

Projet : Jeu d'Échecs



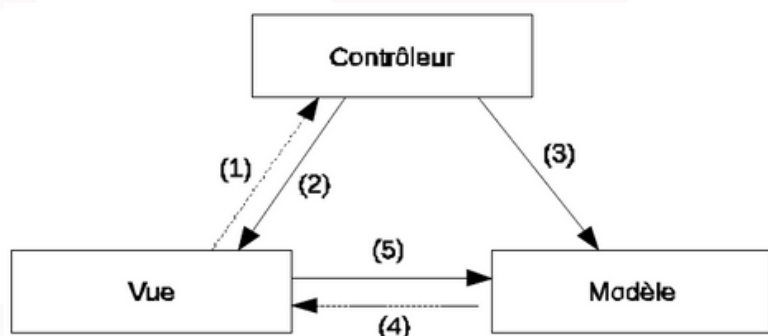
Chafae GALLOUJ & Abir HANNED

POLYTECH LYON
3A INFORMATIQUE

Introduction

Ce projet a consisté à développer une application de jeu d'échecs en Java en suivant le modèle MVC (Modèle-Vue-Contrôleur), en respectant les principes de la programmation orientée objet (POO). Nous avons utilisé Swing pour l'interface graphique et intégré le Design Pattern Observer pour synchroniser modèle et vue.

L'objectif initial était de créer une version fonctionnelle du jeu permettant à deux joueurs de s'affronter, puis de proposer des extensions pour enrichir l'expérience de jeu.



Architecture MVC

Notre projet est organisé en plusieurs packages et dossiers, chacun ayant une fonction précise :

Contrôleur :

Ce dossier contient les classes chargées de gérer les interactions entre l'utilisateur et le système. il y a notamment :

ContrôleurConsole.java qui gère la saisie et les commandes en mode console, permettant de jouer sans interface graphique,

ContrôleurEchiquier.java qui traite les clics et événements dans la version graphique, transmettant les actions aux composants du modèle et actualisant la vue en conséquence.

Modele :

C'est ici que réside toute la logique métier et les règles du jeu d'échecs. ce package est le cœur du programme :

il contient les classes représentant les pièces (Piece, Tour, Fou, etc.) avec leurs déplacements spécifiques,

la classe Plateau qui représente le damier et sa configuration,

Jeu qui orchestre la partie, vérifie les règles (échec, mat, promotion, roque, prise en passant),

Joueur qui représente un participant (humain ou IA),

un sous-package decorateur qui ajoute des fonctionnalités spécifiques aux pièces, comme la mise en évidence des cases accessibles,

un sous-package ia qui regroupe l'intelligence artificielle, notamment la classe IAJoueurMinimax.java qui décide des coups de l'ordinateur

la structure du projet

```
+---captures
|
| echiquier_fin_20250518_171
| 106.png
+---Images
\---src
    +---contrôleur
    | ContrôleurConsole.java
    | ContrôleurEchiquier.java
    +---modele
    | | Case.java
    | | Coup.java
    | | Jeu.java
    | | Joueur.java
    | | PGNExporter.java
    | | Piece.java
    | | Plateau.java
    | |
    | +---decorateur
    | | DecCavalier.java
    | | DecFou.java
    | | DecPion.java
    | | DecReine.java
    | | DecRoi.java
    | | DecTour.java
    | | DesCasesAccessibles.java
    | |
    | +---ia
    | | IAJoueurMinimax.java
    | |
    | \---Pieces
    | Cavalier.java
    | Fou.java
    | Pion.java
    | Reine.java
    | Roi.java
    | Tour.java
    |
    \---Vue
    InterfaceUtilisateur.java
    VueConsole.java
    VueEchiquier.java
    VueGraphique.java
    Main.java
    Demo.java
+---sauvegardes
|
| partie_20250518_171106.pgn
```





Vue :

Ce package contient tout ce qui concerne l’affichage et l’interface utilisateur :

- VueEchiquier.java qui affiche le plateau en version graphique avec les pièces et gère les interactions visuelles,
- VueConsole.java qui offre une interface textuelle pour jouer au jeu dans un terminal.

Pour tester une règle précise du jeu, il suffit d’exécuter la classe Demo.java. elle propose des scénarios ciblés (ex : roque, prise en passant, etc.) que l’utilisateur peut tester via des clics.

si vous voulez lancer une partie complète, il faut exécuter la classe Main.java, qui initialise tout le jeu avec l’interface graphique ou console.

Fonctionnalités et Extensions

Nous avons structuré le développement du projet en plusieurs étapes, correspondant aux fonctionnalités de base et aux extensions avancées. Chaque fonctionnalité a été estimée en termes de complexité et de temps de développement, exprimée en pourcentage approximatif de l'effort total.

Gestion du plateau (15 %)

Création d’un échiquier de 64 cases avec placement initial des pièces selon les règles officielles.

Coups spéciaux (25 %)

Roque, prise en passant, promotion et détection de l’échec et mat.

Vue console (10 %)

Version textuelle du jeu permettant de jouer dans un terminal, utile pour les tests et le débogage.

Intelligence artificielle (10 %)

IA basée sur l’algorithme Minimax, capable de jouer contre un humain en évaluant les positions.

Mouvements classiques (20 %)

Implémentation des déplacements standards pour chaque type de pièce, avec vérification de la validité des coups.

Vue graphique (15 %)

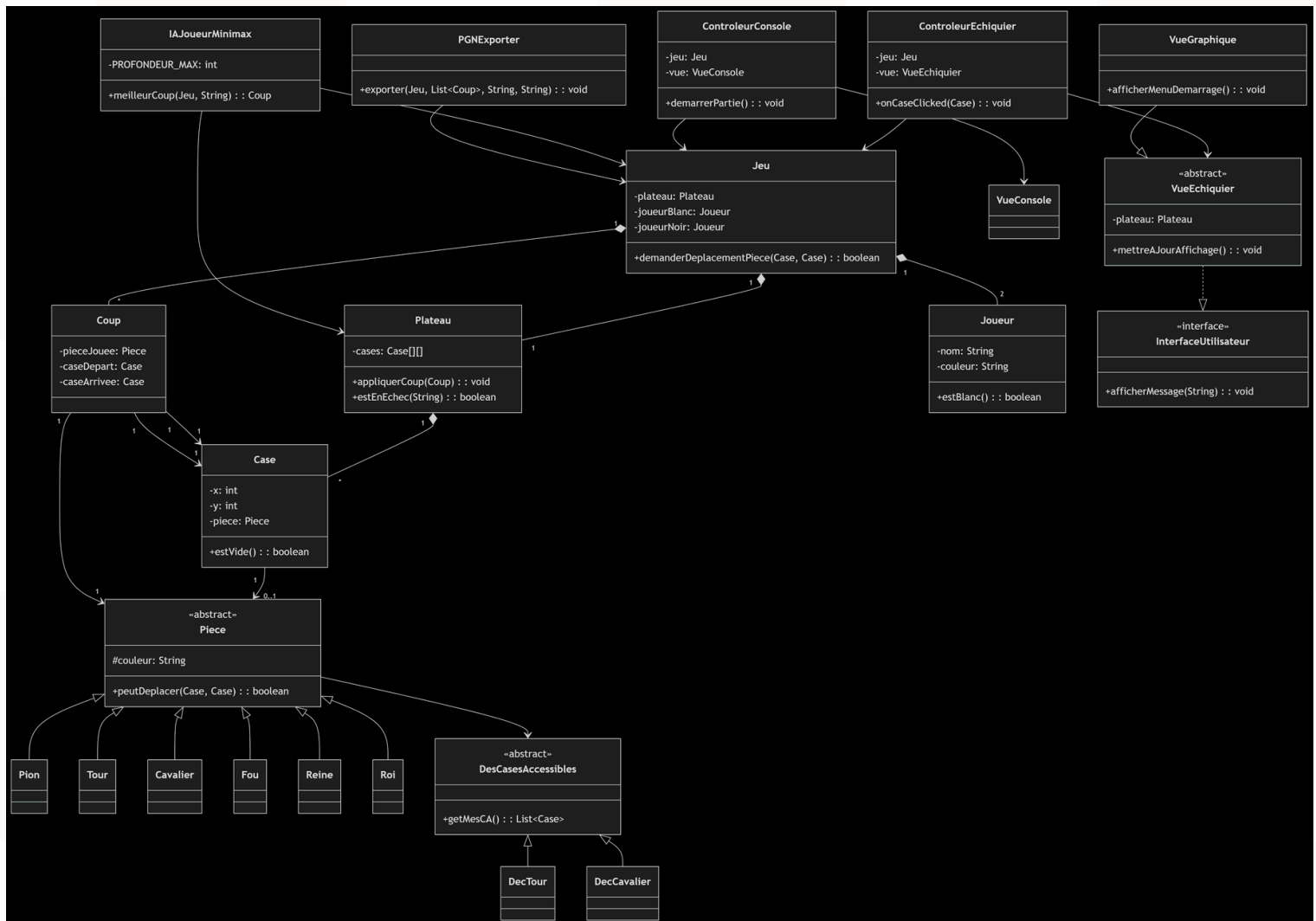
Interface Swing cliquable affichant le plateau et les pièces, avec mise à jour dynamique après chaque coup.

Sauvegarde PGN & export PNG (5 %)

Enregistrement des parties au format PGN et capture d’écran du plateau exportée en PNG.



Diagramme UML :



Déroulement du jeu

Lancement du jeu – Main.java

Lors de l'exécution de Main.java, une fenêtre s'ouvre et invite l'utilisateur à choisir entre deux modes d'affichage :

- Vue graphique : avec une interface visuelle et des interactions à la souris
- Vue console : en ligne de commande.

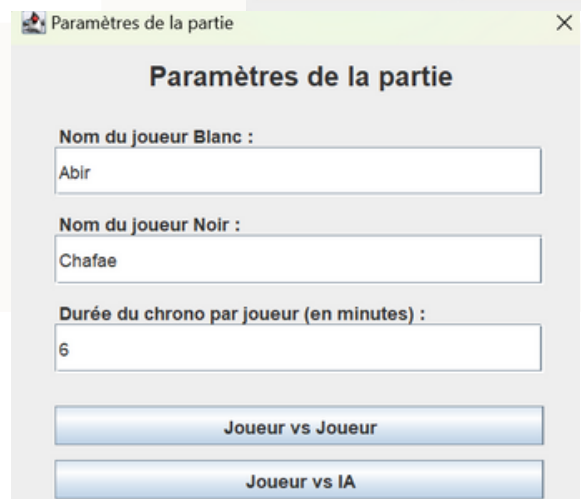
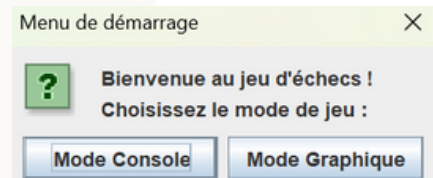
Si l'utilisateur opte pour la vue graphique, un formulaire s'affiche afin de :

- saisir le nom du joueur blanc ;
- saisir le nom du joueur noir ;
- définir la durée de la partie (en minutes).

Un autre choix est également proposé à l'utilisateur :

→ Souhaitez-vous jouer un match joueur contre joueur, ou joueur contre IA ?

Si l'option joueur contre IA est sélectionnée, l'intelligence artificielle jouera systématiquement les pièces noires.

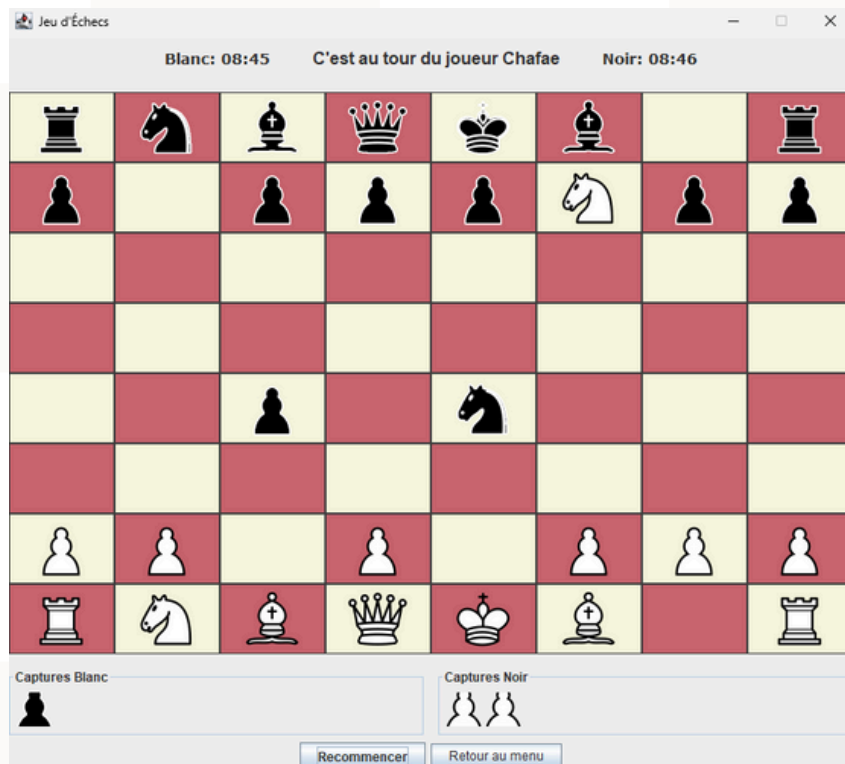


Démarrage de la partie

une fois tous les éléments configurés, le jeu se lance avec :
l'échiquier initialisé ;
les deux joueurs créés (humains ou IA) ;
le chronomètre activé (si il est pas défini, il prend par défaut 5min).

Démonstration des règles du jeu

Toutes les règles officielles du jeu d'échecs ont été implémentées.
déplacement de base de chaque pièce (pion, cavalier, etc.) :



Prise en passant

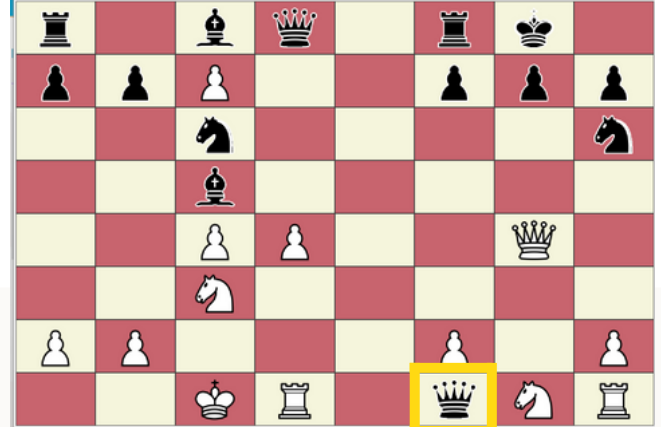


Roque court et long





Promotion du pion



Détection d'échec



Echec et mat



Conclusion

Ce projet a permis de mettre en pratique plusieurs concepts fondamentaux de l'ingénierie logicielle, notamment l'architecture MVC, la POO, les design patterns (Observer, Décorateur) et la conception d'interfaces graphiques.

Il offre une base solide pour un jeu d'échecs évolutif, intégrant à la fois des interactions en console et une interface graphique, ainsi qu'une première version d'intelligence artificielle.

De nombreuses améliorations sont encore possibles, comme l'ajout d'un système de sauvegarde/chargement plus avancé, une IA plus sophistiquée, ou encore une interface utilisateur plus ergonomique, mais les fondations sont désormais posées pour continuer à faire évoluer l'application.

