

1 ^{er} cycle Grenoble INP - ESISAR	Compte rendu PXSI PX221-Info	Alexandre VINHAS Guillemot MOUSSU
---	------------------------------	--------------------------------------

Description du projet – PX221 – 3 à 4 pages maximum

Vous devrez respecter ce format de rendu.

1 - Justification de vos choix (ce ne sont pas uniquement les choix en eux-mêmes qui seront évaluées, mais bien leur justification)

Vous devez ici justifier vos choix d'implémentation concernant les points suivants:

1- Algorithme choisi, et modalités d'implémentation

Pour déterminer un coup, notre programme va tester tous les coups possibles puis évaluer les positions qu'il obtient afin de déterminer lequel sera le meilleur. S'il y en a plusieurs ex-aequo on choisit aléatoirement l'un des choix. Cela permet d'éviter de jouer les mêmes parties en boucle, et potentiellement de perdre plusieurs fois la même partie. Ensuite, il retourne le coup qu'il a joué afin que la fonction main l'envoie au serveur.

Pour jouer un coup, et en même temps déterminer s'il est valide, on regarde d'abord si la case est vide, puis on teste dans les 8 directions si ce coup retournerait des pions adverses. S'il est légal, il ne reste ensuite qu'à poser le jeton.

Pour évaluer un coup avec précision, on n'évalue pas la position qu'il nous donne mais on anticipe les coups suivants pour évaluer les positions futures. On détaillera l'évaluation dans la stratégie de jeu.

Pour anticiper les coups, nous avons implémenté l'algorithme minimax, en convention negamax, avec l'élagage alpha-beta. Cet algorithme nous permet d'anticiper plusieurs coups avant d'évaluer les positions obtenues. La convention negamax nous paraissait plus simple car elle évitait de se soucier du joueur qui avait le trait, il n'y avait que le joueur actuel, et le joueur adverse. L'élagage alpha-beta nous permet quant à lui d'optimiser le fonctionnement en n'analysant non pas tous les coups possibles mais seulement les branches les plus pertinentes.

2- Structure de données utilisées

La structure interne du programme (qui sert à déterminer les coups à jouer) n'est composée que de 17 octets. Les 8 premiers servent à stocker la présence des pions sur les cases, les 8 suivants à stocker leur couleur. Enfin, le dernier stocke le joueur qui a le trait ainsi que des coordonnées que les fonctions se passent.

Cette structure est la plus légère que nous ayons imaginé pour stocker les informations dont nous avons besoin. La petite taille de cette structure nous permet d'être efficace lors de l'exécution, car elle est copiée des millions de fois lors de l'appel récursif pour l'anticipation des coups.

3- Stratégie de jeu

Pour évaluer une position, on se base principalement sur 3 critères : la position des pions sur le plateau, les coins acquis par les joueurs car ils sont des zones stratégiques importantes pour contrôler la zone alentour, et le nombre de pions de chaque joueur car c'est la seule chose qui importe en fin de partie.

2 – Performance obtenues

1- Indiquer ici les performances obtenues par votre programme

Le programme est capable de se connecter au serveur et d'y jouer des parties. Il répond au cahier des charges sur ce qui concerne le déroulé des parties. Enfin, l'algo peut anticiper 7 coups en avance car aller plus loin risquerait de prendre trop de temps, ce qui risque de nous faire perdre des parties par timeout.

2- justifier si c'est le cas les raisons techniques pour lesquelles les performances obtenues sont différentes de celles attendues

Nous aurions aimé pouvoir le pousser à une profondeur de 8 mais malgré les nombreuses optimisations de notre part, l'ordinateur reste trop lent, ce qui est compréhensible au vu de la complexité exponentielle de l'anticipation des coups.

3 – Introspection de ses pratiques

Indiquer ici le degré de difficulté RELATIF des point suivants (sur une echelle de 0 (très facile) à 5 (très difficile)), pour cela vous ne pouvez cocher que 18 cases (faites des choix)

- ■ □ □ - organiser son temps de travail personnel de manière à avoir les livrables à temps
- □ □ □ - organiser le travail avec son binôme
- ■ □ □ - utiliser le langage C
- □ □ □ - concevoir le programme pour le rendre maintenable, propre en concis
- ■ ■ □ - comprendre l'algorithme à utiliser
- □ □ □ - debugger le programme en utilisant des outils adaptés
- ■ ■ □ - comprendre ce qui était attendu dans ce projet
- □ □ □ - trouver ou maintenir la motivation pour le projet

4 – champ libre, indiquer les points positifs, les points d'amélioration possibles pour les années suivantes.

Le projet était intéressant, ludique, peut-être un peu compliqué mais c'est l'objectif

L'idée d'un tournoi motive la plupart des groupes à s'impliquer dans le projet

Pour les années suivantes, ce serait peut-être mieux de découper le projet en plus petits morceaux (ex : mécanique de jeu, parcours de l'arbre, et connexion serveur), et fixer au moins une échéance de mi-projet pour faire un point et passer d'une étape à la suivante