Victor Goubet
Thomas Graff
TD D

Rapport

Le programme fournit est un MinMax intégrant un élagage Alpha-Beta et une fonction heuristique. Nous pouvons nous pencher sur ses quelques spécificités :

$Action$:

Pour la fonction action on préfère parcourir chaque colonne en partant du bas. En effet cela permet lors des premiers coups d'arriver à la tête de pile plus rapidement et donc d'arrêter la boucle plus rapidement. Cela est d'autant plus utile du fait que c'est lors des premiers coups que le programme est le plus long à réfléchir.

$MinMax\ -MaxValue-MinValue$:

Pour chaque choix d'une action on ne retourne pas uniquement sont score mais également la profondeur de cette action on a donc un résultat de la forme [score, profondeur]. Cela est utile dans le cas ou on doit faire le choix entre plusieurs actions de même score. Ici on choisira l'action qui est la moins profonde. Cela a du sens car plus une action est profonde plus on a accumulé d'erreur à cause de nos hypothèses. On prend donc l'action avec le moins d'approximation.

$TerminalUtility$:

Ici on a choisi de regrouper la fonction terminal et utility en une seule fonction. En effet dans le cas du puissance 4 la fonction d'évaluation des scores reprend certaines mécaniques utilisées dans Terminal. On retourne donc à chaque fois un résultat du type $[IsTerminal, Score]$.

$Heuristique$:

Pour évaluer une grille on va évaluer chacune des actions possibles au tour suivant. Pour évaluer une action on va se mettre premièrement dans la peau du joueur locale, puis, dans celle de l'adversaire. On aura donc pour chaque action une $FitnessLocal$ et une $FitnessAdversaire$. Afin d'évaluer le score total de la grille on somme les $FitnessLocal$ et on soustrait les $FitnessAdversaire$. Cependant on ajoute un coefficient d'attaque devant la somme des $FitnessLocal$ et un de défense devant celle des $FitnessAdversaire$.

On a donc l'expression du score d'une grille :

$$score = \sum_{\forall\, a \in actions(s)} (\, CA * fitnessLocal(a) - CD * fitnessAdversaire(a)\,)$$

Pour obtenir ces deux fitness on appelle la fonction $EvalAction(a)$ qui retourne un tableau « Fitness » du type $[FitnessLocal, FitnessAdversaire]$.

Ce tableau Fitness est calculé de la façon suivante :

$$Fitness = \sum_{x \epsilon [L,C,D1,D2]} EvalLCD(x, [joueur1, joueur2])$$

Il reste donc à savoir comment calculer la fitness d'une LCD. Pour cela on définit une fonction $EvalLCD$

Cette fonction prend en entrée une Ligne Colonne ou Diagonale ainsi que les joueurs et retourne un résultat du type $[FitnessLocal, FitnessAdversaire]$ *(juste pour la LCD).*

Pour calculer chacune des deux fitness on commence par réduire la LCD à la limite de jouabilité.

Par exemple si on a la ligne : $[.|.|X|O|.|action|.|O]$ on la restreint à $[X|O|.|action|.|O]$

Ensuite, on va traiter chaque zone de potentiel alignement, dans cet exemple on en a 3 :

- $[X|O|.|action]$
- $[O|.|action|.]$
- $[.|action|.|O]$

Pour chacun des ces cas on va regarder s'il est possible de gagner et si oui, combien de pions ont déjà été placés. On augmentera alors le fitness en conséquence.

Ici on voit que dans le 1er cas il est impossible de gagner pour les deux joueurs, on retourne donc $[[False, none], [False, none]$ et on n'augmente pas les fitness.

Dans le 2ème cas, le joueur O peut gagner et il y a déjà un pion de placer, on retourne donc $[[True, 1], [False, none]$. On augmente donc le fitness du joueur O.

Pour cela on utilise le modèle polynomial suivant :

$$Fitness += 3^{nombre\ de\ pions}$$

Ce modèle permet de donner beaucoup plus d'importance à une forte présence de pions.

Pour finir dans le 3ème cas, le joueur O peu gagner on retourne $[[True, 1], [False, none]$.

On augmente donc encore le fitness du joueur O qui était déjà à 3 et qui passe donc à 6.

On a finalement pour la ligne une FitnessO de 6 et une FitnessX de 0.

Afin d'obtenir la fitnessO et X totale il faudrait répéter l'opération pour la colonne et les deux diagonales associées à cette action.

Enfin, dans un souci de temps la $ProdondeureMax$ a été définit à 3 pour atteindre 8.5sc.