

Projet : Jeu d'Échecs

[lien wikipedia](#)

Application : développer une application permettant de jouer aux échecs

Modélisation : Modélisation Objet, Modèle Vue Contrôleur, Design Pattern Observer, Swing, etc.

Travail à réaliser

Développer une application graphique java **objet et modulaire** la plus aboutie possible (utilisant Swing) pour le jeu d'échecs, respectant le modèle MVC.

(1) Implémenter les fonctionnalités de base permettant de jouer aux échecs en respectant le modèle MVC et les consignes de modélisation orientée objet.

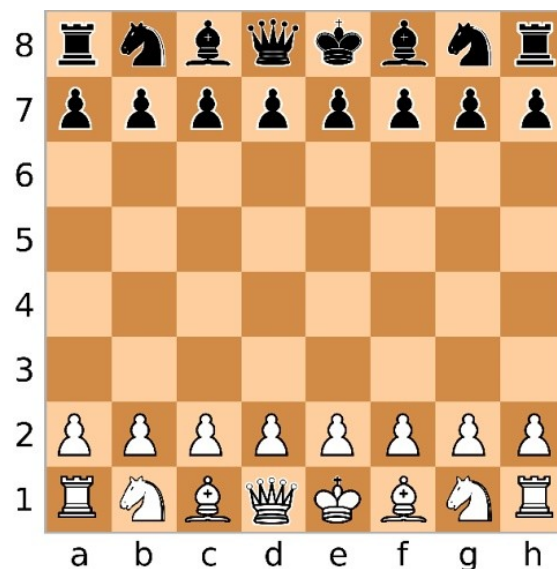
(2) Une fois l'application fonctionnelle, développer des extensions, et réaliser l'application la plus aboutie possible.

Travail en binômes

- Travail personnel entre les séances.
- Évaluations individuelles.
- Rapport par binôme : 6 pages au maximum, listes de fonctionnalités et extensions (indiquer la proportion de temps associée à chacune d'elles), copies d'écran, 1 diagramme UML de votre choix.
- Démonstration lors de la dernière séance encadrée (présence des deux binômes obligatoire).
- Test anti-plagiat sur les code soumis.

Application de base : Jeu d'échecs

À partir du code qui vous est fourni, compléter et développer votre modèle MVC permettant à deux joueurs de se confronter au jeu d'échecs.



Traitements centraux à développer dans le modèle (modélisation étudiée en CM) :

- Calcul de l'état d'échec d'une position [échec | pas échec] pour un joueur.
- Déplacement des pièces : calcul des déplacements physiquement possibles + validation des positions de jeu (le joueur n'est pas en échec à la fin du tour de jeu) + prise des pièces. Cas particuliers : prise en passant, roque, promotion.
- Calcul des fins de parties par Échec et mat ou Pat (les autres conditions de fin de partie ne sont pas à développer dans l'application de base : répétition, série de coups sans prise, situation d'égalité suivant les pièces restantes).
- Fonction de jeu d'une partie.

Traitement à développer dans la vue et le contrôleur :

- Affichage du plateau et des pièces.
(images de pièces disponibles : <https://github.com/lichess-org/lila/tree/master/public/piece>)
- Récupération des interactions utilisateur pour déplacer les pièces.

Extensions

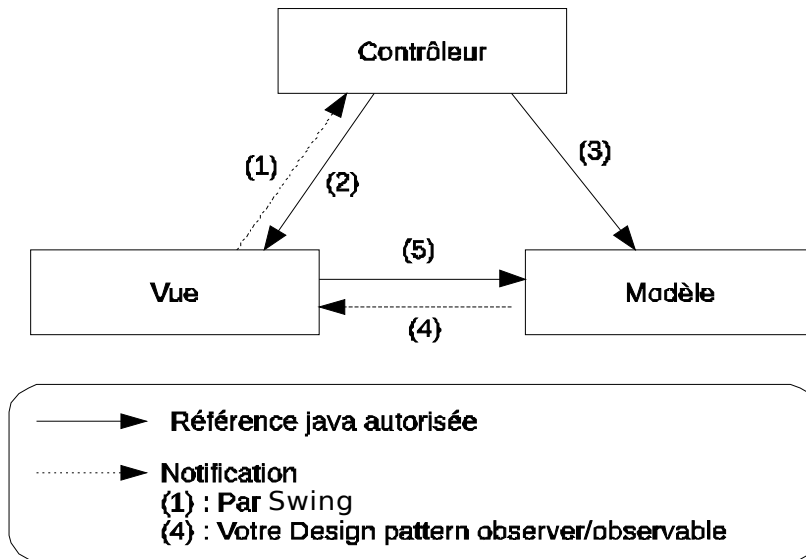
Modélisation :

- Vue console : connecter votre modèle à une vue console, en conservant la vue graphique et sans modifier la partie modèle de votre programme.
- Jeu de damier : choisissez deux jeux qui se jouent sur un damier (échecs, dames, go, variantes du jeu d'échecs, etc.). Pensez votre application de façon générique (traitements communs / traitements spécifiques).

Autres :

- Calcul des fins de partie par répétition. Pour cela utiliser une hashmap, avec pour clef la position de jeu...
- Enregistrement PNG : respecter le format standard permettant d'enregistrer une partie d'échecs.
- Éditeur PNG : permettre l'édition d'une partie PNG (commentaires, etc.).
- Vue 3D.
- Connecteur réseau à un serveur d'échecs.
- Développer une IA.
- Une extension de votre choix.

Rappel MVC



MVC Strict :

- (1) Récupération de l'événement Swing par le contrôleur
- (2) Répercussions locale directe sur la vue sans exploitation du modèle
- (3) Déclenchement d'un traitement pour le modèle
- (4) Notification du modèle pour déclencher une mise à jour graphique
- (5) Consultation de la vue pour réaliser la mise à jour Graphique

Application Calculette :

- (1) récupération clic sur bouton de calculette
- (2) construction de l'expression dans la vue (1,2...9, (,))
- (3) déclenchement calcul (=)
- (4) Calcul terminé, notification de la vue
- (5) La vue consulte le résultat et l'affiche