

Programmation Avancée : Projet jeu d'échec en C++

Rapport et Documentation:

Nom : BENCHEIKH LEHOCINE
Prénom : Mohammed Amine
Groupe: IAD

1. Fonctionnalités implémentées :

A. fonctionnalité obligatoires (required):

- Permettre à deux joueurs locaux de faire une partie de jeu d'échecs.
- Afficher un échiquier en utilisant des lettres alphabétiques pour afficher les pièces.
- Permettre aux joueurs de capturer les pièces de l'adversaire (se fait sans indication du joueur, il suffit de se déplacer vers une case occupée par une pièce de l'adversaire).

B. fonctionnalité d'amélioration (improvements):

- Castling : vous pouvez faire un castling de votre roi en entrant le déplacement "O-O" pour faire un castling avec le rook de côté du roi et "O-O-O" pour faire un castling avec le rook de côté de la reine.
- En passant: vous pouvez faire un en passant lorsque il est possible; aucune règle d'introduction des déplacements n'est imposée.
- Pawn promotion : Lorsqu' un des vos pawns arrive au côté de l'adversaire, vous serez demandé de donner le type de la promotion que vous désirez (Q pour reine, R pour Tour, N pour chevalier).
- Clock system : lorsque vous lancez une partie, vous serez demandé de choisir le type de contrôle de temps que vous préférez. On a implémenté que les modes Fast qui sont mieux adaptés pour un jeu sur ordinateur (un mode classic généralement est plus simple mais prend beaucoup de temps (deux heures pour les 40 premiers coups). Clock fait tout ce qui concerne le contrôle de temps.
- check et checkMate : les deux fonctionnalités sont implémentées, le programme vérifie toujours s'il y a un checkMate pour arrêter le jeu ou un check pour ne pas permettre certains mouvements. La méthode check et checkMate de la class Player implémentent ces fonctionnalités.
- Resign et Draw : dans son tour, un joueur peut demander un resign et arrêter le jeu (son adversaire gagne par conséquent). Aussi, un joueur peut proposer un match nul à son adversaire (la syntaxe est expliquée dans le help menu). le programme peut détecter un match nul en vérifiant si l'un des joueurs est en

situation de StaleMate, un manque du matériel pour faire un échec ou un joueur réclame un match nul à cause d'une répétition d'un état de l'échiquier plus de 3 fois.

- Par défaut, les pièces sont représentées par leurs codes unicodes. Si le joueur veut une représentation basique, il peut l'indiquer lors de la création du jeu.
- Le joueur peut charger un jeu d'un fichier .pgn, sauvegarder son jeu au format .pgn, ou simuler l'historique d'un match à partir d'un fichier .pgn. Les fonctionnalités de format pgn sont implémentées par la classe PGNio (en utilisant que les bibliothèques standard d'entrée sortie).
- Le programme permet à l'utilisateur d'introduire une configuration de l'échiquier par un fichier .txt qui est composé de deux lignes, chaque ligne contient les pièces d'un joueurs et leurs positions (format indiqué dans le menu help).

C. Fonctionnalité optionnelle :

Le programme donne la possibilité de jouer contre un agent intelligent qui utilise l'algorithme MinMax pour faire ses coups. L'algorithme MinMax utilise une fonction d'évaluation qui se base sur la technique des tables des positions des pièces (On donne plus de scores lorsque les pièces sont présente dans des positions spécifiques).

L'agent n'est pas vraiment un bon joueur car la profondeur de la recherche que j'ai utilisé est trop petite (une profondeur plus grande rend l'algorithme très lent).

2. Utilisation de programme : un fichier readme explique comment compiler et lancer le programme.