Cahier des charges – Projet – PolyChess

Contexte

Dans le cadre d'un cours de gestion de projet informatique, on nous a demandé de faire un projet basé sur la création d'un jeu d'échec, tout en veillant sur une gestion du projet efficace et organisée. Par groupe de 4, nous devions réfléchir à comment aborder le projet, sachant qu'il sera délégué l'année prochaine à d'autres étudiants, le projet devra donc être clairement défini dans le but de simplifier le relai.

Objectifs

L'objectif du programme est de créer un jeu d'échec sur python, où un joueur peut jouer contre une IA, en passant par une interface console dans un premier temps, puis potentiellement par une interface plus graphique dans un second temps. L'IA sera d'abord capable de jouer sans réfléchir, puis à terme, elle devra être capable de trouver les meilleurs coups possibles.

Ressources

Nous sommes 4 dans notre groupe, avec des connaissances hétérogènes dans le domaine de la programmation. Nous en tenons compte dans le choix et la répartition des tâches. Nous avons également décidé d'un chef de projet qui devra suivre de près l'avancement globale du projet et qui effectuera des rapports de fin de séances.

Résultats attendus

Les éléments suivants sont ceux que l'on pense finir durant ce projet :

- Représentation des pièces et de l'échiquier. Interface console,
- Définition des coups légaux,
- Boucle moteur (associé aux coups légaux,
- Fin de partie (mat, nul),
- Coups spéciaux,
- L'IA joue (aléatoirement),
- L'IA comprend la valeur des pièces et évalue les n meilleurs coups sur p coups théoriques (arbre probabiliste),
- Implémentation d'une bibliothèque d'ouvertures (Polyglot) avec analyse,
- Implémentation d'une table des fins de partie (Syzygy) avec analyse.

Ce qui restera à faire :

- Coups spéciaux (roque : prise en compte de l'échec lors du déplacement, prise en passant)
- Relier correctement l'analyse des ouvertures, l'analyse des fins de partie, et le calcul probabiliste des meilleurs coups, de sorte à rendre l'IA plus performante,
- Création d'une meilleure interface (sur PyGame ou autre).

Description fonctionnelle

Fonction rules

Elle créer les pièces, leur associe des noms (Roi, Dame, Tour, Fou, Cavalier, pion), des couleurs (blanc ou noir), et le transforme avec un attribut (exemple : Tn pour une tour noire). Elle leur affecte également des valeurs, en fonction de la puissance des pièces, cela sera utile pour créer un arbre probabiliste utilisé ensuite par l'IA pour l'étude des meilleurs coups.

Elle créer aussi une table de débordement, qui est une liste semblable à une matrice de 10x12 avec des coefficients -1 pour les lignes 0, 1, 10 et 11 et les colonnes 0 et 9, et le reste est rempli normalement de 0 à 63 pour la « sous-table virtuelle » de 8x8 à l'intérieur. Une table de placement y est également créée, de la même façon en 8x8, qui correspond aux indices de la table de débordement où il n'y a pas de -1. Ces tables permettront de placer les pièces en fonctions des indices et savoir si un coup est hors du terrain et donc illégal.

En utilisant les règles des échecs et les tables créées précédemment, on implémente des fonctions de coup légaux pour chaque pièce, qui ont en entrée la pièce, sa position, et qui retourne une liste de coups possibles.

Fonction chessboard

Cette fonction permet de créer l'échiquier, pour qu'il soit visible par l'interface console. Elle affiche également les coordonnées (chiffres et lettres), qui simplifieront plus tard l'utilisation pour le joueur. Au départ les pièces sont situées à leur positions initiales. Grace à la fonction main, le joueur pourra sélectionner une case, et dans ce cas la fonction chessboard utilisera la liste des coups possibles pour afficher quels coups sont effectivement possibles. La fonction pourra également déplacer les pièces avec en entrée la pièce, sa position, et la position désirée.

Si un joueur est échec et mat, alors il affiche quelle couleur a perdu et arrête la partie.

Le joueur entrera les coordonnées avec des chiffres et des lettres mais elles seront retranscrites avec un index, de 0 à 63, qui correspond en fait à l'échiquier pour notre programme. Si une pièce choisie d'être déplacée sur une pièce adverse, c'est une prise, et la pièce adverse disparait et est remplacée par la pièce attaquante.

Le programme va regarder combien de pièces menace une pièce désignée, cela est utile dans le calcul des meilleurs coups à l'aide de l'arbre probabiliste : lorsqu'une pièce va manger une pièce adverse, et donc gagner des points, elle en gagnera moins si elle est menacée par des pièces après sa prise.

Fonction main

C'est ici que toutes les fonctionnalités de la fonction chessboard prennent sens. Elle permet la communication entre le joueur et le programme. Pour se faire, elle autorise différentes actions par le joueur : jouer, connaitre les coups possibles, afficher l'échiquier, afficher l'échiquier avec les coups possibles. Il s'agit d'une boucle, à chaque demande, elle exécute les fonctions correspondantes, affiche le résultat, puis redemande au joueur quelle action il veut effectuer.

Après que le joueur ait demandé un déplacement et qu'il a été exécuté, la fonction main va venir interroger l'IA, qui lui retournera un coup en retour.

Fonction IA :

Pour effectuer un calcul des meilleurs coups, on demande une profondeur p et un top n des meilleurs coups pour chaque profondeur. Par exemple si on demande n = 5 et p = 3, la fonction IA va calculer les 5 meilleurs coups en fonction des valeurs des pièces et des menaces potentielles suivant le coup, et ce, pour les trois prochains coups. Le temps de calcul évolue donc énormément si on augmente p et n. Pour connaitre les meilleurs coups à une profondeur plus grande que 1, elle simulera les coups de l'adversaire avec la même méthode de calcule de point, et agira en fonction.

Délais

Le projet a débuté le 27/11/2019, et nous arrêterons de travailler dessus le 10/01/2020, mais le projet en lui-même ne sera pas terminé.