

Algorithmique 2 : Puissance 4

- Quentin Dubois
- Sami El Kateb
- Florian Latapie

Fonction de score

Recherche des différentes approches possibles

- score de l'ennemi augmente de manière exponentielle
- score de l'ennemi calculé de manière linéaire

Prise en compte

- Prendre en compte les pions qui peuvent menait à des puissances 4, c'est à dire les combinaisons suivantes

Représentation de la grille

Pour le moment, nous comptons partir sur une matrice contenant des entiers

- 0 : représentant l'absence de pions
- 1 : joueur 1 : humain
- 2 : joueur 2 : machine

Ci-dessous est présentée la variable *possibleWinningMovesOnALine* contient un tableau de tableaux de coups possiblement gagnants, tableau 1 contient les combinaisons à 1 pion, le tableau 2 contient les combinaisons à 2 pions et ainsi de suite.

Possiblement gagnant dans l'idée où il est possible à partir d'uniquement ces positions de pouvoir avancer vers un alignement de 4 pions.

"0120" n'est pas possible, car parmi ces 4 cases, il n'y a pas de possibilité de gagner plus tard dans cette ligne/direction

Afin de trouver pour les colonnes et diagonales, il est possible de faire une transposée et une [transposée à 45°] afin de réutiliser la même formule pour les 4 directions (horizontal, vertical, diagonal 1 et diagonal 2).

```
possibleWinningMovesOnALine = [  
    ["1000", "0100", "0010", "0001"], // 1 pion  
    ["1100", "0110", "0011", "1010", "0101", "1001"], // 2 pions  
    ["0111", "1011", "1101", "1110"], // 3 pions  
    ["0111"] // Gagné  
]
```

De même pour les combinaisons pour le joueur 2, machine.

Score machine

Notre score correspond à un entier permettant d'évaluer une position de jeu par rapport à d'autres. On augmente le score dès qu'on obtient des combinaisons présentes dans *possibleWinningMovesOnALine* pour l'ordinateur. De même, on diminue le score dès qu'on obtient des combinaisons présentes dans *possibleWinningMovesOnALine* pour l'humain.

```
i = column in possibleWinningMovesOnALine

if i < 3
    score += 1 * (i + 1);

if i == 4:
    score += 100000 (big number)
```

Score humain

Dans le cas d'une fonction exponentielle

```
if i < 3
    score -= 1 ** (i + 1)

if i == 4 :
    score -= 100 000
```

Dans le cas d'une fonction symétrique

```
if i < 3
    score -= 1 * (i + 1)

if i == 4 :
    score -= 100 000
```

Complexité de la fonction d'évaluation

Notre fonction d'évaluation est constante. En effet, la taille de la grille est constante et le nombre de pions à aligner aussi.

Nous n'avons donc pas de crainte par rapport à l'explosion de notre fonction d'évaluation au cours de la partie. Nous allons tout de même essayer d'évaluer la constante.

Nous posons les variables suivantes :

- $w = 7$ (largeur de la grille)
- $h = 6$ (hauteur de la grille)
- $p = 4$ (nombre de pion pour gagner)
- $m = 15$ (nombre de motifs qu'on recherche dans la grille)
- $r = 3$ (nombre de rotation)

Notre fonction d'évaluation coûte :

$$2 * (w * p) * m * h * 4 + (3 * w * h) = 2 * 7 * 4 * 15 * 6 * 4 + 3 * 7 * 6 = 20400$$

Rotations de la grille coûtent : $(3 * w * h)$: Nous réalisons 3 rotations de grilles vers la verticale, diagonale droite / diagonale gauche

Calcul du score de la grille : $2 * (w * p) * m * h * 4$