Labo 08– Echecs

Auteurs : P.Gillet & L.Haye

Groupe n° L08GrC

Table des matières

[1. UML 1](#_Toc155688308)

[2. Choix de conception 1](#_Toc155688309)

[3. Tests 1](#_Toc155688310)

# UML

# Choix de conception

* Chaque pièce du jeu découle de la classe abstraite Piece.
* Afin de factoriser un peu notre code, nous avons représenter les coordonnées x, y d’une case avec la classe Point, disponible dans java.awt.
* Avant d’effectuer un mouvement, une pièce doit vérifier si ce déplacement est autorisé. C’est ensuite que le mouvement est effectué sur le plateau de jeu.
* Afin de factoriser notre code, nous avons fait en sorte que les pièces ayant le même type de mouvement (linéaire, diagonale et/ou restreint par une distance) utilise la même interface.
* Pour vérifier que le mouvement d’un joueur ne le mette pas en échec lui-même, nous effectuons ce mouvement avec une pièce factice puis vérifions si le roi est sain et sauf. En cas d’échec et mat, il ne sera donc plus possible d’effectuer de mouvement.

# Tests

**Tests de la classe Board**

|  |  |
| --- | --- |
| **Description du test** | **Résultat attendu et observé** |
| Constructeur de l’échiquier | OK |
| Réinitialisation de l’échiquier | OK |
| Vérification que la partie commence bien par les pièces de couleur blanche, puis alterne après chaque tour avec celles de couleur noire | OK |
| Getter pour une case du plateau | OK |
| Déplacement d’une pièce tant que celui-ci est légal | OK |
| Placement d’une pièce sur l’échiquier | OK |
| Vérification qu’une pièce ne puisse pas réaliser un mouvement si celui-ci met son roi en échec | OK |
| Suppression d’une pièce | OK |
| Roque | OK |
| Prise en passant | OK |
| Retour de la dernière pièce jouée | OK |
| Définition du nombre de tours de jeu | OK |
| Placement des pièces en début de partie | OK |
| Promotion de pion | OK |
| Déplacement impossible pour des cases vides | OK |
| Déplacement impossible en cas de collision | OK |

**Tests similaires à chaque type de pièces**

|  |  |
| --- | --- |
| **Description du test** | **Résultat attendu et observé** |
| Getter du type d’une pièce | OK |
| Getter du nom d’une pièce | OK |
| Méthode toString | OK |
| Déplacement impossible si celui-ci met son roi en échec | OK |

**Tests de la Reine**

|  |  |
| --- | --- |
| **Description du test** | **Résultat attendu et observé** |
| Mouvements rectilignes et diagonaux, sans restriction de distance et tout en respectant les collisions sur son chemin | OK |

**Tests du Roi**

|  |  |
| --- | --- |
| **Description du test** | **Résultat attendu et observé** |
| Mouvements rectilignes et diagonaux, de 1 seule case et tout en respectant les collisions sur son chemin | OK |
| Petit et grand roque à gauche et à droite | OK |

**Tests du Pion**

|  |  |
| --- | --- |
| **Description du test** | **Résultat attendu et observé** |
| Mouvements rectilignes en face de celui-ci, d’1 case ou 2 s’il est à sa position de départ, tout en respectant les collisions sur son chemin | OK |
| Ne peut prendre de pièce uniquement si la pièce adverse est en diagonal de ce dernier et dans la bonne direction | OK |
| Capture en passant | OK |
| Promotion | OK |

**Tests du Fou**

|  |  |
| --- | --- |
| **Description du test** | **Résultat attendu et observé** |
| Mouvements diagonaux, sans restriction de distance et tout en respectant les collisions sur son chemin | OK |

**Tests du Cavalier**

|  |  |
| --- | --- |
| **Description du test** | **Résultat attendu et observé** |
| Mouvements en « L », avec restriction de distance et ignorant les collisions sur son chemin | OK |

**Tests de la Tour**

|  |  |
| --- | --- |
| **Description du test** | **Résultat attendu et observé** |
| Mouvements rectilignes, sans restriction de distance et tout en respectant les collisions sur son chemin | OK |