
Rapport de laboratoire

HEIG-VD
TIC

Laboratoire POO

Labo 8

Réalisé par :

Ivan Vecerina
Thibault Seem
Alice Grunder

A l'attention de :

M. Donini
M. Decorvet

Dates :

Début du laboratoire : 21 octobre 2021
Fin du laboratoire : 4 novembre 2021

Table des matières

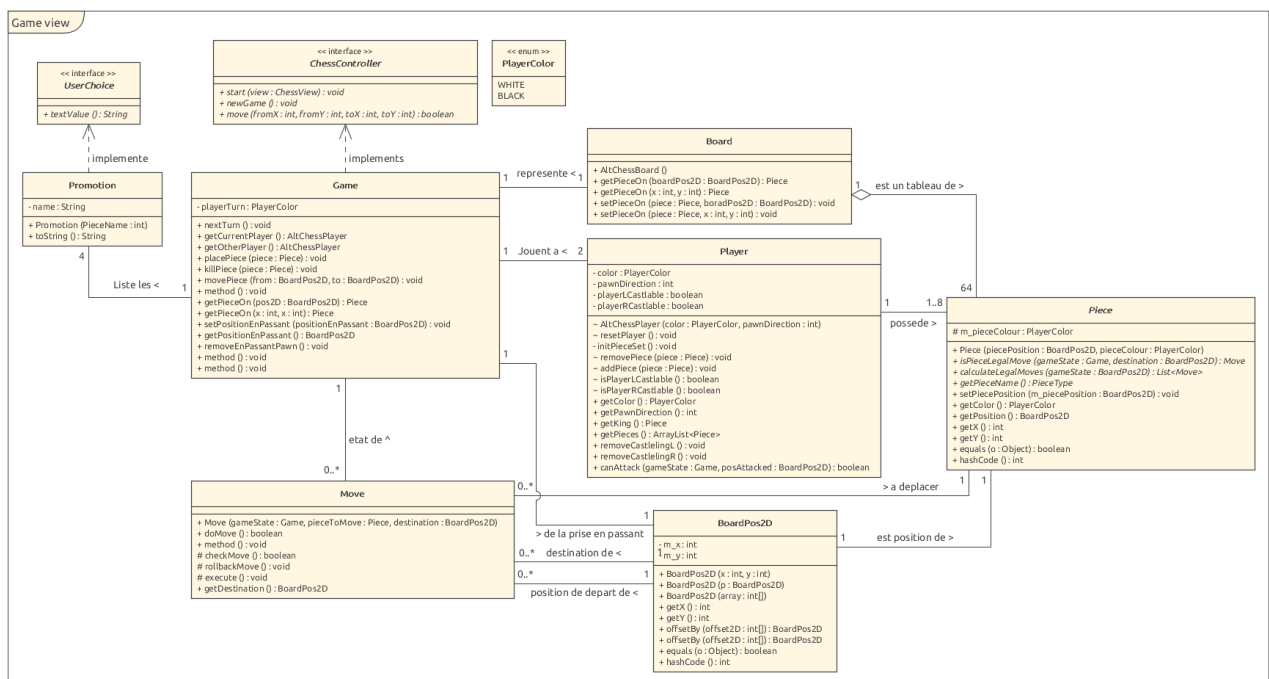
1 Introduction	3
2 Diagramme de classe	3
2.1 Base.....	3
2.2 Pieces	3
2.3 Moves	4
2.4 Diagramme de la donnée	4
3 Choix de modélisation	4
3.1 Description des classes.....	4
3.1.1 Game	4
3.1.2 Board	4
3.1.3 Player.....	5
3.1.4 Promotion.....	5
3.1.5 BoardPos2D.....	5
3.1.6 Piece	5
3.1.7 Move et ses filles	6
4 Tests.....	6

1 Introduction

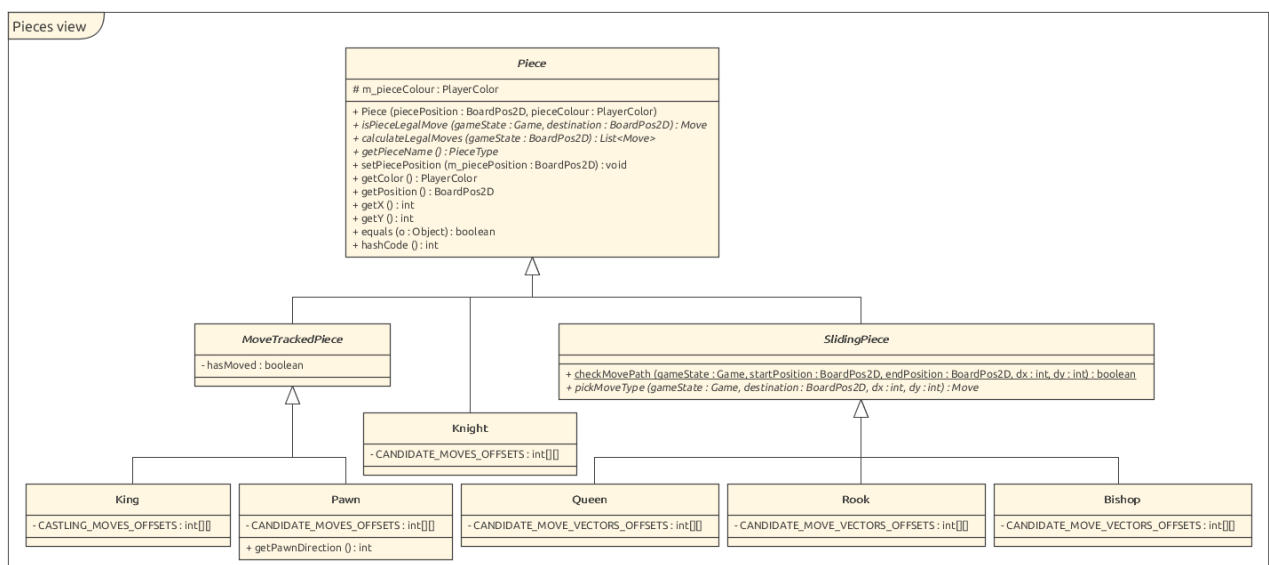
Dans ce laboratoire, nous devons développer la logique d'un jeu d'échec. L'interface nous est déjà fournie, elle consiste d'un échiquier et des ses pièces. Elle permet sélectionner deux cases avant d'envoyer cette information au contrôleur ; la première case choisie étant celle avec la pièce à déplacer et la seconde case la cible du déplacement. Nous devons implémenter les déplacements, la gestion des tours de joueur, la prise de pièces adverse, implémenter la promotion, la prise en passant et les roques ainsi que la détection de mise en échec.

2 Diagramme de classe

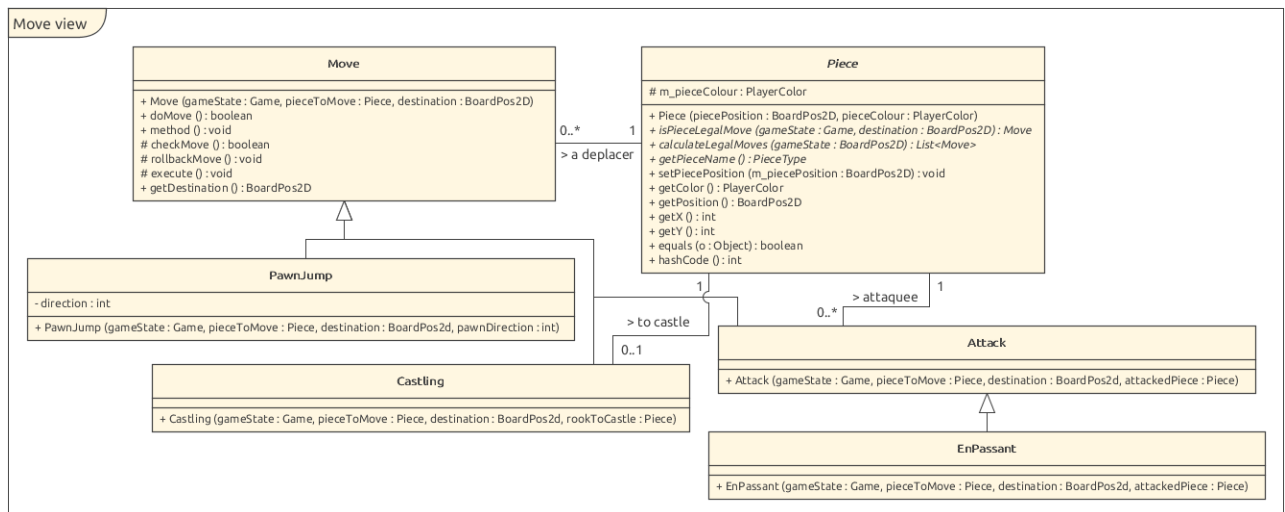
2.1 Base



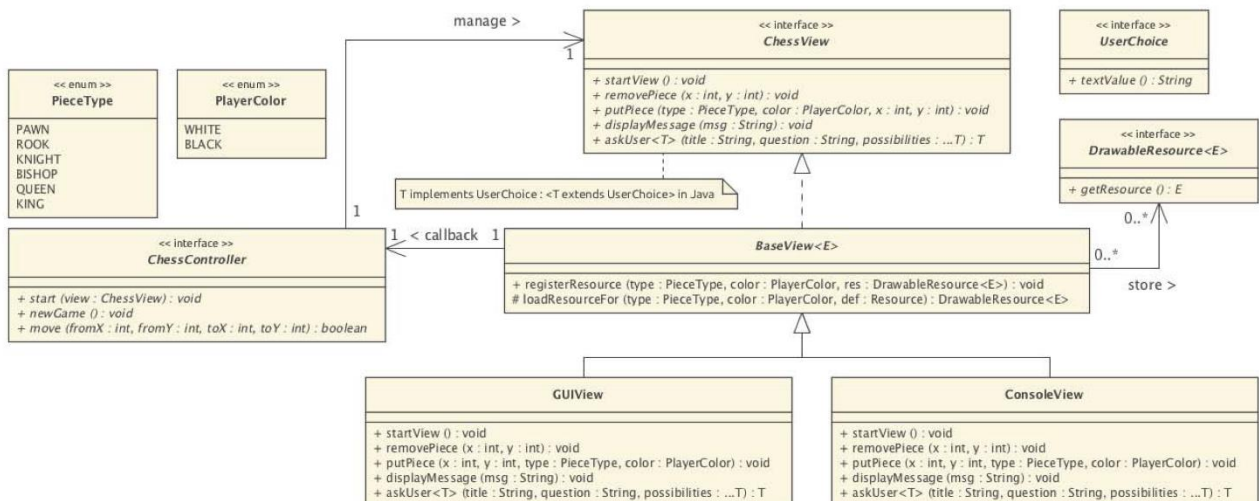
2.2 Pieces



2.3 Moves



2.4 Diagramme de la donnée



3 Choix de modélisation

3.1 Description des classes

3.1.1 Game

La classe *Game* implémente l'interface *ChessController* fournie pour le laboratoire. Elle représente l'univers du jeu. C'est cette classe qui communique avec l'interface. Elle gère le démarrage du jeu, le déplacement effectif des pièces, permet la communication entre les différentes parties du jeu et qui indique si un mouvement est illégal (Mouvement sur soi-même, sur un allié, en dehors du terrain, le reste est géré par une méthode de *Piece* qui est appelée ensuite et La classe *Move*). Elle gère la promotion de pion quand ils arrivent sur la ligne finale en face. C'est aussi ici qu'est enregistré la position permettant une prise en passant.

3.1.2 Board

Cette classe est simplement un tableau 8x8 de *Piece*. Elle met en place le plateau de jeu et permet d'interagir avec ses méthodes. C'est grâce à elle que l'on peut trouver si une case à une pièces et quelle est cette pièce.

3.1.3 Player

Cette classe permet de gérer les joueurs, elle liste les pièces de chaque joueur, indique si le grand ou petit roque sont faisables et permet d'initialiser le placement des pièces au début de partie. Elle implémente aussi une méthode permettant de savoir si un joueur peut attaquer une certaine case (permet la détection de mise en échec).

3.1.4 Promotion

Cette classe permet simplement de créer les différentes instances de promotion possible, Tout cela est mis en place dans la classe

3.1.5 BoardPos2D

Cette classe permet d'obtenir une position sur l'échiquier. Toutes les pièces ont leur position enregistrée de cette manière. La classe met à dispositions des méthodes pour obtenir une position en offset d'une autre, si une position donnée se trouve dans les limites du plateau et override les méthodes *equals* et *hashCode*.

3.1.6 Piece

Cette classe abstraite est centrale au jeu, elle permet de représenter les pièces du jeu. Une pièce à une position et une couleur, elle a une méthode qui lui permette d'indiquer si un mouvement donné est légal (vérifie qu'il n'y ait pas de pièces sur le chemin si nécessaire et que c'est bien un mouvement que la pièce peut effectuer : par exemple le fou ne peut se déplacer qu'en diagonale), cette méthode renvoi le type de mouvement (*Move*) qui sera effectué : un simple déplacement, une prise ou d'autre encore. Aussi, elle override les méthodes *equals* et *hashCode*

3.1.6.1 MoveTrackedPiece

Cette classe abstraite permet de mettre en place les pièces dont il est important de savoir si elles ont déjà effectué : Le roi et les pions.

3.1.6.1.1 King

Cette pièce est le roi, il peut seulement se déplacer dans les 8 cases adjacentes ou effectuer un roque : s'il n'y a pas de pièces entre un roi et une des ses tours, que les deux n'ont jamais bougé, que le roi n'est pas en échec ou les deux cases en direction de la tour ne sont pas menacées par l'adversaire alors le roi se déplace de deux cases en direction de la tour et cette dernière se déplace de l'autre côté.

3.1.6.1.2 Pawn

Cette pièce est le pion, il ne peut qu'avancer d'une case, ne peut pas prendre en vertical mais seulement en diagonale d'une case en avant. Si c'est son premier mouvement il peut faire un "saut" et avancer de 2 cases (si elles sont toutes les 2 vides) plutôt. Si un tel mouvement est effectué un pion adverse peut effectuer ce qu'on appelle une prise en passant : il prend comme si le pion précédant n'avait avancé que d'une case.

3.1.6.2 SlidingPiece

Cette classe abstraite permet de mettre en place les pièces qui se déplacent et prennent sans limites de distance dans les directions "normales" (verticalement, horizontalement, ou en diagonales) et qui "glissent" : elles ne peuvent pas "sauter" des pièces dans leur déplacement. Cette classe met en place des méthodes qui permettent de vérifier qu'un mouvement demandé respecte bien cette règle de glissement et de donner le mouvement effectué.

3.1.6.2.1 Queen

Cette pièce est la dame, elle peut se déplacer dans toutes les directions : verticalement, horizontalement, ou en diagonales

3.1.6.2.2 Rook

Cette pièce est la tour, elle peut se déplacer verticalement et horizontalement.

3.1.6.2.3 Bishop

Cette pièce est le fou, elle peut se déplacer diagonalement.

3.1.6.3 Knight

Cette pièce est le cavalier, il est particulier car il ne peut se déplacer et prendre que en sautant d'une case à une autre en "L" deux cases dans une direction verticale ou horizontale et une case dans l'autre. Le cavalier n'est pas embêté par les pièces sur son chemin.

3.1.7 Move et ses filles

La classe *Move* (et ses filles : *Attack* et sa fille *EnPassant*, *Castling*, et *PawnJump*) est la classe qui essaye de déplacer une pièce. Si cela devrait mettre le joueur actuel en échec ce mouvement est annulé. Les méthodes *doMove*, *tryMove*, *checkMove*, et *rollbackMove* font cela, dans *doMove* on essaye (*tryMove*) de se déplacer, puis on vérifie qu'on ne se met pas en échec (*checkMove*) et s'il faut on revient en arrière (*rollbackMove*)

3.1.7.1 Attack

Cette fille de *Move* met en place la prise de pièces adverse.

3.1.7.2 EnPassant

Cette prise particulière permet de faire la prise en passant à l'aide de la position enregistrée dans *Game*.

3.1.7.3 Castling

Ce mouvement particulier est le roque tel qu'expliqué ci-dessus dans *King*.

3.1.7.4 PawnJump

Ce mouvement particulier du pion lui permet de faire un déplacement de 2 cases plutôt que d'une si c'est son premier mouvement. Quelconque

4 Tests

Action effectuée	Résultat attendu	Résultat obtenu juste ?
Quelconque		
Se déplacer de manière à mettre son roi en échec	Le mouvement n'est pas effectué	OUI
Cliquer sur une case vide puis une autre quelconque	Il ne se passe rien	OUI
Cliquer sur une pièce adverse puis une autre case	Il ne se passe rien	OUI
Tenter un mouvement impossible (par exemple; un mouvement de cavalier avec autre chose qu'un cavalier, ou de se déplacer de plus de une case avec un roi ou un pion qui à déjà bougé)	Le mouvement n'est pas effectué	OUI
Pion		
Avancer un pion de 1	Le pion avance de 1 case	OUI
Avancer un pion de 2 à son premier coup	Le pion avance de 2 cases	OUI
Avancer un pion de 2 après son premier coup	Le mouvement n'est pas effectué	OUI
Prendre une pièce directement en diagonale	Le pion se déplace sur la case et l'autre pièce est détruite	OUI
Prendre en passant (Si un pion adverse a fait un déplacement de 2 le tour précédent et que s'il avait fait un déplacement de 1 notre pion aurait pu le prendre alors on se déplace derrière ce pion adverse et on le prend)	Notre pion se déplace derrière le pion adverse et ce dernier est détruit	OUI
Avancer sur la dernière ligne (promotion)	Le pion se déplace sur la case et une fenêtre s'ouvre demandant quelle promotion on veut effectuer.	OUI
Essayer d'avancer sur une case occupée	Le mouvement n'est pas effectué	OUI

Action effectuée	Résultat attendu	Résultat obtenu juste ?
Essayer d'avancer un pion de 2 à son premier coup alors que la case entre la cible et le pion est occupée	Le mouvement n'est pas effectué	OUI
Roi		
Déplacer le roi sur une des 8 cases adjacentes inoccupé et qui ne sont pas mise en échec	Le roi se déplace	OUI
Déplacer le roi sur une des 8 cases adjacentes occupée par un allié	Le roi ne se déplace pas	OUI
Déplacer le roi sur une des 8 cases adjacentes inoccupé et qui sont en échec	Le roi ne se déplace pas	OUI
Déplacer le roi sur une des 8 cases adjacentes occupée par un adversaire et dont la case n'est pas mise en échec	Le roi se déplace et prends la pièce	OUI
Déplacer le roi sur une des 8 cases adjacentes occupée par un adversaire et dont la case est mise en échec	Le roi ne se déplace pas et la pièce adverse ne subit rien	OUI
Faire un roque (grand et petit) lorsque le roi n'a jamais bougé, que la voie est libre, que ni le roi, ni les cases entre le roi et le roque ne soit en échec et que la tour du roque n'ait jamais bougé (2 tests)	Le roque est effectué	OUI
Tenter un Roque lorsqu'une des conditions listées ci-dessus n'est pas en place (10 tests)	Le roque n'est pas effectué	OUI
Dame		
Déplacement de la dame de 1 sur toutes les cases autour d'elle (cases vide)	La pièce est déplacée	OUI
Déplacement de la dame de n cases dans toute les direction (sans pièce dans le chemin)	La pièce est déplacée	OUI
Déplacement de la dame sur une case contenant une pièce alliée	La dame ne bouge pas	OUI
Déplacement de la dame sur un case contenant une pièce ennemie	La dame est déplacée et la pièce sur la case d'arrivée est supprimée	OUI
Déplacement de la dame de n cases, mais avec une pièce quelconque entre elle est la case d'arrivée	La dame ne bouge pas	OUI
Fou		
Déplacement sur une case vide en diagonale de n case (sans pièces sur le chemin)	La pièce est déplacée	OUI
Déplacement sur une case occupée par un allié en diagonale de n case (sans pièces sur le chemin)	La pièce n'est pas déplacée	OUI
Déplacement sur une case occupée par un adversaire en diagonale de n case (sans pièces sur le chemin)	La dame est déplacée et la pièce sur la case d'arrivée est supprimée	OUI
Déplacement en diagonale de n case (avec pièces sur le chemin)	La pièce n'est pas déplacée	OUI
Tour		
Déplacement sur une case vide en ligne de n case (sans pièces sur le chemin)	La pièce est déplacée	OUI
Déplacement sur une case occupée par un allié en diagonale de n case (sans pièces sur le chemin)	La pièce n'est pas déplacée	OUI

Action effectuée	Résultat attendu	Résultat obtenu juste ?
Déplacement sur une case occupée par un adversaire en diagonale de n case (sans pièces sur le chemin)	La dame est déplacée et la pièce sur la case d'arrivée est supprimée	OUI
Déplacement en ligne de n case (avec pièces sur le chemin)	La pièce n'est pas déplacée	OUI
Cavalier		
Déplacement du cavalier d'une case horizontalement et de deux cases verticalement sur une case libre	La pièce est déplacée	Oui
Déplacement du cavalier d'une case verticalement et de deux cases horizontalement sur une case libre	La pièce est déplacée	OUI
Déplacement du cavalier d'une case horizontalement et de deux cases verticalement sur une occupée par un allié	La pièce ne bouge pas	OUI
Déplacement du cavalier d'une case horizontalement et de deux cases verticalement sur une case occupée par un ennemi	la pièce est déplacée et l'autre pièce est mangée	OUI
Déplacement du cavalier d'une case verticalement et de deux cases horizontalement sur une case libre par-dessus une autre pièce	La dame ne bouge pas	OUI
Déplacement du cavalier d'une case horizontalement et de deux cases verticalement sur une case libre	La pièce est déplacée	Oui