SAE JAVA S201 :
Implémentation d’un jeu d’échecs

Membres du groupe :

ISHAQ Faizan
RABOUDI Yacine
BUT 1 informatique : Cérynie

Comment lancer le programme :
Dans un terminal de commande Linux, compiler (javac) puis exécuter (java) le fichier fenetrePrincipale.java du répertoire gui. Une fenêtre s’ouvrira alors, débutant le jeu d'échecs.

Diagrammes des classes(UML) :

Case

- ligne: int
- colonne: int
- piece: Piece

+ Case(ligne: int, colonne: int)
+ getLigne(): int
+ getColonne(): int
+ getPiece(): Piece
+ setPiece(piece: Piece): void
+ estVide(): boolean
+ toString(): String

Echiquier

- grille: Case[][]

+ Echiquier()
+ getCase(ligne: int, colonne: int): Case
+ deplacerPiece(depart: Case, destination: Case, joueur: Joueur, adversaire: Joueur): boolean
+ initialiser(joueurBlanc: Joueur, joueurNoir: Joueur): void
+ verifierPromotion(piece: Piece, joueur: Joueur): void
+ estEchecEtMat(joueur: Joueur, adversaire: Joueur): boolean
+ afficherEchiquier(): void

Partie

- echiquier: Echiquier
- joueurBlanc: Joueur
- joueurNoir: Joueur
- joueurCourant: Joueur

+ Partie(nomBlanc: String, nomNoir: String, tempsInitialMillis: long)
+ getJoueurCourant(): Joueur
+ getAdversaire(): Joueur
+ getEchiquier(): Echiquier
+ getJoueurBlanc(): Joueur
+ getJoueurNoir(): Joueur
+ effectuerTour(...): boolean
+ estPat(joueur: Joueur): boolean
+ estEchec(joueur: Joueur): boolean
+ estEchecEtMat(joueur: Joueur): boolean
+ estTerminee(): boolean
+ getVainqueur(): Joueur
+ afficherEtat(): void

Joueur

- nom: String
- estBlanc: boolean
- pieces: List<Piece>
- historique: List<String>
- tempsRestantMillis: long
- debutChrono: long
- chronoEnCours: boolean

+ Joueur(nom: String, estBlanc: boolean, tempsInitialMillis: long)
+ ajouterPiece(piece: Piece): void
+ retirerPiece(piece: Piece): void
+ getPieces(): List<Piece>
+ estBlanc(): boolean
+ getNom(): String
+ demarrerChrono(): void
+ arreterChrono(): void
+ getTempsRestantMillis(): long
+ getTempsFormate(): String
+ aEncoreSonRoi(): boolean
+ getRoi(): Roi
+ ajouterHistorique(coup: String): void
+ getHistorique(): List<String>

+ afficherHistorique(): void
Piece (abstract)

---
estBlanc: boolean
position: Case

+ Piece(estBlanc: boolean, position: Case)
+ Piece(estBlanc: boolean)
+ estBlanc(): boolean
+ getPosition(): Case
+ setPosition(position: Case): void
+ estDeplacementValide(destination: Case, echiquier: Echiquier): boolean [abstract]
+ getSymbole(): char [abstract]

Sous-classes de Piece
Cavalier, Dame, Fou, Pion, Roi, Tour

<<extends>> Piece

+ Constructeurs spécifiques
+ estDeplacementValide(...): boolean
+ getSymbole(): char

CaseGraphique

- ligne: int
- colonne: int
- partie: Partie
- parent: PlateauPanel

+ CaseGraphique(partie: Partie, ligne: int, colonne: int, parent: PlateauPanel)
+ maj(): void
+ getLigne(): int
+ getColonne(): int
PlateauPanel
PlateauPanel

- cases: CaseGraphique[][]
- partie: Partie
- caseSelectionnee: Case

+ PlateauPanel(partie: Partie)

+ caseCliquee(cg: CaseGraphique): void
+ majAffichage(): void

FenetrePrincipale

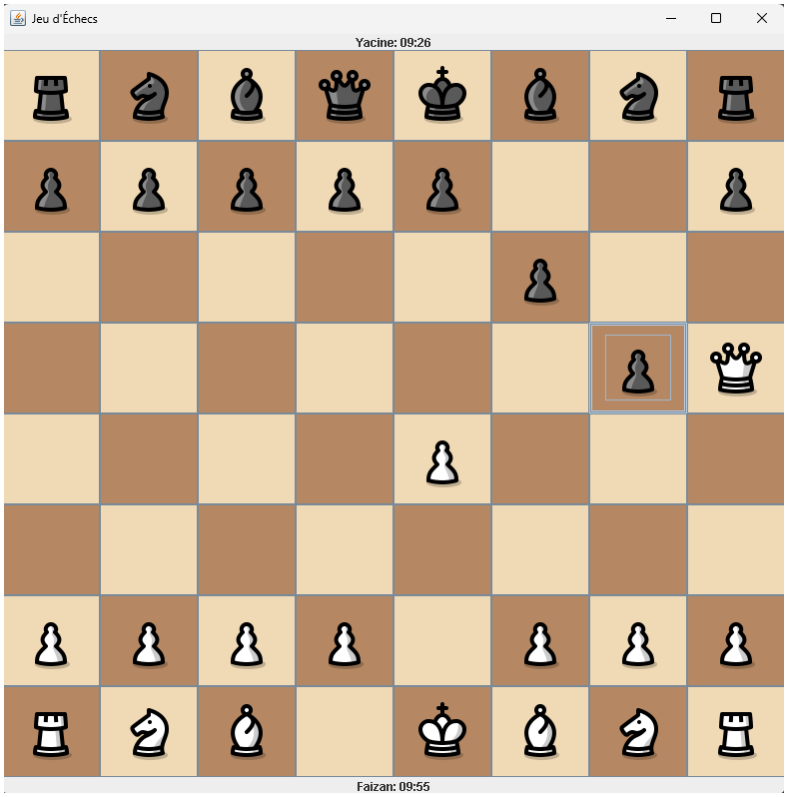
- partie: Partie
- chronoHaut: JLabel
- chronoBas: JLabel
- boutons: JButton[][]
- caseSelectionnee: Case
- partieTerminee: boolean

+ FenetrePrincipale(partie: Partie)
+ gererClic(ligne: int, col: int): void
+ rafraichir(): void
+ main(args: String[]): void

Bilan :
Au lancement du jeu, une interface vous demandera de saisir des pseudonymes et choisira au hasard qui sera blanc, qui sera noir ensuite il vous sera demander un temps en minutes pour initier un temps total de jeu pour chacun des joueurs.
A chaque tour, l'historique des coups joués est affiché dans le terminal.
Globalement, le jeu fonctionne et permet de jouer à une partie d'échec comme sur un réel plateau.
Les différents pions se déplacent correctement et respectent les règles du plateau (un pion bloqué par un autre par exemple).
L'échiquier aussi fonctionne normalement.
Le problème principal réside dans la classe Partie, notamment la fonction echecEtMat et estEchec qui ne fonctionne pas correctement, un problème que nous n'arrivons pas à corriger.

Cela résulte par le mauvais fonctionnement de la fin de la partie : l'écran de fin ne s'affiche pas une fois le roi en échec et mat. Voici des captures d'écran détaillant cela :

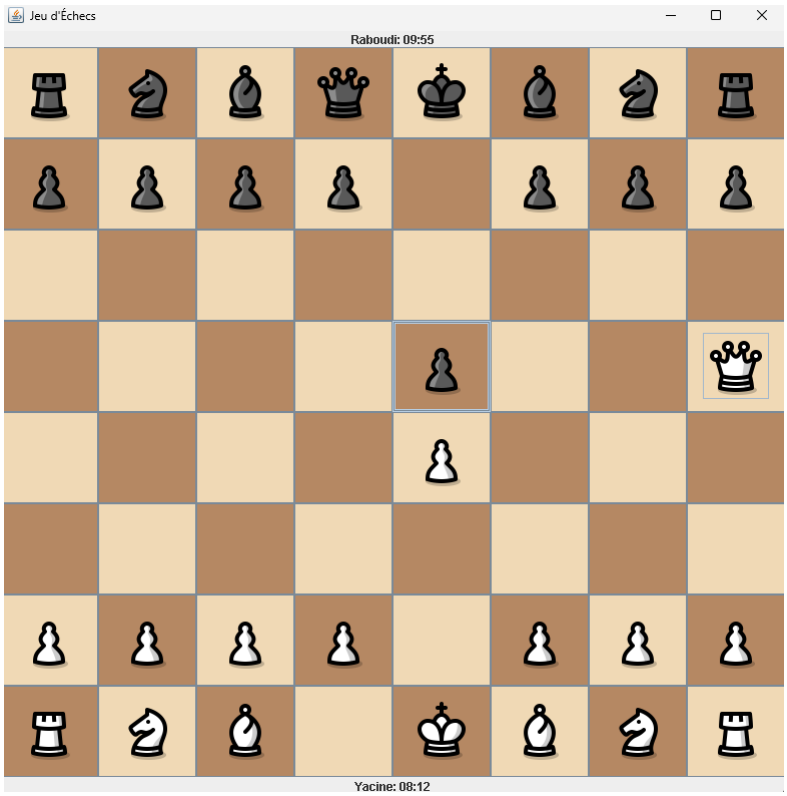
Comme nous pouvons le voir, la méthode estEchec fonctionne bien car je menace le roi mais un pion peut le protéger,



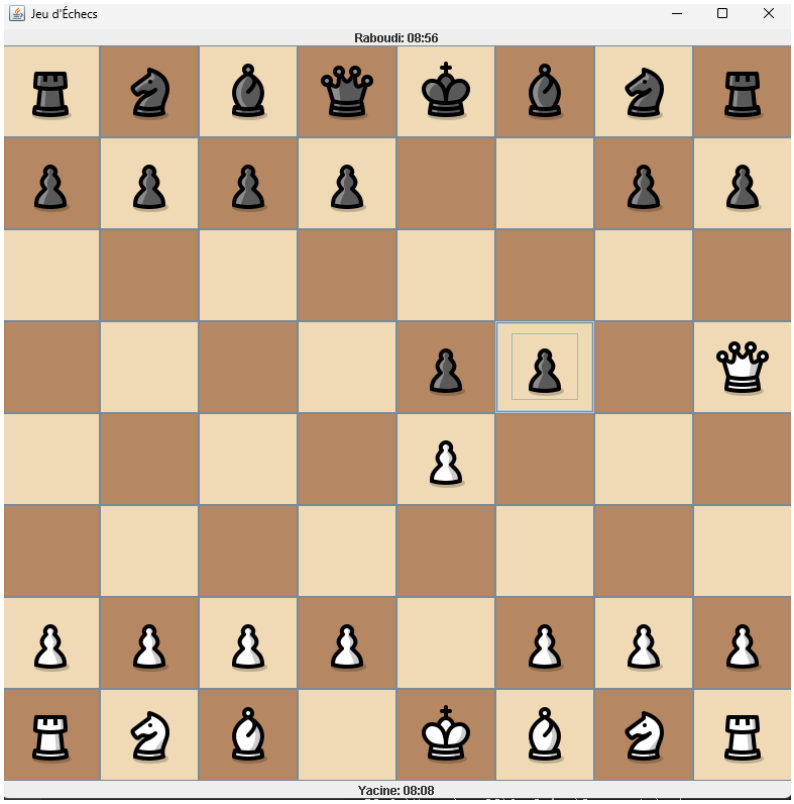
ensuite comme nous le voyons, dans cette situation le roi noir est censé être en échec et mat mais ceci ne fonctionne pas,

J’ai dû bouger un pion pour que ma méthode echecEtMat voit que le roi est en echec et mat et mette fin à la partie.

La méthode estEchec aussi ne marche pas à 100% correctement:



Je mets en place les pièces de cette manière pour mieux visualiser l'erreur,



Ici comme vous pouvez voir le roi noir est en échec mais ça ne l'affiche pas,

Cependant quand je fais un autre coup là il détecte l'échec.

Cela fait qu'il y a deux conditions de fin qui ne fonctionnent pas correctement, sur 3 conditions : Échec et Mat et Temps Écoulé et Pat.

Liste des classes faites par chacun des membres :
R. Yacine : Tout ce qui concerne l'interface graphique, partie, et echiquier.
I. Faizan : pieces, joueur, case, partie (en partie), echiquier (en partie)

Les différents pions du jeu ont été écrits par les deux membres et débuggés ensuite. Le travail a été divisé équitablement.