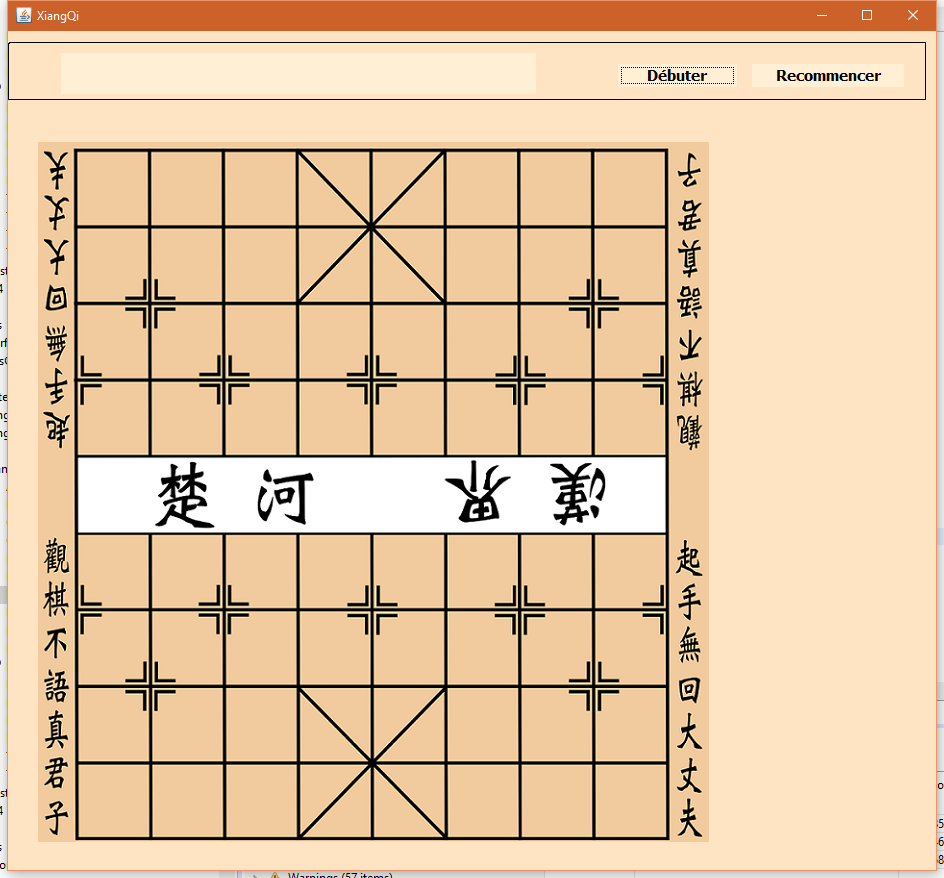
**TRAVAIL SYNTHESE**

**PARTIE 3 : INTERFACE GRAPHIQUE**

L’interface graphique que nous utiliserons sera appelée FrameXiangQi : elle est réalisée partiellement pour vous. En l’exécutant dans votre projet, vous devriez voir apparaître :

panelControle : contenant les éléments extérieurs à la zone de jeu



panelConteneur : avec un GridLayout permettant le positionnement en grille 10x9 des JLabels

grille[4][4] : un JLabel qui contiendra éventuellement un ImageIcon

Le bouton **Débuter** place le jeu pour un début de partie par les rouges.

Le bouton **Recommencer** place le jeu pour un début de partie par les noirs.

**LIBRAIRIES D’INTERFACE GRAPHIQUE (GUI)**

L’application utilise l’ensemble de librairies Swing de Java (javax.swing) qui permet de créer des interfaces graphiques.

Note : JavaFX est une technologie plus récente qui a pour objectif de remplacer Swing.

Les éléments graphiques principaux utilisés par l’application sont :

* **JFrame** : classe principale et fenêtre principale, conteneur qui contient les autres éléments.
* **JPanel** : zones délimitées de la fenêtre principale (panneaux) qui peuvent être décorées et qui contiennent les différents éléments.
* **JButton** : boutons, ici **Débuter** et **Recommencer**.
* **JLabels** : champs pour afficher du texte. Ici on utilise une grille de JLabels transparents superposés aux intersections de l’échiquier et on s’en sert pour ancrer l’affichage de graphiques représentant les pièces.
* **Listener** : les écouteurs servent à détecter des événements du GUI, dans notre cas des clics de souris. On associe un écouteur à chaque élément avec lequel on veut interagir.

**POINTS IMPORTANTS DE LA CONCEPTION**

L’application traite séparément et en parallèle :

* L’état du jeu (les pièces dans le tableau **Intersection[10][9] jeu**)
* L’affichage graphique (les graphiques de pièces sur l’affichage du jeu **JLabell[10][9] grille**)

Les JLabels ont une méthode setIcon() qui permet l’affichage d’un graphique à l’emplacement du JLabel.

Dans les JFrame, un Ecouteur (listener) est une classe qui permet de détecter et traiter des événements. Dans cette application, c’est le relâchement d’un bouton de souris (la fin d’un clic) qui est traité. On associe un écouteur à chaque bouton et à chaque JLabel de la grille pour détecter les clics de souris sur ces éléments

La méthode **afficherPiecesPourDebut** vous est donnée afficher les pièces sur le jeu dans leur position de départ. Inspectez ce code pour comprendre comment faire afficher une pièce à une intersection donnée.

La méthode **enleverToutesLesPièces** retire toutes les pièces de l’affichage du jeu, ainsi que dans les zones de pièces capturées. Inspectez ce code pour comprendre comment retirer une pièce de l’affichage d’une intersection.

La méthode **recommencer** utilise les deux méthodes précédentes pour préparer l’affichage pour une nouvelle partie ET contient le code pour passer le contrôle du jeu aux noirs ET rafraichir l’affichage suite à des changements.

**SÉQUENCE D’UN DÉPLACEMENT**

Le déplacement d’une pièce peut être décrit avec les étapes suivantes :

1. Lorsqu’on clique sur une pièce, celle-ci est retirée de l’affichage, c’est comme si le joueur avait la pièce en main. Note : vous devez avoir un moyen de garder en mémoire la pièce qui est dans la main du joueur.
2. Ensuite, lorsque le joueur clique sur une autre intersection, les vérifications suivantes doivent être effectuées :

* le déplacement doit être valide
* le déplacement doit être possible
* les rois ne peuvent pas se trouver face à face

Si ces conditions ne sont pas respectées, le déplacement n’a pas lieu et le jeu est en attente que le joueur clique sur une autre intersection.

1. Si ces conditions sont respectées, alors la pièce est déplacée et les actions appropriées doivent être entreprise : poser la pièce au bon endroit (possiblement le même), capturer ou non une pièce, gagner la partie etc.

Si la partie n’est pas terminée, alors la main doit être passée au joueur suivant. La méthode alterne est déjà codée pour vous et est utilisée pour ce faire.

Notes :

* N’oubliez pas de traiter l’affichage ET l’état du jeu dans le tableau d’intersections, les deux doivent être traitées en parallèle.
* Si un joueur sélectionne une pièce et la repose au même endroit, on considère qu’il n’a pas encore joué son coup (changé d’idée) et c’est encore à lui.

**FIN DE PARTIE**

La partie se termine lorsqu’un roi est capturé.

Le jeu devrait émettre un message comme quoi la partie est terminée, ainsi que le joueur vainqueur.

De plus, après une fin de partie, plus aucune pièce ne peut être déplacée sur le jeu tant qu’on ne débute pas une nouvelle partie (débuter ou recommencer)

**A RÉALISER :**

Faites une sauvegarde de votre projet partie 2 pour en conserver l’état au besoin.

Copiez le fichier FrameXiangQi.java avec les autres fichiers java de votre application.

Copiez le répertoire icones dans la racine de votre application (au même endroit que le répertoire src).

Coder l’application : tout le code qui vous reste à faire se trouve dans la méthode mouseReleased, à l’intérieur de la logique déjà en place.

**Notes :**

Les champs suivants sont disponibles :

**pieceTampon** : pour garder en mémoire une pièce que le joueur prend dans sa main.

**iconeTampon** : pour garde en mémoire le graphique d’une pièce que le joueur prend dans sa main

La syntaxe pour récupérer le graphique d’une pièce à une intersection;

iconeTampon = (ImageIcon)grille[*ligne*][*colonne*].getIcon()

La syntaxe pour assigner un graphique à une intersection :

grille[*ligne*][*colonne*].setIcon(*nom\_de\_graphique*);

La syntaxe pour ajouter une pièce capturée dans sa zone désignée :

panelNoirs.add(new JLabel (*nom\_du\_graphique*);

panelRouges.add(new JLabel (*nom\_du\_graphique*);

**PONDÉRATION**

Dans la grille d’évaluation, la note de la partie 3 est intégrée à la note de la partie 2, soit 1/3 de la partie 2.

**REMISE**: remettre le projet dans Léa, section Travaux, sous Projet Final – Partie 3, au même endroit où vous avez récupéré cet énoncé.

* L’ensemble de votre projet ( chemins relatifs pour les icônes SVP ), dans un fichier compressé (zip).