

## **RAPPORT DU TP4 MODULE TPGO**

### **OPTIMISATION D'UN PROGRAMME D'UN** **JEU D'ECHEC**

**2CS SIL2 :**

**BINOME :**

- HEBBACHE LOUCIF
- LEBDIRI ABDERRAHMANE

**ANNEE : 2018/2019**

## 1. Introduction :

Dans ce TP, il est demandé d'optimiser un programme d'un jeu d'échecs, et ceci en améliorant une des (05) fonctions d'estimation données ;

Les fonctions d'estimation sont utilisées lorsque la profondeur des arbres du jeu considérée est importante pour atteindre des configurations feuilles dès le début ; elles permettent alors de retourner des couts estimés sur des configurations non terminales ; Ces fonctions dépendent de chaque jeu et influent de façon importante sur la qualité de jeu de la machine.

**Nous avons choisis la première fonction d'estimation donnée estim1(...) et nous l'avons amélioré par une autre fonction estim(...)** ;

## 2. Présentation de la fonction à améliorer estim1(...) :

La fonction donnée estim1(...) n'est pas bonne car elle est basée uniquement sur le nombre du type de pièces possédées par un joueur, son évaluation est effectuée ainsi :

Elle parcourt la matrice du jeu, et elle calcule le score en calculant la somme pondérée de pièces de chaque joueur ainsi:

$$\text{ScrQte} = ( (\text{pionB} * 2 + \text{cfB} * 6 + \text{tB} * 8 + \text{nB} * 20) - (\text{pionN} * 2 + \text{cfN} * 6 + \text{tN} * 8 + \text{nN} * 20) ) * 100.0/76 :$$

Les poids sont fixés comme suit: pion:2 ; cavalier/fou:6 ; tour:8 et reine:20

Le facteur 100/76 pour ne pas sortir de l'intervalle] -100 , +100[

Puisque le jeu d'échecs est un jeu stratégique :la position d'une pièce représente un autre point de sa force, il est clair que cette fonction estime1(...) n'est pas très efficace puisqu'elle ne prend pas en compte la position de chaque pièce.

### 3. Présentation de la fonction améliorée estim(...) :

Cette fonction est une solution au problème de la précédente (estim1(...)) ,i.e :

Son évaluation est la même, mais elle ajoute au score le poids de la position de chaque type de pièce :

```
for (i=0; i<8; i++)  
  
    for (j=0; j<8; j++) {  
  
        switch (conf.mat[i][j]) {  
            case 'p' : pionB++;  
                ScrQte += piece_square_table[0][i*8+j];  
                break;  
            case 'c' : cB++;  
                ScrQte += piece_square_table[1][i*8+j];  
                break;  
            case 'f' : fB++;  
                ScrQte += piece_square_table[2][i*8+j];  
                break;  
            case 't' : tB++;  
                ScrQte += piece_square_table[3][i*8+j];  
                break;  
            case 'n' : nB++;  
                ScrQte += piece_square_table[4][i*8+j];  
                break;  
        }
```

La table des poids de la position de chaque type de pièce se trouve sur internet ; elle donne le poids d'une pièce selon son type et sa position ;

Ci-dessous des captures de cette dernière :

```

// evaluation table by position
const int piece_square_table [5][64] = {
{
// Pawns - early/mid
// A B C D E F G H
| 0, 0, 0, 0, 0, 0, 0, 0, // 1
-1, -7,-11,-35,-13, 5, 3, -5, // 2
1, 1, -6,-19, -6, -7, -4, 10, // 3
1, 14, 8, 4, 5, 4, 10, 7, // 4
9, 30, 23, 31, 31, 23, 17, 11, // 5
21, 54, 72, 56, 77, 95, 71, 11, // 6
118,121,173,168,107, 82,-16, 22, // 7
0, 0, 0, 0, 0, 0, 0, 0 // 8
},{
// Knights - early/mid
// A B C D E F G H
|-99,-30,-66,-64,-29,-19,-61,-81, // 1
-56,-31,-28, -1, -7,-20,-42,-11, // 2
-38,-16, 0, 14, 8, 3, 3,-42, // 3
-14, 0, 2, 3, 19, 12, 33, -7, // 4
-14, -4, 25, 33, 10, 33, 14, 43, // 5
-22, 18, 60, 64,124,143, 55, 6, // 6
-34, 24, 54, 74, 60,122, 2, 29, // 7
-60, 0, 0, 0, 0, 0, 0, 0 // 8
},{
// Bishops - early/mid
// A B C D E F G H
|-7, 12, -8,-37,-31, -8,-45,-67, // 1
15, 5, 13,-10, 1, 2, 0, 15, // 2
5, 12, 14, 13, 10, -1, 3, 4, // 3
1, 5, 23, 32, 21, 8, 17, 4, // 4
-1, 16, 29, 27, 37, 27, 17, 4, // 5
7, 27, 20, 56, 91,108, 53, 44, // 6
-24,-23, 30, 58, 65, 61, 69, 11, // 7
0, 0, 0, 0, 0, 0, 0, 0 // 8
},{
// Rooks - early/mid
// A B C D E F G H
|-2, -1, 3, 1, 2, 1, 4, -8, // 1
-26, -6, 2, -2, 2,-10, -1,-29, // 2
-16, 0, 3, -3, 8, -1, 12, 3, // 3
-9, -5, 8, 14, 18,-17, 13,-13, // 4
19, 33, 46, 57, 53, 39, 53, 16, // 5
24, 83, 54, 75,134,144, 85, 75, // 6
46, 33, 64, 62, 91, 89, 70,104, // 7
84, 0, 0, 37,124, 0, 0,153 // 8
},{
// Queens - early/mid
// A B C D E F G H
|1,-10,-11, 3,-15,-51,-83,-13, // 1
-7, 3, 2, 5, -1,-10, -7, -2, // 2
-11, 0, 12, 2, 8, 11, 7, -6, // 3
-9, 5, 7, 9, 18, 17, 26, 4, // 4
-6, 0, 15, 25, 32, 9, 26, 12, // 5
-16, 10, 13, 25, 37, 30, 15, 26, // 6
1, 11, 35, 0, 16, 55, 39, 57, // 7
-13, 6,-42, 0, 29, 0, 0,102 // 8
}};

```

La formule du score est donc :

**ScrQte += ( (pionB\*100 + cB\*320 + fB\*330 + tB\*500 + nB\*900) - (pionN\*100 + cN\*320 + fN\*330 + tN\*500 + nN\*900) ) \* 100.0/76 ;**

En testant le programme avec cette nouvelle fonction, on remarque une amélioration claire de la qualité du jeu ;