



Université Claude Bernard Lyon 1



# QChess

## Compte-rendu de projet

Haidari Omran - Thomas Dylan - Pittore Axel

Professeur : M. HADDAD

3A Informatique



# Sommaire

<b>Introduction</b>	<b>2</b>
<b>Les classes de base</b>	<b>3</b>
<b>L'intelligence artificielle</b>	<b>4</b>
<b>L'interface graphique</b>	<b>5</b>
<b>Annexes</b>	<b>6</b>
Légende de l'interface graphique :	6
<b>Conclusion</b>	<b>7</b>



# I. Introduction

Notre projet consiste en un jeu d'échec avec une intelligence artificielle ayant plusieurs niveaux de difficulté. Ce jeu d'échec comporte des spécificités. En effet, il y a seulement une tour, un cavalier, un fou et un roi comme pièces majeures. Il ne comporte pas de dame. De plus, les pions sont rangées sur deux lignes. Nous devons également prendre en compte la promotion de la pièce, le saut de deux cases de la pièce ainsi que le mouvement de roque entre le roi et la tour.

Concernant l'interface graphique, nous devons afficher le plateau de jeu, l'historique des déplacements ainsi que l'historique des pièces prises.



## II. Les classes de base

Le jeu d'échec ayant plusieurs pièces, chacune ayant sa particularité, il a été nécessaire de créer une classe propre à chacune pour traiter au cas par cas. Chaque pièce est une classe fille de la classe abstraite Pièce. Cette classe contient des informations communes à chaque pièce comme le type de pièce, sa position, sa couleur, un booléen pour savoir si la pièce a déjà bougé (utile pour les pions ou la tour et le roi pour le roque). Enfin chaque pièce détient un tableau de coordonnées permettant de calculer ses possibilités de déplacement en fonction de la position de la pièce. Une pièce ne pourra pas se déplacer si elle sort du plateau ou qu'une pièce de la même couleur se situe sur la case d'arrivée choisie.

Ces pièces seront gérées une fois sur le plateau (classe Board) qui détient l'ensemble des pièces de bases. C'est à partir de ce plateau que l'on vérifiera la possibilité pour une pièce de se déplacer. Ce plateau est composé de cases (classe Tile), chacune contenant les infos concernant sa position et la vérification si elle est occupée ou pas.

Une classe Move est utilisée pour gérer les déplacements des pièces et ainsi vérifier le sens de déplacement et si il s'agit d'une attaque pour manger une autre pièce.

Tout cet ensemble permet notre gestion des actions de chaque joueur. Les joueurs blancs sont gérés dans la classe WhitePlayer et les noirs dans BlackPlayer.

Ces deux classes héritent de la classe Player qui contient les mouvements possibles pour le joueur, le roi du joueur afin de permettre de vérifier si il est en échec. Cette classe dispose donc de toutes les fonctions de calcul de déplacement des différents pions du joueur sur les différentes cases du plateau.





### III. L'intelligence artificielle

Nous avons décidé de créer une intelligence artificielle en implémentant l'algorithme MinMax dans notre jeu d'échec. La classe MiniMax va servir à implémenter l'algorithme, afin de calculer le meilleur coup possible. Celle-ci prend en compte le résultat de la différence du score des blancs avec le score des noirs. Si le résultat est positif, les blancs ont l'avantage, si le résultat est négatifs, les noirs ont l'avantage. Si c'est égale à 0, aucun des deux n'a l'avantage. Ce score est retrouvé en allant visiter tous les mouvements possibles après *depth* étapes.

Il y alors *depth* niveaux de difficulté possibles. Plus le *depth* est grand, plus la difficulté est grande, et plus le temps de calcul est grand.



## IV. L'interface graphique

Les différents éléments de l'interface ont été séparés afin de gérer au mieux chaque partie. Le tout a été réalisé avec la bibliothèque graphique Swing, celle-ci étant mieux maîtrisée par l'ensemble du groupe.

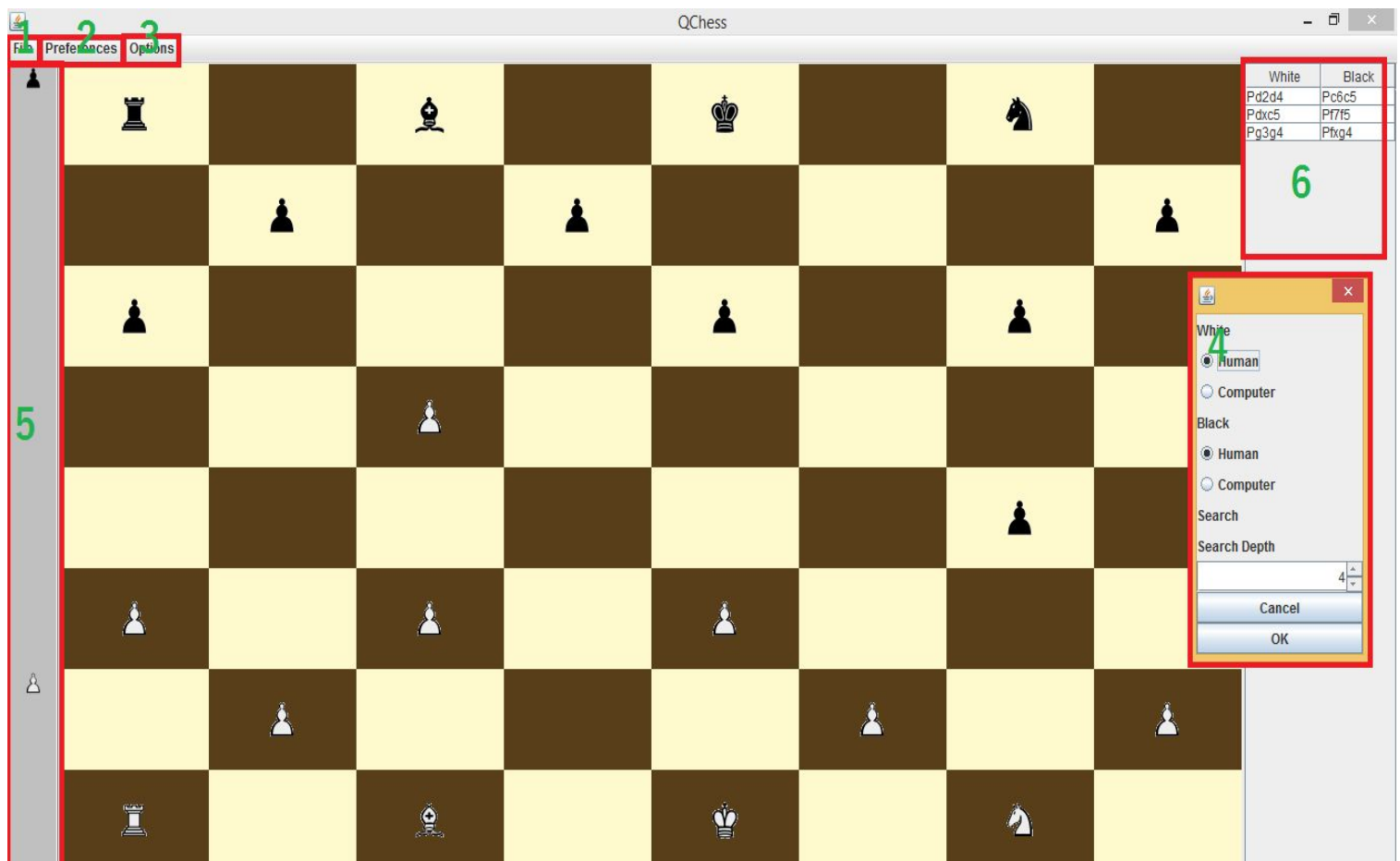
Le classe Table contient le plateau principal, celui qui affichera les informations des classes de bases et qui récupèrera l'action de l'utilisateur sur l'interface.

La classe GameHistoryPanel concerne l'affichage des mouvements réalisés par chaque joueur et rend état de l'avancée de la partie afin de permettre aux joueurs de ne pas perdre le fil du jeu.

Enfin la classe TakenPiecePanel permet l'affichage des pièces capturées par soi-même et l'adversaire.



## V. Annexes



### Légende de l'interface graphique :

1. File : permet de quitter l'application
2. Preferences : permet d'inverser l'échiquier, et de préciser si l'on veut voir les déplacements possibles quand une pièce est sélectionnée
3. Options : permet d'afficher la fenêtre 4 (Setup Game)
4. Fenêtre Setup Game : Permet de définir les joueurs et la difficulté de l'IA (*depth*)
5. Historiques des pièces prises
6. Historiques des coups joués



## VI. Conclusion

Nous avons réussi à développer le jeu d'échec avec les spécificités demandées. Nous avons pu afficher sur le plateau les différents déplacements autorisés spécifiques à chaque pièce. Nous avons pris en compte le saut de deux cases du pion, la promotion du pion et le mouvement de roque. Le mouvement de roque a été implémenté mais n'est pas affiché sur le plateau de jeu.

Concernant l'interface graphique, nous avons utilisé la bibliothèque graphique de Java : Swing. Le plateau de jeu, l'historique des déplacements, l'historique des pièces prises sont affichées. Il y a également un menu qui permet de choisir les joueurs, ainsi que le niveau de difficulté de l'intelligence artificielle.

