

Sujet du projet – Deuxième partie

1 Travail demandé

Pour cette deuxième partie, il vous est demandé de compléter et d'étendre votre simulateur graphique du jeu d'échecs développé dans la première partie.

Pour le rendre complet quant aux règles du jeu d'échecs¹, il s'agira d'ajouter la prise en compte :

- des règles spéciales de déplacement et de prise des pièces du roque, de la prise en passant et de la promotion ;
- des conditions de fin de partie particulières :
 - perte au temps : il faudra pour cela mettre en place le jeu à la pendule², en permettant, à travers l'interface graphique, la programmation de la cadence utilisée en sélectionnant le nombre des périodes, leurs durées et nombres de coups respectifs, si elles fonctionnent avec incrément ou non, etc.
 - différents cas menant à une partie nulle³ :
 - pat ;
 - triple répétition de la position ;
 - règle des 50 coups ;
 - impossibilité de mater.

Ensuite, il vous faudra développer un système de sauvegarde et de chargement de parties, qui devra permettre :

- de sauvegarder une partie en cours ou qui vient de s'achever (c'est-à-dire sauvegarder tout l'historique, tout le déroulement d'une partie) ;
- de charger une partie sauvegardée ou importée afin de pouvoir la rejouer ou la reprendre à un certain point.

Ce système devra au minimum prendre en compte le format standardisé PGN (Portable Game Notation⁴).

Enfin, il vous est demandé de mettre en œuvre au moins une variante du jeu d'échecs parmi celles présentées dans l'article Wikipédia suivant (en excluant les variantes uniquement basées sur une modification de la position de départ) : https://fr.wikipedia.org/wiki/Variante_du_jeu_d%27%C3%A9checs. Nous vous demandons que vous nous soumettiez d'abord une proposition de variante que vous aimeriez mettre en œuvre en expliquant comment vous pourriez le faire en tirant profit de la programmation orientée objet.

2 Critères d'évaluation

Globalement, les mêmes critères que pour la première partie s'appliquent pour cette partie.

2.1 Rapport et soutenance

Vous devrez rendre un rapport de 2 à 3 pages (en format PDF, généré par L^AT_EX) expliquant les choix d'implémentation que vous aurez rencontrés. Dans le cas où votre programme présenterait des défauts, il faudra les mentionner dans le rapport en précisant leurs raisons. Une soutenance d'une dizaine de minutes par groupe sera organisée à la fin de la deuxième partie du projet durant laquelle vous nous ferez une démonstration de votre programme.

2.2 Fonctionnalités du code

Bien évidemment, votre projet sera évalué par ses fonctionnalités. S'il remplit tout ce qui est demandé, rajouter d'autres fonctionnalités pourra apporter un bonus. L'ergonomie de l'interface graphique pour ce qui concerne son intégration des nouvelles fonctionnalités sera également prise en compte dans l'évaluation de votre projet : il n'est pas nécessaire que celle-ci soit très évoluée ou ait une apparence particulièrement travaillée, mais elle devra être suffisamment fonctionnelle et intuitive à l'utilisation.

1. <https://fr.wikipedia.org/wiki/%C3%89checs>
2. https://fr.wikipedia.org/wiki/%C3%89checs#Jeu_.C3.A0_la_pendule
3. <https://fr.wikipedia.org/wiki/Nulle>
4. https://fr.wikipedia.org/wiki/Portable_Game_Notation

2.3 Organisation du code

Votre projet devra impérativement être organisé hiérarchiquement, en le séparant en fichiers, classes et méthodes. Vous tâcherez de séparer du mieux possible les différentes fonctionnalités ajoutées au cours de cette partie en classes. Gardez sous le coude la règle de ne jamais avoir de fonction trop longue ou de fichier trop grand.

2.4 Qualité du code

L'utilisation adéquate de la programmation orientée objet et de Scala sera un critère important dans l'évaluation, notamment pour compléter et étendre votre programme en factorisant et en réutilisant au mieux votre code. Dupliquer du code dans des classes sous-entendrait une mauvaise compréhension de l'héritage. De même, préférez des directives fonctionnelles concises à des boucles `for` et `if` imbriquées, voir la Section 4.4 de l'introduction à Scala. La mise en forme, la présence de commentaires et la cohérence des noms de classes, méthodes et variables devront être suffisamment décentes pour une lecture agréable du code. Nous vous demandons également de documenter votre code, comme expliqué dans la Section 7 de l'introduction à Scala.

3 Dates importantes

- Le code et le rapport seront à rendre avant le mardi 28 mars à 23h59.
- La soutenance pour la deuxième partie aura lieu le vendredi 31 mars.