Livrable II

Projet Informatique 31

ChessMate

Thomas Dahmen

Oscar Bouvier

Jean Forissier

Raphaël Macquet

Fonctionnement général

Ce livrable est constitué du programme 1, à savoir un jeu d'échecs que nous avons développé en Python, muni d'une interface graphique et d'une intelligence artificielle min-max. Le programme 1 contient trois modules :

1. **echecs.py**

   Il s'agit du moteur de jeu : il contient le plateau de jeu avec les pièces, les règles de déplacement, et c'est aussi dans le moteur que l'intelligence artificielle est implantée. Il communique étroitement avec l'interface graphique.

2. **gui.py**

   Il s'agit de l'interface graphique sans intelligence artificielle. C'est le script qu'il faut lancer si l'on veut joueur à deux sans intelligence artificielle, via la commande (UNIX) :

   ```
   $ python3 gui.py
   ```

   Les cas d'échec sont gérés par notre programme. Nous pourrions par la suite ajouter un message de succès lorsqu'il y a échec et mat, ainsi que la possibilité (essentielle pour tout bon joueur) de roquer.

3. **gui_IA.py**

   Il s'agit de l'interface graphique avec intelligence artificielle. C'est le script qu'il faut lancer si l'on veut joueur seul contre l'intelligence artificielle, via la commande (UNIX) :

   ```
   $ python3 gui_IA.py
   ```

   Malheureusement pour l'instant, au bout de quelques tours l'intelligence artificielle devient trop lente pour pouvoir finir une partie en un temps décent, et de plus elle ne nous donne pas encore satisfaction concernant son "intelligence" (certains mouvements qu'elle produit sont mauvais). Pour rendre l'intelligence artificielle plus forte, il faudrait revoir notamment sa fonction d'évaluation.

Modules utilisés

Python est un langage interprété, il est donc nécessaire de posséder Python 3 avec certains modules afin de lancer le programme :

1. Numpy
2. PyQt5
3. anytree
4. math

Tous ces modules, à l'exception de anytree (que nous fournissons dans le livrable) sont disponibles dans la distribution Anaconda. Notons que pour les tests unitaires nous avons utilisé le module Unittest, pour les exécuter il est donc aussi nécessaire d'avoir ce module.

Documentation

Le code des différents modules est commenté progressivement, avec en plus une docstring et une documentation HTML générée par l'outil Sphinx. Cette documentation des classes et fonctions utilisées se trouvent dans le dossier Documentation.

Tests unitaires

Nous avons réalisé des tests unitaires pour vérifier que notre moteur de jeu calcule correctement les positions accessibles pour chaque type de pièce. Le résultat (positif) de ces tests et le code utilisé se trouve dans le dossier Unit tests.