Rapport projet PR70

Table des matières

Objectif du Projet

L'objectif de ce projet est de concevoir un jeu d'échecs doté d'une interface graphique complète. Le jeu doit inclure trois modes principaux :

- **Mode joueur contre joueur** : deux joueurs jouent à tour de rôle.
- **Mode 1 joueur** : un joueur affronte un ordinateur.
- **Mode chargement de partie** : possibilité de charger une partie enregistrée, que ce soit en mode joueur contre joueur ou joueur contre ordinateur.

Les fonctionnalités suivantes doivent être implémentées :

- Lancer une nouvelle partie.
- Sauvegarder une partie en cours.
- Charger une partie enregistrée.
- Afficher le plateau de jeu avec les pièces dans leur position initiale.
- Permettre aux joueurs de déplacer leurs pièces en respectant les règles du jeu d'échecs.
- Vérifier la validité des coups et signaler les éventuelles erreurs.
- Identifier les différentes situations de jeu (par exemple : échec, échec et mat, pat, etc.).
- Afficher l'historique des coups effectués et le score actuel.
- Annoncer la fin de la partie en indiquant le gagnant ou un match nul.

Implémentation

Le projet est organisé en quatre packages, chacun regroupant un ensemble de fonctionnalités bien définies. Cette structuration vise à améliorer la lisibilité du projet et à faciliter les modifications éventuelles.

L'objectif principal de cette organisation est de séparer clairement la partie graphique (interface utilisateur) de la partie logique (gestion des règles et du fonctionnement interne du jeu).

Cette séparation contribue à rendre le code plus modulaire, compréhensible et maintenable, tout en respectant les bonnes pratiques de la programmation orientée objet.

1. Package Logic

Le package Logic contient les classes responsables de la logique du jeu d'échecs. Il gère les règles du jeu, les mouvements des pièces et les conditions de victoire.

Classe Game :

- Gère les règles du jeu d'échecs, y compris les mouvements des pièces, les conditions de victoire et les états de la partie.
- Vérifie si un roi est en échec, échec et mat ou pat.
- Contient des méthodes pour changer de couleur.

Classe Move :

- Gère les mouvements des pièces sur le plateau.
- Vérifie la légalité des mouvements et génère tous les mouvements légaux possibles pour une pièce donnée.

Classe Piece :

- Représente une pièce d'échecs.
- Contient des informations sur le type de pièce, sa couleur et sa position actuelle.

- Gère les mouvements spécifiques à chaque type de pièce.

Classe Board :

- Représente le plateau de jeu.
- Gère l'affichage et la disposition des cases et des pièces sur le plateau.

Classe GameTimer :

- Gère les chronomètres pour chaque joueur, permettant de jouer avec une limite de temps.
- Affiche le temps restant pour chaque joueur et déclenche des alertes lorsque le temps est écoulé.

Classe Historize :

- Gère l'historique des mouvements effectués pendant la partie.
- Permet de visualiser les mouvements précédents et de revenir en arrière si nécessaire.

Classe Mouse :

- Gère les événements de la souris pour l'interaction avec le plateau de jeu.
- Permet de sélectionner et de déplacer les pièces.

Classe Save :

- Gère la sauvegarde et le chargement des parties.
- Permet de sauvegarder l'état actuel du jeu et de le restaurer ultérieurement.

Classe Score :

- Gère le score des joueurs.
- Calcule et affiche le score en fonction des pièces capturées et des conditions de victoire.

1. Difficultés Rencontrées :

- Gestion des Règles du Jeu : Assurer que toutes les règles du jeu d'échecs soient correctement implémentées et vérifiées a été un défi, en particulier pour les mouvements spéciaux comme le roque et la prise en passant.
- Optimisation des Mouvements : Générer et vérifier tous les mouvements légaux de manière efficace a nécessité des optimisations pour éviter les ralentissements.

2. Package Panel

Le package Panel contient les classes responsables de l'interface utilisateur de l'application d'échecs. Il gère l'affichage des différents écrans et les interactions de l'utilisateur avec le jeu.

Classe PanelManager :

- Gère les différents panneaux de l'application, y compris le panneau de titre et le panneau de jeu.
- Initialise la fenêtre principale de l'application et définit le panneau de titre comme vue par défaut.
- Permet de lancer une nouvelle partie et de revenir au panneau de titre.

Classe TitlePanel :

- Affiche le titre du jeu et les options pour démarrer une nouvelle partie.
- Gère les événements de clic pour lancer le jeu avec le mode sélectionné.

Classe GamePanel :

- Affiche le plateau de jeu et gère les interactions de l'utilisateur avec les pièces.

- Met à jour l'affichage en fonction des mouvements des pièces et des événements du jeu.

Classe HistorizePanel :

- Affiche l'historique des mouvements effectués pendant la partie.

- Permet de visualiser les mouvements précédents et de revenir en arrière si nécessaire.

Classe Board :

- Représente le plateau de jeu.

- Gère l'affichage et la disposition des cases et des pièces sur le plateau.

Classe Button :

- Représente un bouton interactif dans l'interface utilisateur.

- Gère les événements de clic pour les actions spécifiques de l'interface.

Classe Frame :

- Gère la fenêtre principale de l'application.

- Contient les méthodes pour initialiser et afficher les différents panneaux de l'application.

1. Difficultés Rencontrées :

- Gestion des Événements de la Souris : Assurer une interface utilisateur réactive et intuitive a été un défi, en particulier pour la gestion des événements de la souris et la mise à jour dynamique de l'interface.

- Mise à Jour de l'Interface : Maintenir une interface graphique fluide et sans décalage lors des mouvements des pièces a nécessité des optimisations.

3. Package Piece

Le package Piece contient les classes représentant les différentes pièces du jeu d'échecs. Chaque classe de pièce hérite d'une classe de base Piece et implémente les mouvements spécifiques à chaque type de pièce.

Classe Piece :

Classe de base pour toutes les pièces d'échecs.

Contient des attributs communs tels que la couleur, la position (colonne et rangée), et le type de pièce.

Méthodes pour vérifier les mouvements valides et les captures.

Classes de Pièces Spécifiques :

Rook (Tour) : Implémente les mouvements horizontaux et verticaux.

Bishop (Fou) : Implémente les mouvements diagonaux.

Knight (Cavalier) : Implémente les mouvements en "L".

Queen (Reine) : Combine les mouvements de la tour et du fou.

King (Roi) : Implémente les mouvements d'une case dans toutes les directions.

Pawn (Pion) : Implémente les mouvements de base, la capture en diagonale, et la promotion.

Gestion des Mouvements :

Chaque classe de pièce a une méthode canMove peut bouger en fonction de ces propriétés

La fonction isLegalMove vérifie quand à elle que le mouvement ne permet pas a l'adversaire de mangé le roi.

Type :

Identification de chaque pièces

4. Package Computer

Le package Computer contient les classes et les algorithmes nécessaires pour implémenter une intelligence artificielle (IA) capable de jouer aux échecs contre un joueur humain. Ce package gère la logique de prise de décision de l'IA en analysant les positions des pièces et en choisissant les meilleurs mouvements possibles.

Évaluation des Positions :

- L'IA évalue les positions en attribuant des scores aux différentes configurations de pièces sur l'échiquier.

- Les scores sont basés sur des critères tels que la valeur matérielle des pièces, le contrôle du centre, la sécurité du roi, etc.

Algorithmes de Recherche :

- Minimax : Un algorithme de recherche qui explore les mouvements possibles jusqu'à une certaine profondeur et évalue les positions résultantes.

- Élagage Alpha-Bêta : Une optimisation de l'algorithme Minimax qui réduit le nombre de positions à évaluer en éliminant les branches non prometteuses.

Difficultés Rencontrées :

- Complexité de l'Algorithme : Implémenter et optimiser l'algorithme Minimax avec élagage alpha-bêta a été un défi en raison de la complexité et des exigences en termes de calcul.

- Évaluation des Positions : Développer une fonction d'évaluation efficace et précise pour les positions d'échecs a nécessité de nombreux ajustements et tests.

Difficultés générales rencontrées

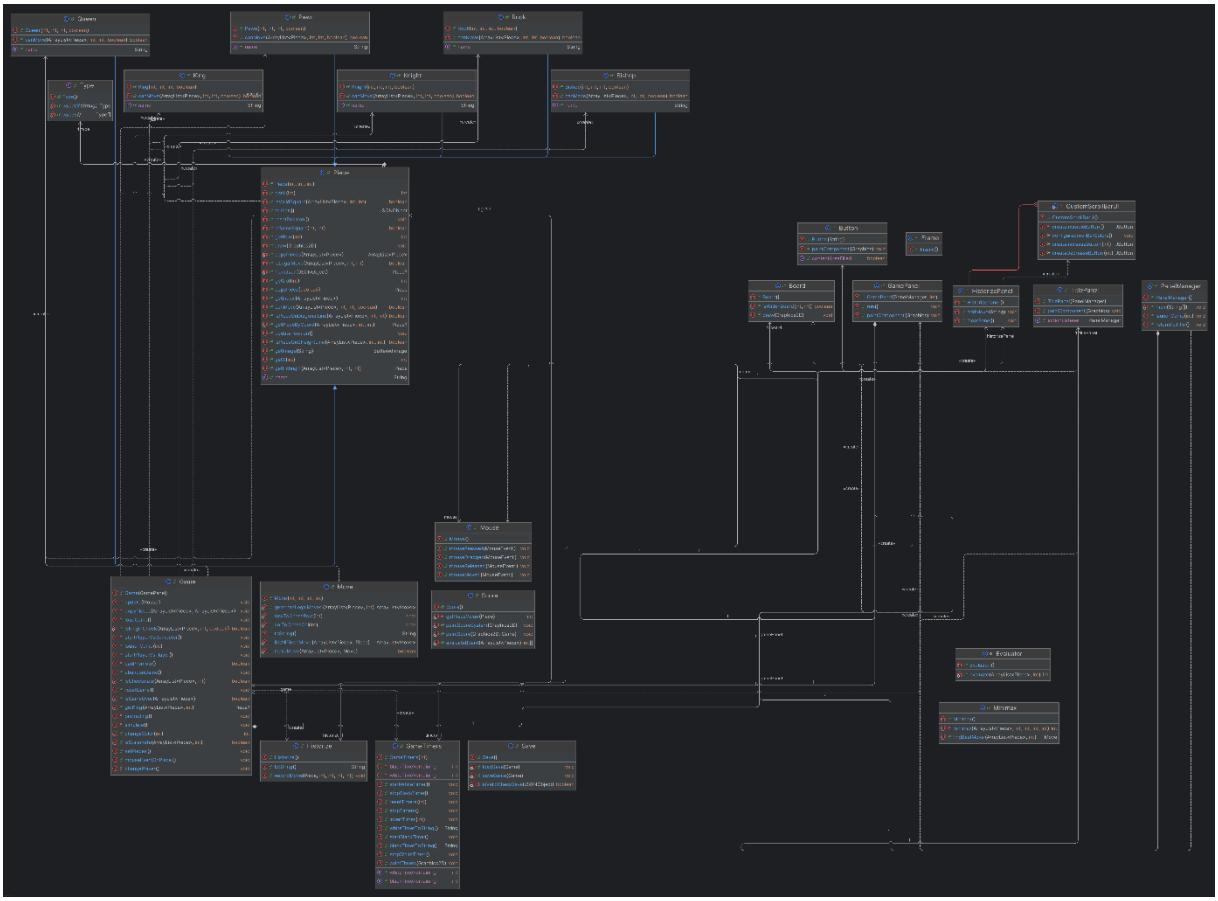
Au début du projet, la logique du jeu et les interfaces graphiques étaient mélangées dans une seule classe comptant plus de 1 500 lignes de code. À mesure que le projet progressait, il devenait de plus en plus difficile de s'y retrouver, notamment lorsqu'un partenaire apportait des modifications.

Face à ces difficultés, nous avons décidé, à mi-parcours, de réorganiser l'ensemble du projet en tirant pleinement parti des principes de la programmation orientée objet. Cette restructuration nous a conduits à créer des classes distinctes pour chaque concept, ce qui a considérablement amélioré la clarté et la maintenabilité du projet.

Un autre défi majeur a été le contrôle de la validité des mouvements. En effet, les types de mouvements possibles sont très nombreux, et certains mouvements sont conditionnés par des règles spécifiques, comme l'interdiction de déplacer une pièce si le roi est en échec, par exemple.

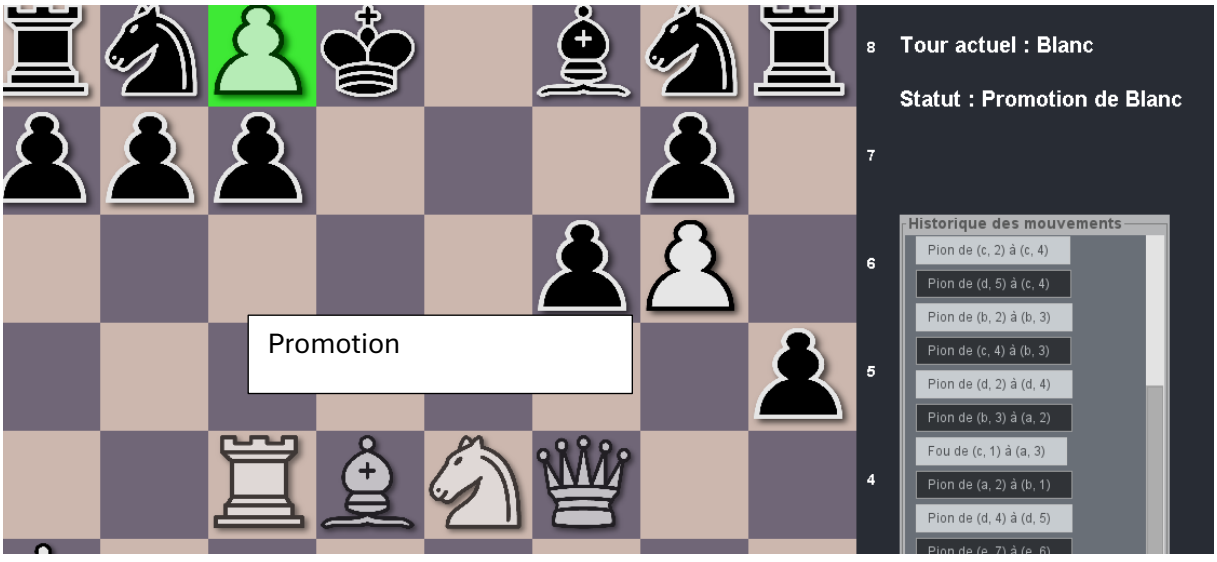
Enfin, la mise en œuvre de l'intelligence artificielle a également posé des difficultés. Ce n'est pas tant la programmation en elle-même qui a été complexe, mais plutôt la compréhension de l'algorithme minimax et de la technique d'élagage alpha-beta pruning.

Diagramme UML

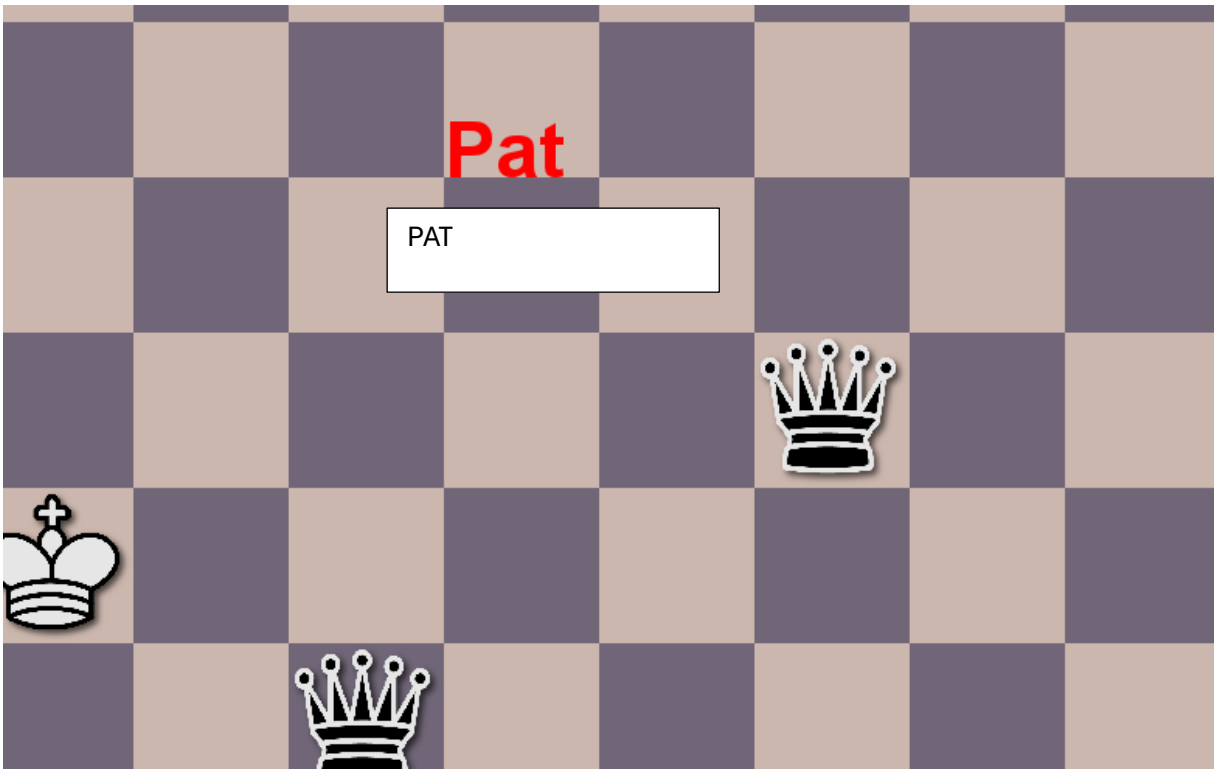


Annexes


Plateau de jeu
Timer
(Joueur vs Joueur)
CHESS
Tour actuel : Noir
Statut : En cours
Echec
Status
Temps restant Blanc : 09:52
Temps restant Noir : 09:45
Score Blanc : 0
Score Noir : 0
Score
Système de points :
Pion : 1 point
Cavalier : 3 points
Fou : 3 points
Tour : 5 points
Dame : 9 points
Roi : Inestimable
Historique des mouvements
Pion de (f, 2) à (f, 4)
Pion de (f, 7) à (f, 6)
Pion de (e, 2) à (e, 3)
Pion de (d, 7) à (d, 6)
Fou de (f, 1) à (e, 2)
Pion de (d, 6) à (d, 5)
Fou de (e, 2) à (h, 5)
Historique
Abandonner
Retour à l'écran titre
Réinitialiser la partie
Sauvegarder la partie
Button
D'action

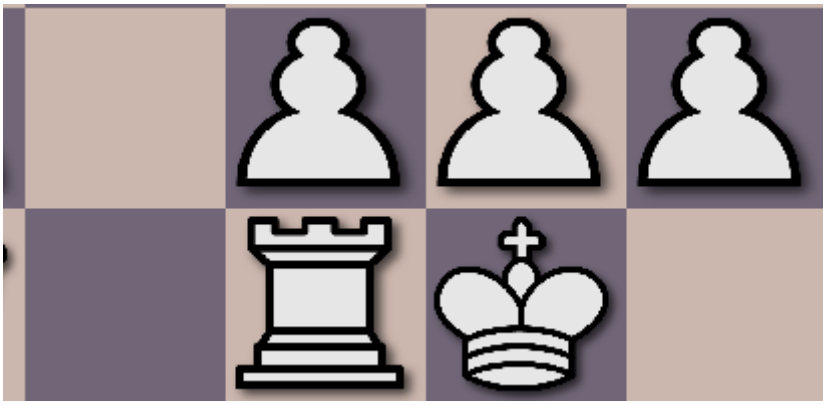

Tour actuel : Blanc
Statut : Promotion de Blanc
Historique des mouvements
Pion de (c, 2) à (c, 4)
Pion de (d, 5) à (c, 4)
Pion de (b, 2) à (b, 3)
Pion de (c, 4) à (b, 3)
Pion de (d, 2) à (d, 4)
Pion de (b, 3) à (a, 2)
Fou de (c, 1) à (a, 3)
Pion de (a, 2) à (b, 1)
Pion de (d, 4) à (d, 5)
Promotion


Echec et mat
Echec et mat

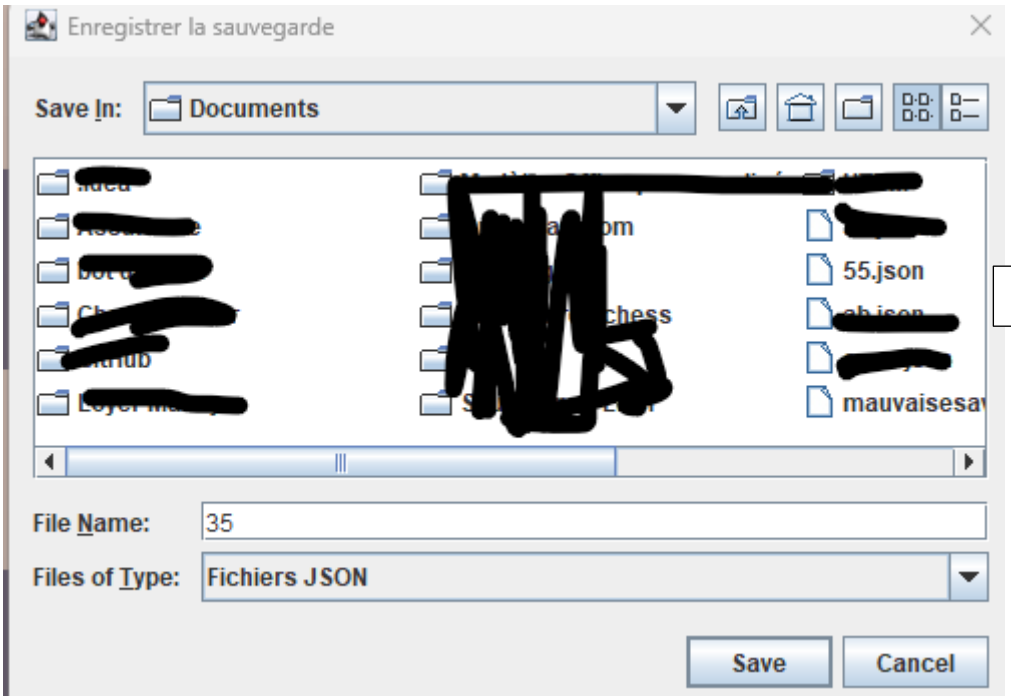

Pat
PAT


Echec
Joueur contre Joueur
Joueur contre Ordinateur
Charger une Partie
TitlePanel

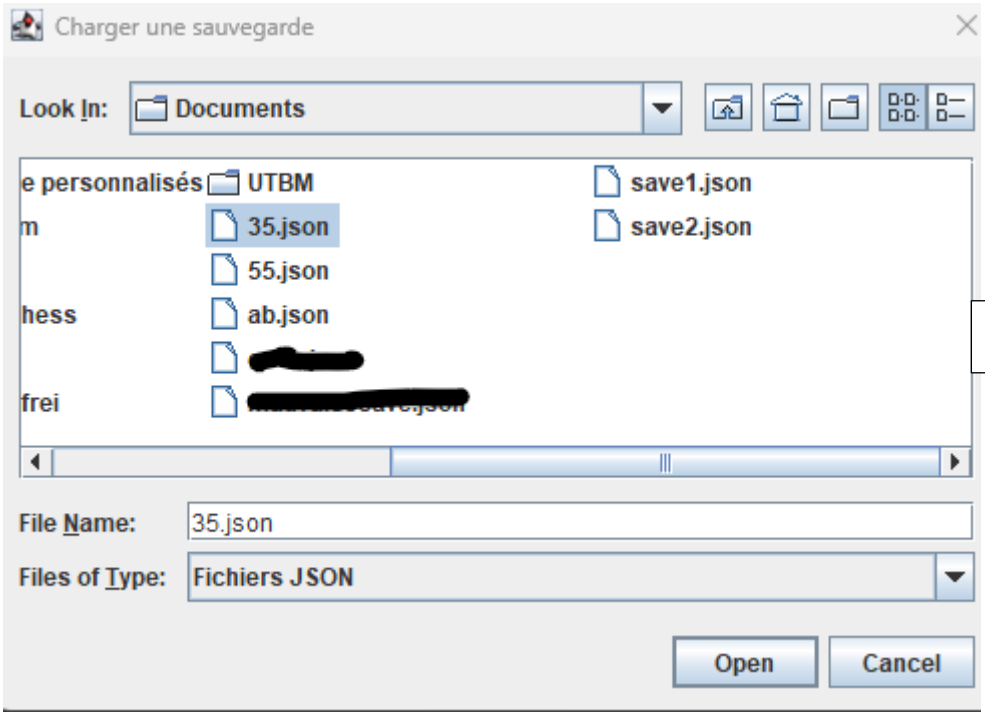
Roque



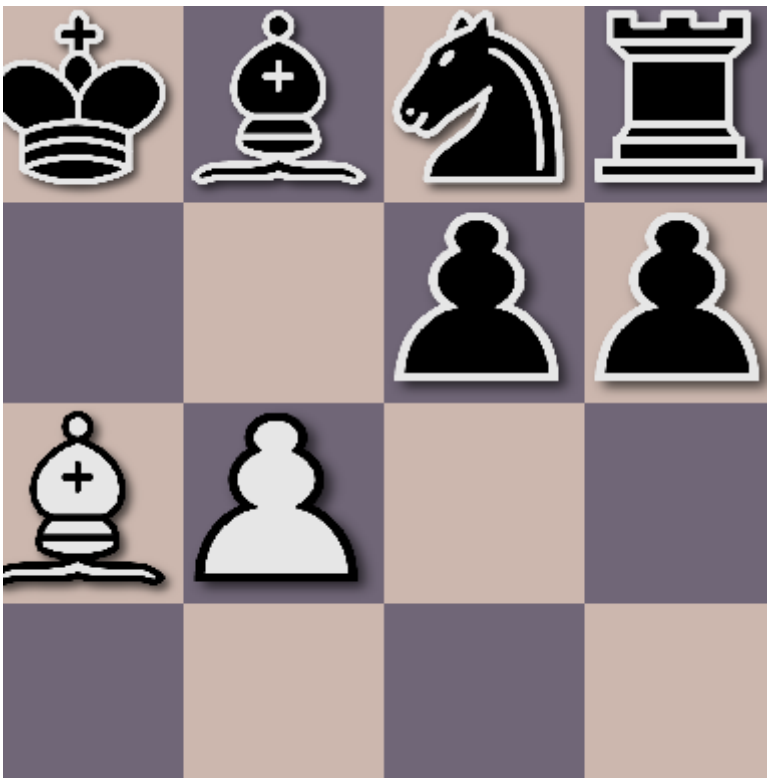

Tour actuel : Noir
Statut : VICTOIRE BLANC
Historique des mouvements
Fou de (f, 1) à (c, 4)
Pion de (d, 7) à (d, 6)
Fou de (c, 4) à (e, 6)
Pion de (d, 6) à (d, 5)
Pion de (e, 4) à (d, 5)
Reine de (d, 8) à (d, 7)
Cavalier de (g, 1) à (f, 3)
Pion de (c, 7) à (c, 6)
Roi de (e, 1) à (g, 1)
Abandon
Abandon



Sauvegarde



Charger



En passant