

# Rapport Projet Long

## Pour le 15 mars 2023

Ange Herman KOUE-HEMAZRO, Eric Nzaba

February 28, 2024

## 1 Introduction

**But du projet.** Implémenter une IA pour le jeu d'échecs basée sur la Recherche arborescente Monte-Carlo.

**Métrique.** On utilise l'api de lichess.org pour la partie graphique de notre projet et l'api donne aussi la possibilité de jouer contre une IA (niveaux entre 1 et 8) basée sur l'algorithme de stockfish qui est un des plus puissant qui n'utilise pas les réseaux de neurones. Notre projet sera un succès si notre IA arrive à battre au moins les 3 premiers niveaux de stockfish. Vu que l'algorithme de Monte Carlo se base sur une simulation de coup pris au hasard, ce sera dur de battre les niveaux supérieur de stockfish. Du coup le but sera de faire plusieurs simulations de parties contre stockfish et on sortira les résultats sous forme de pourcentage.

## 2 Implementation

**Comment on réalise notre projet.** Nous codons en python et comme dit plus haut, on utilise l'api de lichess.org pour ne pas avoir à coder la partie graphique du jeu.

**Logiciel déjà codé** . Nous avons déjà codé la représentation du jeu d'échec ainsi que les mouvements des pièces mais nous avons encore quelques bugs à régler du côté du roi. Nous avons déjà fait la connexion avec l'api et codé la partie de l'api qui va nous permettre de jouer des coups. L'algorithme de monte carlo a déjà été implémenté aussi.

**Structure en modules/packages.** Nous avons 3 packages sans compter le package des tests. Le package `ai` est celui qui contient les fichiers en rapport avec notre IA basé sur Monte Carlo. Le package `api` quand à lui contient tout

ce qui est en rapport avec l'api de lichess.org, les fonctions de connexions, de récupération et d'envoi de données etc. Enfin le package **chess** contient tout ce qui est en rapport avec notre représentation du jeu d'échecs, les fonctions pour avoir tous les coups possibles etc ...

**Représentation des données.** Nous avons représenté le plateau de jeu comme un tableau d'entier pour que ça prenne le moins de place que possible dans la mémoire vu que notre algorithme de recherche passera par des milliers de plateau de jeu. Nous comptons aussi introduire plus tard fichier pour stocker les mouvements de debut de game comme celui qui permet de liberer le fou ou la reine.

**Technologies sur lesquelles notre code s'appuie.** Notre IA se base sur l'algorithme de recherche arborescente Monte Carlo et nous utilisons aussi l'api de lichess.org pour la partie graphique.

### 3 Jalons

**Tâches à réaliser pour conclure le projet.**

1. Regler les bugs dans les mouvements des pièces
2. Permettre à un utilisateur lambda de jouer contre notre IA

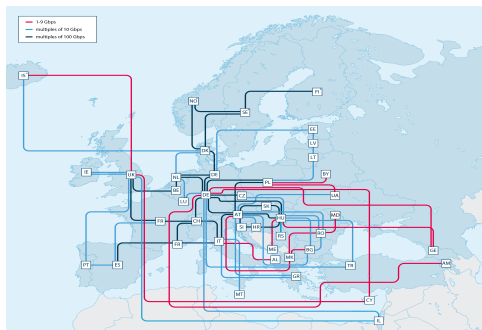


FIGURE 1 : A picture of the GEANT network.

---

**Organisation temporelle des tâches.**

**Tâches déjà terminées.**

1. Partie de l'api qui permet à notre IA de jouer contre celles de lichess

2. Implémentation de l'algorithme de recherche arborescente Monte Carlo
3. Implémentaion des mouvements d'échecs sauf celui du roi ( Encore quelques bugs)