Livrable 1

Projet tuteuré

Mathis DELOGE, Antoine PETOT, Ange PICARD
Arthur CARCHI, Lucas FOUGEROUSE, Vincent DERECLENNE

1 Introduction

Notre projet consiste en la réalisation d'un jeu d'échec doté d'une prise en charge réseau, afin de jouer en ligne, ainsi qu'une intelligence artificielle pour pouvoir jouer seul.

Ce projet n'a pas pour but d'innover ou de proposer quelque chose de fondamentalement nouveau dans la conception et la réalisation d'un jeu d'échec. Nous voulons réunir les matières des modules techniques vues aux S2 et au S3. Ceci nous permettra de programmer un jeu d'échec complet (qui comporte une IHM, une base de données et le tout conçu avec une approche objet) et découvrir le monde de l'intelligence artificielle et la gestion réseau.

2 Existant

2.1 Moteurs

Les moteurs d'échecs sont le cœur des jeux d'échecs. Ce sont eux qui gèrent toutes les contraintes des parties d'échecs et ont un module d'intelligence artificielle intégré.

2.1.1 StockFish

`https://stockfishchess.org/`

Stockfish est un programme d'échecs open source développé par Tord Romstad, Marco Costalba, Joona Kiiski et Gary Linscott. Ce programme est libre et gratuit et est le meilleur logiciel d'échecs non commercial au monde depuis mai 2014. Sur l'ensemble des logiciels, il est considéré comme un des meilleurs avec Komodo 9.3 et Houdini 4.

Ce moteur est conçu pour la performance (il a gagné plusieurs d'IA d'échecs). C'est pourquoi il a été codé en C++, un langage orienté objet basé sur le langage C très apprécié pour sa vitesse d'exécution. Pour modélisé le plateau de jeu, StockFish utilise des bitboards. Ceux-ci permettent de représenter un plateau sous la forme d'une matrice de bits (1 si la case possède une pièce, 0 sinon). Cependant, lors de leur création dans le moteur, on va préférer stocker ces matrices sous la forme d'un entier non signé de 64 bits (puisque le plateau possède 64 cases). Ainsi, le moteur est très rapide et très peu gourmand en mémoire, mais leur implémentation est loin d'être aisée.

2.1.2 Komodo

`https://komodochess.com/`

Komodo est un programme d'échecs commercial et non libre créé par Don Dailey, Mark Lefler et Larry Kaufman en 2007. Il a été considéré comme le meilleur programme d'échecs en 2016.

2.1.3 GNU Chess

`https://www.gnu.org/software/chess/`

GNU Chess est un logiciel libre de jeu d'échecs, sous les termes de la licence publique générale GNU, maintenu par la collaboration de développeurs. Ne disposant que d'une saisie des coups en ligne de commande, il peut être considéré comme un moteur d'échecs. Il est souvent utilisé avec un environnement graphique comme XBoard ou GlChess pour la 3D.

Tout comme StockFish, GNU Chess utilise des bitboards et son intelligence artificielle est basée sur un algorithme de recherche [1] appelé élagage Alpha-Bêta (cf. partie 5.1) et a subi une refonte total pour sa version 5 en 1999, car son code était mal écrit.

2.2 Deep Blue

`https://fr.wikipedia.org/wiki/Deep_Blue`

Deep Blue est un superordinateur conçu par IBM destiné à jouer aux échecs. Il a battu Garry Kasparov, un des meilleurs joueurs d'échecs au monde, en 1997. Ce fut un événement remarquable dans le monde de l'intelligence artificielle, car dès lors, aucun ordinateur n'avait réussi à battre un joueur humain aux échecs, et on pensait que ce ne serait pas possible avant un long moment.

3 Solutions similaires

3.1 WJChess 3d

`https://fr.jeffprod.com/wjchess/`

WJChess 3d est jeu d'échec pour pc fonctionnant sous windows. Il inclut de nombreuses options et fonctionnalités et à la particularité d'avoir une partie graphique assez élaborée et modifiable à volonté. Ce logiciel ne permet qu'un mode 2 joueur ou contre l'ordinateur, il intègre les possibilités de sauvegarder, reprendre une partie ou même créer une position grâce à l'importation / exportation des position au format FEN. De plus, il respecte toutes les règles des échecs et propose plusieurs niveaux de difficulté avec un temps de recherche ou une profondeur fixe pour l'intelligence artificielle.

3.2 Lichess

`https://fr.lichess.org/`

Lichess est un jeu d'échecs par navigateur qui permet de faire des parties contre d'autres joueurs réels, une fois connecté sur le site, ou bien contre une IA avec différents niveaux de difficulté allant de 1 à 10. Le site propose aussi des tutoriels proposant une initiation aux échecs et différents entraînements en fonction de son niveau de maîtrise.

4 Fonctionnalités principales

- Jouer une partie
  - Contre un adversaire
  - Contre une IA
  - Via le réseau
  - Bouger une pièce
  - Voir l'historique des coups
  - Voir les coups possibles
- Sélectionner niveau IA
- Enregistrer une partie
- Reprendre une partie enregistrée

1. un algorithme de recherche est un type d'algorithme qui, pour un domaine, un problème de ce domaine et des critères donnés, retourne en résultat un ensemble de solutions répondant au problème.

5 Technologies

5.1 Préambule

Etant donné que nous ne cherchons pas à créer le moteur d'échecs le plus puissant, ni commercialisé notre application, nous n'inclurons pas d'éléments de programmation avancé (tel que les bitboards) car elles demandent une connaissance approfondi dans ce domaine. Nous souhaitons donc réussir à faire un jeu d'échecs sans avoir de connaissances particulières, excepté pour l'intelligence artificielle car cette partie ne s'improvise pas.

5.2 Langage

Nous avons choisi de développer WinEchek en C#, avec le framework .NET. Ces solutions de développement nous permettent une grande flexibilité grâce à l'abstraction qu'elles fournissent. En effet, cette abstraction nous permettra d'aller droit au but et de ne pas s'arrêter sur des détails d'implémentation notamment pour le réseau ou l'IHM. Le C++ aurait permis une recherche de la performance bien plus poussée, mais peu pertinente dans le cadre de ce projet. Aussi, le Java propose également une panoplie d'outils pour développer rapidement, mais la bibliothèque graphique nous parait moins flexible que celle fournie par le framework .NET (WPF) Ce projet sera l'occasion de nous familiariser avec le C#, très utilisé dans le monde professionnel. Par ailleurs, ce choix nous fera gagner du temps lors de l'implémentation, ce qui nous permettra d'arriver plus vite à la réalisation de l'IA qui est intéressante pour nous.

5.3 Moteur

5.3.1 Partie nulle

- Nulle par matériel insuffisant :
  - Roi contre Roi
  - Roi et Fou contre Roi
  - Roi et Cavalier contre Roi
  - Roi et Fou contre Roi et Fou
- Règle des 50 coups sans prise de pièces
- Pat
- Consentement mutuel
- Triple répétition

6 Annexe

6.1 Webographie

- http://www.ffothello.org/informatique/algorithmes/
- http://wannabe.guru.org/scott/hobbies/chess/
- http://www.chessopolis.com/computer-chess/#info
- http://www.tckerrigan.com/Chess/TSCP/
- https://chessprogramming.wikispaces.com/
- https://fr.jeffprod.com/blog/2014/comment-programmer-un-jeu-dechecs.html
- https://github.com/Tazeg/JePyChess
- http://codes-sources.commentcamarche.net/source/50090-chess-game-core-librairie-jeu-d-echec-en-c
- http://imagecomputing.net/damien.rohmer/data/previous_website/documents/teaching/13_0fall_cpe/3eti_software_development_c/documents_generaux/02_presentation_projet.pdf
- http://khayyam.developpez.com/articles/algo/genetic/