

Info-f-106

Arthur Nanson - 000476431

15 avril 2020

Table des matières

1	Introduction	2
2	Améliorations du jeu	3
3	Améliorations de l'IA	4
4	Conclusions	5

Chapitre 1

Introduction

La partie 4 du projet d'informatique (Info-f-106) portait en grande partie sur l'amélioration de l'intelligence artificielle implémentée lors de la partie 2 de ce même cours. Cette intelligence artificielle se nomme "Minimax".

Minimax est un algorithme récursif (c'est à dire que c'est une fonction qui s'appelle elle même directement ou indirectement) qui vise à simuler tous les coups possibles, puis, pour chaque coup, simuler les coups possibles de l'adversaire. L'intelligence artificielle peut aussi continuer plus loin en simulant tous les coups possibles pour répondre à tous les coups de l'adversaire mais cet algorithme devient alors très lent. C'est une efficacité d'algorithme exponentielle : plus on a de couches de récursivité, plus l'algorithme prendra de temps.

Pour chaque coup possible de l'intelligence artificielle, on regarde si ce coup la fait gagner. Si oui on renvoie "1" qui signifie la victoire, sinon on regarde les coups de l'adversaire si il nous reste des appels récursifs. Si l'adversaire a un coup gagnant, on renvoie "-1" qui signifie qu'on ne doit surtout pas jouer ce coup. Si aucun de ces cas ne s'est présenté, on renvoie "0". Dans le cas ou plusieurs coups auraient le même score, on en prend un au hasard.

Nous avons donc pour mission d'améliorer cette intelligence artificielle afin de se mesurer à celle des autres étudiants. Dans ce but, on nous a proposé une demi douzaine de façons différentes de faire. Certaines visent à augmenter l'efficacité de l'intelligence artificielle en améliorant son temps de réponse afin d'ensuite pouvoir lui demander plus, par exemple demander plus de couches récursives lors du test des coups. D'autres visent à faire "réfléchir" cette intelligence artificielle en lui rajoutant des conditions ou en lui faisant un réseau de neurones.

L'autre partie de cette partie 4 consistait en une amélioration du jeu en implémentant un timer pour que les intelligences artificielles ne puissent pas prendre un temps trop important pour choisir leur coup.

Nous avons aussi dû implémenter un système qui regarde si le plateau que l'on veut jouer n'a pas déjà été joué pour éviter les parties infinies ou chaque intelligence artificielle joue le meilleur coup pour elle en boucle.

On nous a aussi demandé d'implémenter un bouton qui sert à enregistrer une partie ainsi qu'un autre qui sert à jouer une partie enregistrée.

Chapitre 2

Améliorations du jeu

Pour l'amélioration du code j'ai commencé par le timer qui limite le temps que les intelligences artificielles peuvent prendre pour choisir leur meilleur coup possible. Pour faire ça, j'ai utilisé la librairie `time` de python. Grâce à la fonction `time` de la librairie `time` (`time.time()`), j'ai récupéré l'heure actuelle en seconde. A la fin du tour de l'intelligence artificielle, je refais un `time.time()` et je soustraits le premier au deuxième. Si la différence entre les deux est plus grande que le temps donné aux intelligences artificielles (en l'occurrence une seconde), je mets fin à la partie en disant que l'intelligence artificielle a prit trop de temps pour trouver son meilleur coup.

Ensuite, j'ai travaillé sur la vérification que deux plateaux identiques ne puissent pas être joués durant la même partie avec le même vecteur de décalage. Pour faire ça, je sauvegarde une copie de chaque plateaux et de chaque vecteurs de décalage dans une liste (je les sauvegarde comme un couple dans un tuple). Pendant la recherche de tous les coups possibles, j'ai donc une condition qui vérifie que le coup que l'on veut jouer ne résultera pas en un plateau déjà utilisé. Si c'est la cas, on retire ce coup avant d'appeler la fonction de l'intelligence artificielle. Le fait de sauver tous les plateaux déjà joués dans une grande liste et de parcourir cette liste pour chaque action possible demande beaucoup de temps et n'est pas le moyen le plus efficace. J'avais pensé à convertir chaque plateau en un string composé de zéros, de uns et de deux, ça aurait sûrement aidé à alléger la mémoire mais vu que l'on doit quand même convertir le plateau à chaque fois, je ne pense pas que l'efficacité de l'algorithme aurait été grandement affectée.

La dernière chose que j'ai voulu implémenter mais que je n'ai pas réussi à faire sont les fonctionnalités d'enregistrement. J'ai créé les boutons mais je n'ai pas réussi à sauvegarder un enregistrement. Pour le lecteur de partie il permet de naviguer dans les fichiers de l'ordinateur mais ne permet d'afficher que des images et non des vidéos.

Nous devons aussi changer toutes nos fonctions en différentes classes correspondantes mais comme j'ai pris le corrigé de la partie 3, je n'ai pas eu à le faire.

Chapitre 3

Améliorations de l'IA

Pour améliorer mon intelligence artificielle j'ai choisit la technique de l'objectif. J'ai donc créé une fonction qui donne un objectif à l'intelligence artificielle et qui augmente ou diminue son score en fonction de son approche de réussite de cet objectif.

Dans un premier temps j'ai choisit comme objectif le fait d'avoir plus de pions dans chaque ligne que son adversaire. Donc si mon intelligence artificielle a plus de pions dans une ligne, le score augmente. Il ne change pas si le nombre de pions est égal et il diminue si l'adversaire en a plus. Cette intelligence artificielle se concentre donc beaucoup plus sur le fait de contrôler le plateau avec beaucoup de pions plutôt que de jouer avec le décalage des lignes qu'elle n'utilise que quand elle peut gagner comme ça. Pour faire ça, je regarde ligne par ligne qui a le plus de pions et j'augmente le score en fonction du résultat de chaque ligne. J'augmente le score de 0.2 par ligne où mon intelligence artificielle a plus de pions et je le diminue de 0.2 quand elle en a moins que son adversaire. Le score peut donc varier entre -0.8 et 0.8 pour chaque coup (ce qui est moins qu'un coup gagnant et plus qu'un coup perdant), car 4 fois 0.2 = 0.8.

J'ai ensuite amélioré cette technique en instaurant une autre manière d'augmenter le score du mouvement, en plus de comparer le nombre de pions dans chaque ligne, je le compare aussi dans chaque colonne. Avec cet ajout, l'intelligence artificielle ne se soucie vraiment plus du fait de décaler une ligne et se concentre exclusivement sur le fait de contrôler le plateau en nombre de pions (sauf quand un coup gagnant est repéré avec un décalage). Mais cet ajout est moins important que le précédent car le vecteur de décalage n'est pas pris en compte dans ce code, ce qui rend son impact plus faible que la première technique instaurée.

J'ai d'abord essayé de combiner ces deux techniques avec le même coefficient mais je me suis vite rendu compte que la première technique était plus performante. J'ai donc essayé plusieurs coefficients pour les deux techniques et celui qui avait le meilleur taux de victoire est un coefficient de 0.7 fois le score de la première technique et de 0.3 fois la deuxième technique.

Le score final est donc : $\text{score} = 0.7 \times \text{technique-ligne} + 0.3 \times \text{technique-colonne}$.

Chapitre 4

Conclusions

Cette partie 4 du projet à été réalisée en rencontrant quelques difficultés (surtout au niveau du bouton pour enregistrer une partie et de la fonction qui devait servir à lire une partie déjà terminée) mais dans l'ensemble tout s'est plutôt bien passé.

La gestion du temps par timer semble bien fonctionner et je n'ai pas trouvé de manière plus optimale de faire ça. Par contre, comme dit précédemment, il est possible d'améliorer la vérification des plateaux identiques lors d'une même partie, mais je n'ai pas réussi à aller au bout de mon idée et je ne suis pas sûr que ce soit la bonne non plus.

Concernant mon intelligence artificielle, elle à été crée sans trop de soucis et je suis plutôt content du résultat car je m'étais fixé comme objectif de réussir à créer une intelligence artificielle que je n'arriverais pas à battre, ce qui est chose faite! Sur une centaine de parties jouées contre l'algorithme Minimax programmé précédemment, mon intelligence artificielle n'a pas perdu une seule fois et elle bat toujours son adversaire avec un très bon contrôle du plateau (ce qui est son but premier avec l'implémentation du nouveau code).

Comme mon intelligence artificielle reprend l'algorithme Minimax, j'aurais pu descendre plus loin dans les couches de récursivité en instaurant un timer qui nous aurait dit quand on s'approchait un peu trop dangereusement de la limite de temps accordée aux intelligences artificielles pour choisir leur meilleur coup. Mais j'avais peur qu'en faisant ça mon algorithme soit moins performant et ne puisse plus trouver les meilleurs coups, soit à cause d'une erreur de ma part soit car les deux algorithmes n'auraient peut être pas été compatibles l'un avec l'autre.

Pour conclure, cette partie du projet n'est pas parfaite, il manque une ou deux fonctionnalités qui auraient permis de perfectionner ce code mais dans l'ensemble tout y est et je suis plutôt content du rendu final, autant le code que mon intelligence artificielle. Je suis évidemment impatient de voir les résultats de mon intelligence artificielle durant le tournoi!