

Programmation d'un jeu d'échecs

proposé par Arnaud Sangnier (sangnier@liafa.univ-paris-diderot.fr)

Projet de Programmation 2013/2014 - 12 septembre 2013

1 Les échecs

Est-il besoin de présenter ce jeu millénaire qui oppose deux joueurs autour d'un plateau appelé *échiquier* de 64 cases blanches et noires ? Une description détaillée des règles de ce jeu est disponible sur la page *Wikipedia* correspondante : <http://fr.wikipedia.org/wiki/Echecs>.

2 Description du projet

Le but de ce projet est de programmer une interface graphique pour le jeu d'échecs. Ce projet comprendra plusieurs phases de développement et libre à vous de proposer éventuellement d'autres directions.

2.1 Une interface graphique pour jouer à deux joueurs humains

La première phase de ce projet consiste à développer une interface graphique pour jouer à deux joueurs humains. Le déplacement des pièces sur l'échiquier se fera par la souris. Votre interface devra respecter les règles suivantes :

- Faire jouer chacun des joueurs à tour de rôle ;
- N'autoriser que les déplacements légaux, comme par exemple :
 - Le joueur blanc ne peut pas déplacer une pièce du joueur noir ;
 - Le mouvement des différentes pièces doit être respecté (par exemple un fou ne peut pas être déplacé en ligne droite) ;
 - Un joueur ne peut pas faire un mouvement qui le mette en échec ;
- Détecter la fin d'une partie (par échec et mat, ou, pat).

Il n'est pas nécessaire dans un premier temps de programmer les règles spéciales (telles que les roques, la prise en passant ou la promotion d'une pièce lorsque un pion arrive sur la dernière rangée de l'échiquier).

2.2 Sauvegarder des parties et les rejouer

Une fois que vous aurez une interface graphique qui fonctionne, vous pourrez penser à comment sauvegarder une partie si par exemple les deux joueurs veulent suspendre leur partie pour la reprendre plus tard.

Dans un premier temps, vous pourrez penser à un format de fichier texte de votre choix, dans lequel vous stockerez l'état d'une partie (position des pièces sur l'échiquier et couleur du joueur qui doit jouer le premier coup). L'idée est la suivante : au moment où l'utilisateur décide à travers l'interface graphique de sauvegarder une partie, la partie est enregistrée dans un fichier texte au format choisi et lorsque l'utilisateur décide d'ouvrir une partie, il choisit le fichier texte correspondant et la partie associée recommence alors dans l'interface graphique.

Dans un deuxième temps, vous pourrez programmer une façon de sauvegarder tout le déroulement d'une partie, pour cela vous utiliserez la *notation algébrique* (cf http://fr.wikipedia.org/wiki/Notation_algebrique). Ainsi vous pourrez enregistrer toute une partie, mais également vous pourrez visualiser une partie décrite dans ce format. Vous devrez également, lorsqu'une telle partie sera ouverte, proposer une façon à l'utilisateur d'avancer coup par coup ou de retourner au début de la partie ou à la fin.

2.3 Un peu d'intelligence artificielle

Pour finir, une fois les parties précédentes programmées, vous pourrez étendre votre programme de façon à ce qu'un joueur puisse jouer contre l'ordinateur. Pour se faire, vous devrez développer des algorithmes de jeu.

Une façon simple consiste à ce que votre programme joue de façon complètement aléatoire en minimisant les pertes, c'est-à-dire qu'entre plusieurs coups possibles il choisira le coup qui lui fera subir le moins de perte possible au coup suivant (par exemple si il peut bouger un pion qui peut être pris au coup suivant et un autre pion qui n'est pas mis en danger au coup suivant, il choisira de bouger le second pion). Bien entendu ceci n'est qu'une première étape et un tel programme a de fortes chances de perdre rapidement même contre un joueur peu expérimenté.

Afin d'améliorer votre programme, vous pourrez choisir de consulter une liste de parties (qui seront stockées au format précédemment introduit) et si il existe une configuration similaire dans la liste des parties qui est gagnante pour le joueur correspondant à votre programme, alors votre programme jouera de la même façon que dans cette partie. Pour que ce genre de méthodes fonctionne correctement en pratique, il faudrait disposer d'une immense liste de parties et d'une façon rapide de parcourir ses parties, mais ce qu'on attend de vous ici c'est que vous soyez capables de programmer une telle façon de raisonner sans forcément que votre programme soit le meilleur du monde. Ainsi on jugera, si vous avez été capables de programmer la recherche dans différentes parties et le choix du coup à jouer. Si la configuration du jeu n'est pas disponible dans la liste des parties, alors votre programme pourra jouer de façon aléatoire en essayant de minimiser les pertes.

Bien entendu, dans cette partie, vous pouvez proposer vos propres idées pour améliorer l'intelligence artificielle de votre programme et cela sera même considéré comme un aspect extrêmement favorable. N'hésitez pas également à vous documenter pour avoir d'autres idées d'algorithmes.

2.4 Quelques conseils

Pour ce projet, il est important que vous procédiez pas à pas comme décrit ci-dessus. Ainsi, ne commencez pas à programmer la partie intelligence artificielle de votre programme avant d'avoir une interface graphique qui fonctionne correctement.

En ce qui concerne l'aspect sauvegarde des parties, il vous faudra faire attention à définir de façon extrêmement précise le format des fichiers où seront sauvegardés les parties. Ainsi vous devrez afficher un message d'erreur lorsque l'utilisateur essaiera d'ouvrir un fichier qui ne correspond pas à votre format. Il faudra de plus fournir une description détaillée de ce format.

2.5 Extensions possibles

Ce sujet vous donne des lignes directrices pour programmer ce projet, mais toute autre extension (à discuter avec votre encadrant) est la bienvenue. Par exemple, si vous souhaitez programmer une aide pour le coup suivant à jouer, ou si vous souhaitez incorporer une horloge de façon à ce que les deux joueurs puissent jouer en mode *blitz* (cf [http://fr.wikipedia.org/wiki/Blitz_\(echecs\)](http://fr.wikipedia.org/wiki/Blitz_(echecs))).