

WinEchek

Dossier préliminaire

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

1 Introduction

1.1 Besoins

WinEchek est une application Windows qui permettra de jouer aux échecs sur un ordinateur. Les parties pourront être dispensées de différentes manières. La première sera de jouer à deux joueurs humains sur le même ordinateur. La seconde permettra à deux joueurs humains de jouer en réseau (local) sur la même partie. Enfin, la dernière proposera à un joueur seul de dispenser une partie contre l'ordinateur qui sera doté d'une intelligence artificielle.

1.2 Objectif

Le but du projet n'est pas forcément d'implémenter le plus de fonctionnalités possibles, ni même de proposer quelque chose d'innovant. En effet, ce type d'application étant assez difficile à réaliser, les projets existants seront forcément plus intéressants que notre réalisation finale. Notre objectif est donc mettre en pratique, à travers la réalisation d'une application complexe, les différents domaines abordés dans notre formation (IHM, CO, POO, Réseau) mais également de découvrir les bases de la programmation d'intelligences artificielles. Pour résumer, nous réalisons cette application plus dans un but pédagogique que pour combler un besoin (les solutions existantes sont de très performantes, et la plupart proposent déjà nos fonctionnalités).

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 modéliser 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 recherche¹ 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

Pour ce qui est du langage utilisé pour l'implémentation, nous avons évalué trois options, le C++, le C# et Java.

Le principal atout du C++ est la performance, en effet celui-ci permet de descendre plus bas niveau que le C# ou le Java. D'autre part, la bibliothèque standard est moins haut niveau. Les outils fournis en comparaison au Java/C# ne permettent pas un développement rapide ceux-ci étant plus performant, mais moins simple à l'utilisation ce qui dans le cadre de notre projet est peu pertinent, nous ne cherchons pas à réaliser un programme performant. De plus, il nous aurait fallu utiliser un framework comme Qt pour réaliser la graphique, et l'environnement de développement associé (Qt creator) est quelque peu contraignant.

Le Java quant à lui, présente l'avantage d'être plus haut niveau que le C++, d'avoir plusieurs librairies graphiques intégrées. Suite à la réalisation de plusieurs projets en Java (POO, Graphe, Modélisation), nous sommes familiarisés avec sont utilisation, il nous aurait été simple de l'utiliser. L'environnement de développement que nous utilisons (IntelliJ Idea) pour le développement Java est complet et moderne et moins contraignant que QtCreator. Par ailleurs, le fait que Java tourne sur une machine virtuelle permet une portabilité du programme sous différentes plateformes jugé inutile dans le cadre de ce projet.

Notre choix se portera sur le C#. Celui-ci dispose d'un environnement de développement puissant et complet (Visual Studio). Le framework utilisé avec le C# (.NET) est un framework complet et haut niveau. Il permet par exemple une abstraction sur des points qu'il aurait été plus long de développer en C++ comme par exemple la mise en réseau du programme. La création de l'interface graphique est aussi plus simple qu'avec ses deux concurrents, la bibliothèque graphique (WindowsPresentationFoundation) fournie avec le framework simplifie par exemple la réalisation de fenêtre. Et bien que jugé peu pertinente dans le cadre de ce projet, pour ce qui est de la performance le C# se situe entre le C++ et le Java.

Le choix du C# par rapport au Java s'est donc fait sur la flexibilité du Framework graphique fourni par C#, mais c'est également l'opportunité d'apprendre plus en détail ce langage, très utilisé dans le monde professionnel.

En conclusion, le C# nous permet de simplifier un bon nombre de points qu'il est peu intéressant de développer dans le cadre de ce projet. Ce gain de temps nous permettra donc de travailler plus des points intéressants du projet, la qualité et la robustesse, et peu être d'arriver plus vite à la partie final du projet : la réalisation de l'intelligence artificielle.

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/>