Forestier Quentin, Herzig Melvyn

POO1 – 05.01.2021

Échecs

Laboratoire 8



Table des matières

[Introduction 2](#_Toc60337996)

[Notes sur l’implémentation 2](#_Toc60337997)

[Déroulement d’un tour 2](#_Toc60337998)

[Règles de déplacement 2](#_Toc60337999)

[Validité d’un mouvement 2](#_Toc60338000)

[Tests 3](#_Toc60338001)

[Tests généraux 3](#_Toc60338002)

[Tests reine 4](#_Toc60338003)

[Tests tour 5](#_Toc60338004)

[Tests fou 6](#_Toc60338005)

[Tests cavalier 7](#_Toc60338006)

[Tests pion 8](#_Toc60338007)

[Tests roi 9](#_Toc60338008)

[Conclusion 11](#_Toc60338009)

[Annexes 11](#_Toc60338010)

# Introduction

Durant ce laboratoire, il nous a été demandé de réaliser un jeu d’échecs. Une interface graphique était fournie, il ne restait qu’à implémenter le « moteur » de jeu.

# Notes sur l’implémentation

## Déroulement d’un tour

Un tour de jeu se déroule de la manière suivante :

* Le joueur choisit une pièce à déplacer
  + Si la position est vide ou contient une pièce ennemie, le tour recommence.
* Pour le pion choisi, on lui demande s’il peut se déplacer sur la destination selon ses règles de déplacements
  + Si la pièce ne retourne aucun mouvement ou si le mouvement ne respecte pas ses prérequis, le tour recommence.
* Le mouvement est appliqué
  + Si son propre roi est en échec, on inverse le mouvement et le tour recommence.
* Le joueur change
* On vérifie si le nouveau joueur est en échec, échec et mat ou pat.

## Règles de déplacement

Les pièces sont définies par un ensemble de règles et d’attributs qui leur sont liés.

De cette façon, il est très facile d’ajouter des règles de déplacement aux pièces ou de leur en retirer.

Les règles de déplacement vérifient uniquement si la pièce associée peut atteindre une case sans tenir compte de la mise en échec de son propre roi.

Par exemple, pour la prise en passant, il est seulement vérifié que le pion adjacent soit de couleur opposée, qu’il ait sauté de deux cases au tour précédent.

De même pour le roque il est seulement vérifié que le roi et la tour concernée n’aient pas été déplacée.

## Validité d’un mouvement

Un mouvement est valide s’il respecte ses prérequis et s’il ne met pas son propre roi en échec.

Par prérequis, il est entendu des mises en échec à priori du déplacement. Le seul qui nécessite une vérification est le roque. Il est vérifié que les deux ou trois cases de déplacement du roi ne le mettent pas en échec.

Après application du mouvement, toutes les pièces adverses sont interrogées quant à leur capacité à se déplacer sur le roi allié. Si une d’elles y arrive, le mouvement est invalidé. Si l’échiquier ne contient pas un roi pour chaque joueur, le jeu ne peut pas fonctionner et lèvera une RuntimeException.

# Tests

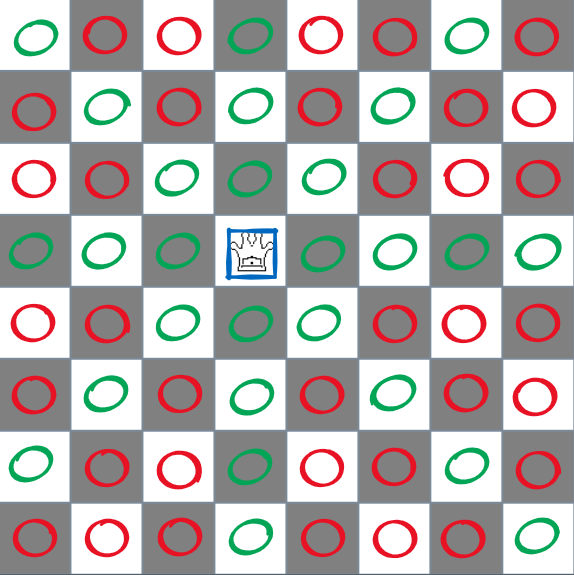
Tous les tests suivants ont été effectués manuellement. Si le résultat attendu dans la colonne « résultat attendu » est vert, cela signifie que le résultat observé correspond, dans le cas contraire la couleur sera rouge et les différences reportées à la suite du tableau. Sauf indication contraire, les mouvements effectués ne laissent pas le roi allié en échec et les tests sont effectués pour les deux couleurs.

## Tests généraux

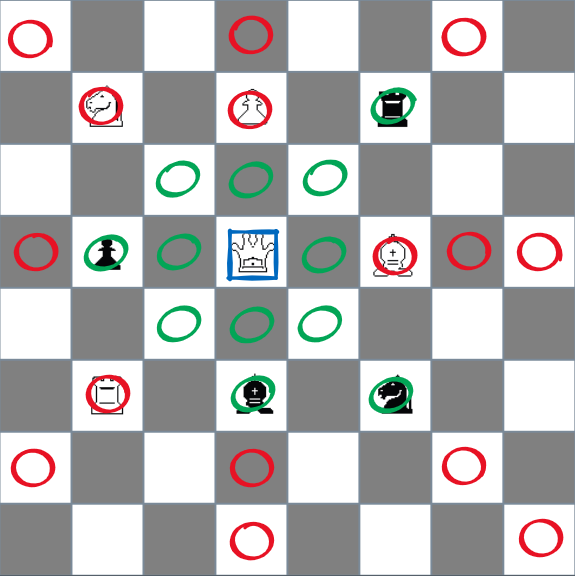
|  |  |
| --- | --- |
| Test effectué | Résultat attendu |
| Le bouton « New game » fonctionne. | Les pièces s’affichent à leur position initiale. |
| Les blancs commencent la partie. | Seule une pièce blanche peut être déplacée. |
| Les tours alternent entre blanc et noir. | Il est impossible de jouer deux fois d’affiler. Les joueurs jouent à la suite. |
| Effectuer un mouvement qui laisse son roi en échec. | Le mouvement n’est pas effectué et le joueur rejoue |
| Le roi adverse est en échec. | Le message « Check ! » s’affiche. |
| Le roi allié est en échec, un mouvement qui ne le sauve pas est effectué. | Le mouvement n’a pas lieu et le joueur rejoue. |
| Le roi allié est en échec, un mouvement qui le sauve est effectué. | Le mouvement est validé et le tour change. |
| Le message « Check ! » est présent et un mouvement invalide est effectué. | Le message persiste. |
| Le message « Check ! » est présent et un mouvement valide est effectué. | Le message disparait. |
| Un joueur est échec et ne peut pas sauver son roi. | Le message « Checkmate ! » apparait et la partie et finie. |
| Un joueur ne peut plus jouer sans se mettre en échec. | Le message « Pat ! » apparait et la partie et finie. |
| Si un pion atteint sa ligne de promotion, son type de promotion est demandé  (Reine, Tour, Fou, Cavalier) | Le type de promotion est demandé. |
| Pour un pion apte à la promotion, elle est effectuée conformément au type demandé. | La promotion est effectuée. |
| Il est impossible de ne pas transformer un pion apte à la promotion. | La demande de promotion persiste. |
| Un pion, après promotion, met le roi adverse en échec conformément à ses nouvelles règles. | Le message « Check ! » s’affiche. |

Nous avons ensuite testé les différents mouvements des pièces. Chaque rond est un déplacement testé et la pièce dans le carré est la pièce déplacée. Les ronds verts et rouges sont des tests réussis, si un comportement inattendu se produit, le rond sera violet. Un déplacement vert indique qu’il s’est déroulé et rouge qu’il n’a pas eu lieu. Les illustrations sont des synthèses de plusieurs situations. Une fois encore, les deux couleurs ont été testées mais une seule est représentée. Parfois nous n’avons pas illustré tous les clics et leur résultat car ils sont déjà démontrés par l’image précédente.

## Tests reine

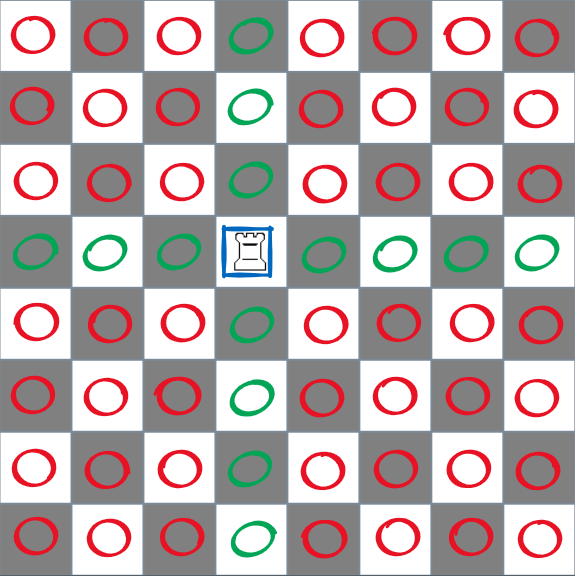


Les déplacements standards de la reine fonctionnent. Elle est capable de se déplacer uniquement verticalement, horizontalement et diagonalement.

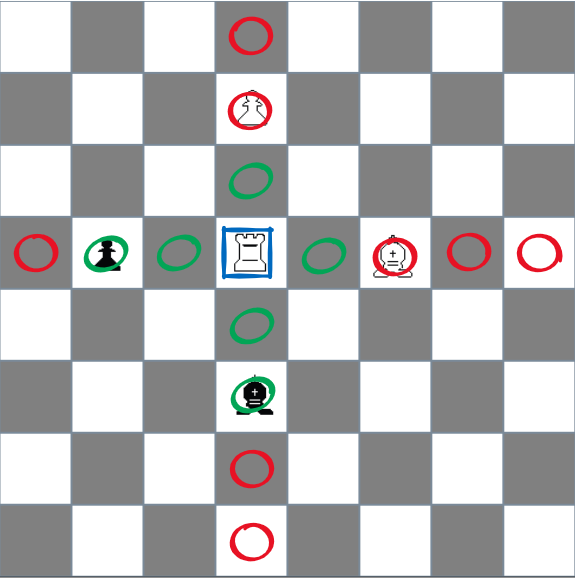


La reine ne peut pas sauter par-dessus d’autres pièces. Elle est capable d’attaquer les pièces adverses mais pas ses alliés.

## Tests tour

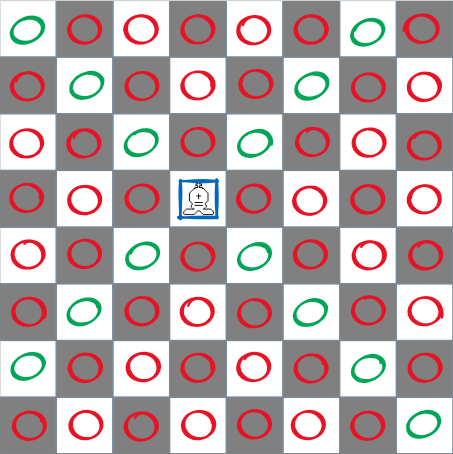


Les déplacements standards de la tour fonctionnent. Elle est capable de se déplacer uniquement verticalement et horizontalement.

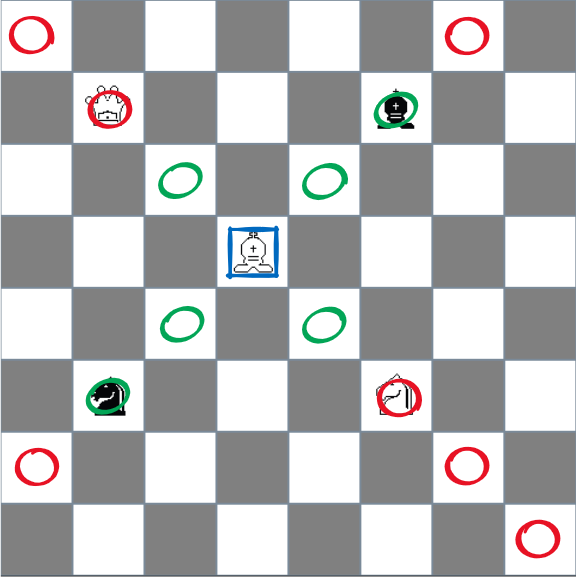


La tour ne peut pas sauter par-dessus d’autres pièces. Elle est capable d’attaquer les pièces adverses mais pas ses alliés.

## Tests fou

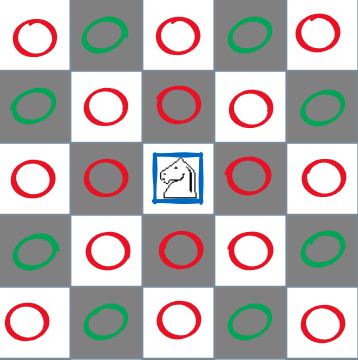


Les déplacements standards du fou fonctionnent. Il est capable de se déplacer uniquement diagonalement.



Le fou ne peut pas sauter par-dessus d’autres pièces. Il est capable d’attaquer les pièces adverses mais pas ses alliés.

## Tests cavalier



Les déplacements standards du cavalier fonctionnent. Il est capable de se déplacer uniquement en « L ».

|  |  |
| --- | --- |
|  |  |

Le cavalier est capable de sauter par-dessus les autres pièces, qu’elles soient alliées ou ennemies. Il peut attaquer les pièces adverses mais pas alliées.

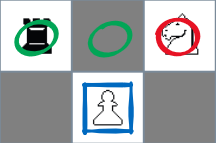
## Tests pion

|  |  |
| --- | --- |
|  |  |

Le pion est capable d’avancer frontalement d’une ou deux cases s’il ne s’est pas encore déplacé. Toutefois, il ne peut avancer que d’une case s’il a déjà bougé.

|  |  |
| --- | --- |
|  |  |

Le pion ne peut pas avancer et ce quelle que soit la pièce devant lui.



Le pion peut attaquer en diagonale ses ennemis mais pas ses alliés.

|  |  |
| --- | --- |
|  |  |

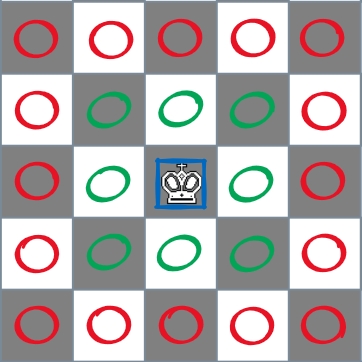
Le pion noir de gauche est le dernier à avoir bougé et a avancé de deux cases quant au pion droite, il a bougé de deux cases mais il n’est pas le dernier à s’être déplacé.

Dans le premier cas, le pion blanc peut prendre en passant l’adversaire de gauche car il s’est déplacé de deux et il est la dernière pièce bougé. Toutefois il peut quand même effectuer une attaque diagonale droite car la reine s’y trouve, mais le pion droite n’est pas attaqué.

Dans le second cas, le pion blanc peut attaquer pour les même raison le pion gauche. Au contraire, il ne peut plus attaquer en diagonale droite car aucune pièce ne s’y trouve. La prise en passant n’est pas possible car le pion noir droite n’est pas le dernier à s’être déplacé.

La prise en passant ne fonctionne que si le dernier mouvement du pion attaqué a été un déplacement de 2 cases.

## Tests roi

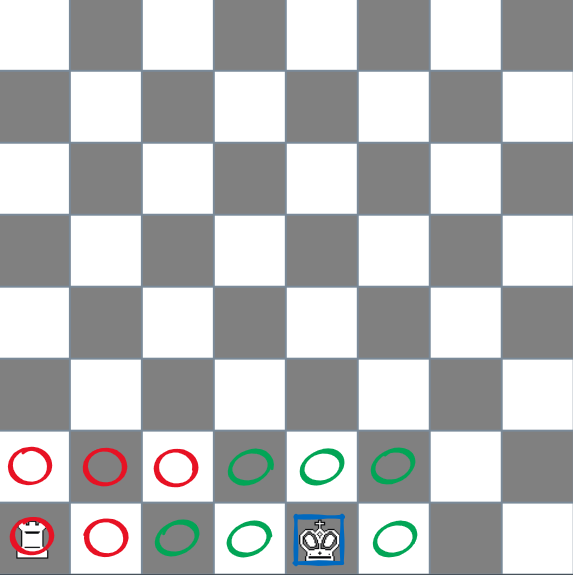


Les déplacements standards du roi fonctionnent. Il est capable de se déplacer uniquement d’une case verticalement, horizontalement et diagonalement.



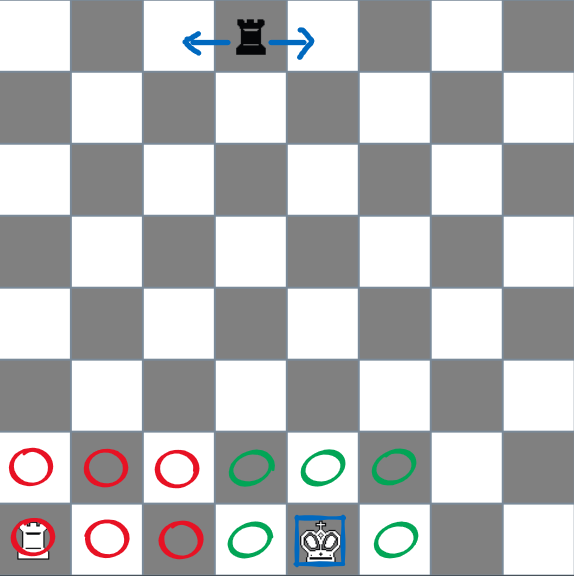
Il peut attaquer ses ennemis mais pas ses alliés.

Les situations suivantes ont été testées pour le grand et le petit roque.

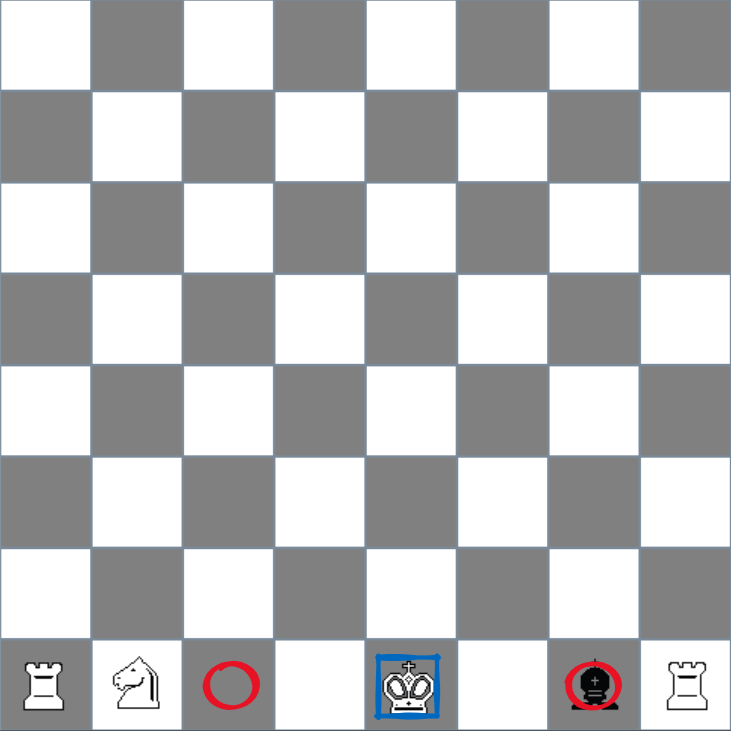


Le roi est capable de faire un roque si aucune pièce ne le sépare de la tour concernées et que les déplacement ne le mettent pas en échec.

Nous avons également testé le roque si la tour ou le roi s’est déjà déplacé. Le mouvement ne peut pas être réalisé comme attendu.



Le roi ne peut pas roquer s’il se trouve en échec ou si une des cases du mouvement le mettra en échec.



Le roque ne peut pas être effectué si une ou plusieurs pièces se trouvent entre le roi et la tour.

# Conclusion

Au terme de ce laboratoire, nous avons pu implémenter toutes les fonctionnalités demandées et même les fonctionnalités bonus. Nous avons été autant exhaustif que possible dans nos tests et aucun bug n’a été détecté. Le jeu est entièrement jouable.

# Annexes

* Les fichiers sources « .java ».
* Le diagramme de classes partiel qui contient que les éléments ajoutés par nos soins.