

# Livrable 1

## Projet tuteuré

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

## 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 logiciel 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 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 recherche<sup>1</sup> appelé élagage Alpha-Bêta (cf. partie 5.1) et a subi une refonte totale pour sa version 5 en 1999, car son code

---

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.

était mal écrit.

## 2.2 Deep Blue

[https://fr.wikipedia.org/wiki/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/>

### 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
  - Seul
  - 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

## 5 Technologies

### 5.1 Intelligence Artificielle

- Algorithme MiniMax
- Elagage Alpha-Beta
- Algorithme Nega-Alpha
- Algorithme Alpha-Beta avec mémoire

#### 5.1.1 Evaluation des pièces

- Vide : 0
- Roi : 0
- Reine : 9
- Tour : 5
- Cavalier : 3
- Fou : 3
- Pion : 1

### 5.1.2 Bonus-Malus

- Pion passé : pion qui ne rencontrera plus de pion adverse et qui peut donc aller en promotion
- Tour en 7ème rangée
- Roi dans une colonne ouverte
- Pions du roque avancés
- Le roi a été déplacé et ne peut plus roquer
- Pions doublés (2 pions dans la même colonne)
- Pion isolé : pion qui ne peut plus être protégé par un autre pion

### 5.1.3 En jeu

- En début de partie, il est préférable de roquer pour protéger le roi. Au contraire, en fin de partie, il faut placer le roi au centre pour éviter un mat sur une arrête ou dans un coin de l'échiquier
- Un cavalier peut atteindre un plus grand nombre de cases s'il est au centre
- Les pions prennent de la valeur s'ils approchent la 8ème rangée (récupérer une dame)

### 5.1.4 Pénalités

- Pions doublés ou triplés : plusieurs pions sur une même colonne se gênent dans leurs mouvements
- Pions bloqués : deux pions se bloqués face à face
- Pions passés : un pion qui ne peut être que bloqué ou pris par l'adversaire constitue un grand danger car il menace la ligne arrière
- Pion isolé : un pion qui n'a aucun pion allié sur ses côtés
- Avoir les 8 pions sur une même ligne

## 5.2 Moteur

### 5.2.1 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://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/>