Projet du module “Bases de la Programmation à Objets 2”

Finales Simples aux Echecs

Mikal Ziane

Au jeu d'échecs, si la partie ne s'est pas terminée plus tôt, trois phases se succèdent : l'*ouverture* où les pièces sont développées, le *milieu de jeu* où les rois restent typiquement à l'abri et la ***finale*** où les rois deviennent actifs. Dans les finales les plus simples, il ne reste que très peu de pièces en plus des deux rois. Il s'agit alors pour le camp le plus fort de mater le roi ennemi. Les débutants apprennent notamment à jouer avec roi et tour contre roi ou roi et dame contre roi ou bien roi et dame contre roi et tour etc.

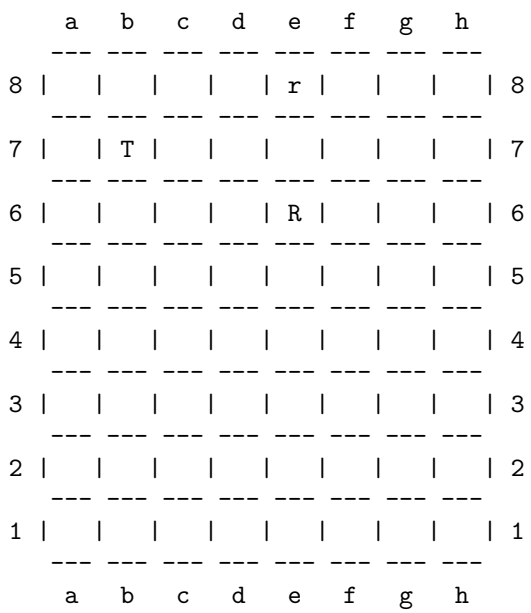
Dans ce projet, votre travail consiste à développer **et tester**, **en respectant une architecture logicielle** prédéfinie, un programme permettant de jouer une finale à partir d'une certaine position de départ. Le minimum requis est de prendre en compte le **roi** et la **tour**. Il n'est pas demandé d'inclure plus de types de pièces, mais il faudra que vous respectiez l'architecture prévue qui vise à **faciliter, dans le futur, l'ajout de nouveau type de pièces**.

Vous n'avez pas à connaitre toutes les règles du jeu d'échecs pour faire ce projet, mais seulement le déplacement des pièces que vous autorisez et les règles du **mat** et du **pat** ainsi que le déroulement d'une partie : chacun des 2 joueurs joue à son tour en déplacant (sauf exceptions qui ne concernent pas ce projet) une seule pièce d'une case à une autre de l'échiquier. Le site de la fédération française inclut une brochure simple sur les règles du jeu ainsi que plusieurs vidéos illustratives : http://www.echecs.asso.fr/Actu.aspx?Ref=11983.

1 Travail à faire

Votre programme doit permettre à deux joueurs de s'affronter dans une finale d'échecs. Les joueurs pourront être soit humains (et ils saisiront alors les coups qu'ils veulent jouer), soit un algorithme (les coups seront alors, par exemple, choisis aléatoirement). Toutes les combinaisons (2 joueurs humains, 1 joueurs humain face à un algorithme, ou encore 2 algorithmes) doivent être possibles.

L'interface avec les utilisateurs de votre programme doit être en mode texte. L'affichage du damier (ici dans sa situation initiale vue du point de vue du joueur blanc) doit respecter le format suivant :



Les pièces blanches sont en majuscules (R, D, T, F, C, P) et les noires en minuscules (r, d, t, f, c, p). Les coups des joueurs sont précisés uniquement par les coordonnées de la case de départ et celles de la case d'arrivée (`b7b8` est un coup légal dans le damier ci-dessus). Bien entendu, les coups illégaux doivent être détectés, signalés et rejetés. Notez que si, comme c'est vivement conseillé, vous ne prenez pas en compte les pions, de nombreuses règles spéciales n'ont pas à être codées (prise différente des déplacements, promotion, prise en passant). Vous n'avez pas à coder non plus la règles des 50 coups, ni la règle de la nulle par répétition de la même position, mais vous devrez prendre en compte l'abandon ou la proposition de nulle.

Qui, quoi et quand?

Votre projet doit être fait par une **équipe de 3** étudiants qui peuvent provenir de groupes différents.

Vous devez soigner la conception de votre programme (quelle hiérarchie de classes et d'interfaces, quelles parties publiques, quels paquetages, quelles dépendances entre tous ces éléments). **Cela sera le critère principal sur lequel votre projet sera noté**.

Des consignes et conseils supplémentaires seront donnés en cours soyez-y attentifs !

Vous devez porter une attention particulière à la rédaction de votre dossier. Sa qualité est déterminante pour l'évaluation de votre travail. La composition de votre dossier doit être la suivante.

- Une page de garde indiquant le nom et **le groupe** des membres de l'équipe et le titre :"projet BPO2 échecs)".
- Une table des matières de l'ensemble du dossier.
- Une brève introduction qui indique ce qui marche et ne marche pas.
- Un diagramme d'architecture comme l'exemple donné ci-dessous. **Ne montrez pas le méthodes ni les attribut**s. Pensez aux dépendances.
- Le listing des **tests unitaires** de vos classes (en précisant ceux qui échouent le cas échéant).
- Le listing complet de vos sources documentés (javadoc).
- Une explication indiquant les tâches à accomplir sur votre programme pour y intégrer d'autres règles du jeu ou des joueurs simulés présentant plus d'intelligence que de jouer uniquement aléatoirement.
- Un bilan du projet (les difficultés rencontrées, ce qui est réussi, ce qui peut être amélioré).

Vous devez rendre votre rapport complet au plus tard le 20 mai sur Moodle.

Exemple de diagramme d'architecture