

Projet IN104 (2025) - Programmation d'une IA pour le jeu d'échecs en C

Encadrant : Tom Boumba

Fiche 6 : Évaluation

Afin d'estimer le meilleur coup possible à jouer dans la position actuelle au moyen de l'algorithme Minimax, nous allons implémenter une heuristique d'évaluation de la position. Dans le cadre de ce projet, nous ne procéderons pas à l'évaluation de la position au moyen d'un réseau de neurones, mais nous utiliserons l'approche HCE (Hand Crafted Evaluation) utilisant des critères basés sur l'intuition, ainsi que sur des principes de stratégie échiquéenne simples. En particulier, on inclura au minimum les critères suivants dans notre fonction d'évaluation :

- l'avantage matériel (i.e la différence de valeur entre l'ensemble des pièces blanches et noires présentes sur l'échiquier)
- l'avantage positionnel (i.e la différence de score positionnel entre l'ensemble des pièces blanches et noires présentes sur l'échiquier)

En fonction du trait, le score sera multiplié par -1 ou non. Par exemple, une position pour laquelle le score est de $+50$ avec trait aux blancs aura pour score -50 si le trait est aux noirs et vice versa. Cette approche de score relatif et non absolu permet de simplifier l'implémentation de l'algorithme Minimax en utilisant sa forme Negamax.

Implémentation

Dans le fichier `eval.h`, on a déclaré puis défini dans `eval.c` les variables globales suivantes :

- `int valeur_piece[12]`, un tableau renseignant la contribution à l'avantage matériel absolu d'une pièce de chacun des types possibles.
- `int score_pos_pion[64]`, un tableau renseignant la contribution à l'avantage positionnel blanc de la présence d'un pion blanc sur chaque case de l'échiquier.
- `int score_pos_cavalier[64]`, un tableau renseignant la contribution à l'avantage positionnel blanc de la présence d'un cavalier blanc sur chaque case de l'échiquier.

- `int score_pos_fou[64]`, un tableau renseignant la contribution à l'avantage positionnel blanc de la présence d'un fou blanc sur chaque case de l'échiquier.
- `int score_pos_tour[64]`, un tableau renseignant la contribution à l'avantage positionnel blanc de la présence d'une tour blanche sur chaque case de l'échiquier.
- `int score_pos_dame[64]`, un tableau renseignant la contribution à l'avantage positionnel blanc de la présence d'une dame blanche sur chaque case de l'échiquier.
- `int score_pos_roi[64]`, un tableau renseignant la contribution à l'avantage positionnel blanc de la présence d'un roi blanc sur chaque case de l'échiquier.
- `int square_miroir[64]`, un tableau permettant d'obtenir l'avantage positionnel noir en inversant les tableaux d'avantage positionnel blancs.

Les valeurs de ces tableaux proviennent de cette page. Il en existe de nombreuses autres implémentations.

- Implémenter la fonction `int eval()`, qui renvoie le score relatif de la position actuelle. Indication : On pourra parcourir les bitboards des pièces de l'échiquier plutôt que ses cases, et utiliser la fonction `ls1b_index` pour calculer l'avantage positionnel.
- Incorporer l'évaluation dans la fonction d'affichage de l'échiquier.
 - Vous pourrez observer le résultat de l'évaluation dans diverses positions, par exemple :

```
parse_fen(position_de_depart);
print_echiquier();
```



Figure 1: Évaluation de la position initiale.

```

parse_fen(tricky_position);
print_echiquier();

```

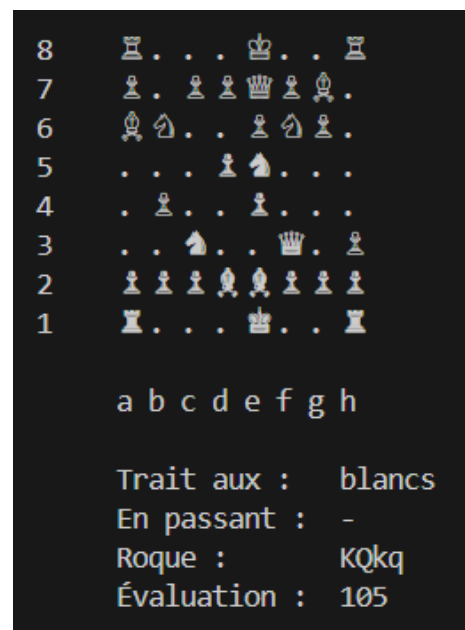


Figure 2: Évaluation d'une position plus complexe.