveau avancé, en gagnant la plupart du temps contre des joueurs débutants et en donnant une bonne résistance contre des joueurs expérimentés. Cependant, nous avons également identifié certaines limitations de notre IA, notamment dans sa capacité à anticiper les coups à long terme et à s'adapter à des situations inattendues.

En conclusion, notre projet a été un succès dans la mesure où nous avons réussi à développer une IA capable de jouer au Puissance 4 à un niveau avancé en utilisant des techniques d'intelligence artificielle. Cependant, il reste des améliorations possibles, notamment en explorant des approches plus avancées telles que les réseaux de neurones ou l'apprentissage par renforcement.

7. RESSOURCES

- Slides du cours
- Algorithme minimax : https://fr.wikipedia.org/wiki/Algorithme_minimax
- Algorithme alpha-bêta : https://fr.wikipedia.org/wiki/%C3%89lagage_alpha-b%C3%AAta
- Documentation Pygame : tps://devdocs.io/pygame/
- Tutos YouTube:

  https://www.youtube.com/watch?v=Zkwe-0X3NcM&t=6896s

  Faire un jeu puissance 4 like - en PYTHON

  Coding Test du puissance 4 en TDD (python 3, unittest, poetry & vscode)

8. ANNEXE

Le fichier « IAconnect4.py » dans lequel se trouve notre code est déposé avec ce présent rapport.