

Rapport IA Morpion

Codeurs :

MONNOYEUR Victor

MESSAOUDI Rayan

MERCERON Annabel

MOISON Romane

Optimisation de l'algorithme :

Avec l'algorithme MinMax, notre IA marche bien et trouve le meilleur coup si on lui laisse assez de temps et si on remplit assez de cases. Cependant, pour un plateau peu rempli, notre IA n'est pas capable de trouver une solution en moins de dix secondes. Nous avons ainsi utilisé trois méthodes pour optimiser notre IA.

Tout d'abord, nous lui avons incorporé l'élagage Alpha Béta qui permet de réduire le nombre de nœuds évalués, sans détériorer la solution. Ainsi, notre IA pouvait retourner l'un des meilleurs coups à jouer en moins de temps (environs deux fois moins de temps en moyenne).

Nous avons ensuite fait un élagage « en profondeur ». Cela permet à l'IA de ne pas regarder les nœuds trop profond dans l'arbre. Avec l'élagage en profondeur, nous pouvons diminuer le temps de calcul de notre IA autant que nous le voulons (en diminuant la profondeur qui est un paramètre de notre IA), mais cela dégrade en même temps la qualité de la solution. Il fallait ainsi une profondeur d'environ dix pour avoir une solution en moins de dix secondes (ce qui est peu).

Cependant, la solution renvoyée par l'IA était alors médiocre. Nous lui avons ainsi ajouté une heuristique. Cette dernière prend en paramètre la liste des actions possibles sur le plateau, attribue une note à chaque action, puis renvoie la liste d'actions triée par note. L'IA regarde ainsi en premier les actions qui ont de fortes chances d'être bonnes, ce qui améliore l'efficacité de l'élagage Alpha Béta. Mais la solution renvoyée en moins de 10 secondes n'était toujours pas satisfaisante. Nous avons donc décidé que l'heuristique ne renvoie pas toutes les actions triées, mais seulement une dizaine d'actions triées. Notre heuristique attribue dans un premier temps une excellente note aux « coups évidents ». C'est-à-dire les cas où nous gagnons en un seul coup, où nous controns une défaite en un coup, où l'on gagne en deux coups, où nous controns une défaite en deux coups. Puis, l'heuristique augmente ou diminue les notes des actions en fonction de leur position sur le plateau, de la densité de pion et de l'alignement de même pion autour d'elle. L'heuristique étant peu efficace pour un plateau complètement vide, l'IA n'utilisera pas l'algorithme MinMax et jouera vers le milieu du plateau dans ce cas.

Grâce à toutes ces modifications notre IA renvoie une bonne solution si le plateau est peu rempli, et la meilleure solution si le plateau est assez rempli, tout cela en moins de dix secondes.

Difficultés rencontrées :

- déterminer si telle ou telle approche étaient considérer comme un dictionnaire de coup
- comprendre le pseudo programme de MinMax et l'élagage alpha bêta
- diminuer le temps d'exécution de notre programme