Rendu EDD strategie

Pinero, Borde, Bonnet

6 avril 2015

Table des matières

1 Strategie fonctionnelle

La strategie fonctionnelle choisie pour le rapport est la stratégie du coin. Pour cela, nous avions fait un premier algorithme qui se contente de jouer vers la gauche tant qu'on le peut, sinon il joue vers le bas encore une fois tant qu'on le peut, de même vers la droite puis vers le haut. Avec cet stratégie, sur 48 parties nous avons un score moyen de 2475, avec 6x64, 20x128, 20x256 et 2x512.

Dans un seconds temps, nous avons fait un second algorithme qui est basé sur le premier mais qui un coups sur deux change les mouvements favoris. Les premiers mouvements favoris sont dans cet ordre, gauche, bas, droite et haut, et les seconds mouvements sont dans cet ordre, bas, gauche, droite et haut. Avec cet stratégie, sur 48 parties nous avons un score moyen de 2259, avec 1x32, 3x64, 21x128 et 23x256.

Les deux stratégies se valent à peu prés, mais c'est bien la première qui est la meilleure.

2 La libraire dynamique

3 Evaluation de la grille

Pour évaluer la grille, nous avons pris en compte quatre paramèrtres, et à chancun de ces paramètres nous leurs avons donné un coéficient pour leurs donner plus ou moins d'importance.

3.1 Le nombre de tile vide

On compte le nombre de tile vide que l'on a après avoir joué et on le mutiplie par

3.2 La plus grande tile

3.3 La grille la plus progressive possible

3.4 La grille la plus régulière possible

4 Reflexion avancé sur l'implementation de expected max

« Expected Max » est un algorithme qui consiste à retourner la direction optimale à jouer. Elle est basée sur l'algorithme « minimax ». Le principe est de donner une valeure à la grille en fonction de plusieurs critères (cf. « 3 Evaluation de la grille ») ainsi noud jouerons le coups avec le maximum de chance de gagné combiné au coup avec le minimum de chance de perdre. Ici nous avons un exemple avec le morpion :

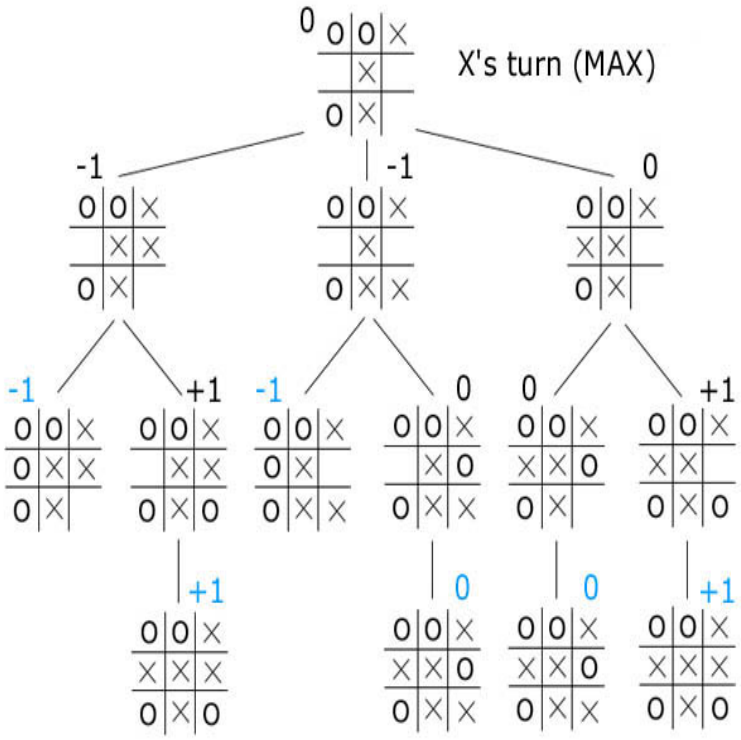

FIGURE 1 – Exemple minimax

Dans la figure 1, on peut voir que le coups le plus interessant à jouer est le troisième, car si on fait la somme de chaque branche de l'arbre (-2, 1, -2, -1, 0, 2), c'est celle-ci qui offre le plus grand score et la perte la moins importante.

L'une des limites de cette méthode est qu'il faille jouer à deux joueurs, or au 2048 on joue tout seul. C'est pour cette raison qu'on utilisera pas le théhorème du « minimax » mais celui d'« Expected Max ». En effet ici on concidèrera le second joueur comme l'ordinateur qui remplira de façon aléatoire une des tiles vides. Cela engage de calculer, pour chaque grille et pour toutes les possibilités de remplissage d'une tile vide, la valeur de cette grille. Il faudra ensuite faire

une moyenne de toutes les valeurs calculés ... (jocker). Pour un algorithme encore plus performant il serait avantageux de pouvoir étendre notre arbre à une profondeur N choisi au début du programme.