

Programme java Echiquier

Règles de déplacement :

La première chose à gérer dans ce TP est d'intégrer à chacune des pièces les contraintes de déplacement. Pour ce faire, nous avons utilisé le principe de l'héritage pour coder les pièces de l'échiquier. En effet, toutes les classes représentant les pièces héritent d'une seule et même classe abstraite nommée *Piece*.

Cette classe *Piece* comprend :

- Les attributs :
 - Un booléen pour savoir la couleur de la pièce
 - Une référence vers l'échiquier
 - Une référence vers la case sur laquelle est la pièce
 - Une image pour l'affichage
- Une méthode abstraite :
 - `listAllDestination()` qui doit renvoyer la liste de toutes les cases sur lesquelles la pièce doit pouvoir se déplacer
- Les méthodes publiques :
 - `void moveTo(Case)` qui déplace la pièce sur la case passée en paramètre si le coup est valide puis change le tour
 - `boolean isCoupOK(Case)` qui vérifie par rapport à la case destination si le coup est valide
 - `void play(Case)` qui déplace la pièce sur la case destination
 - `void cancel(Case, Case, Piece)` qui annule le déplacement de la pièce jouée
 - `Image toImage()` qui renvoie l'image correspondante à la pièce

L'intérêt de cette méthode est que chaque classe fille de *Piece* a donc juste à implanter la méthode `listAllDestination()`. On peut donc définir pour chaque type de pièce la façon dont elle se déplace. Le reste est géré par la classe *Piece*.

Mise en échec et échec & mat

Une fois le déplacement de toutes les pièces codé, il manquait la gestion de la mise en échec du roi. Pour ce faire, nous avons agrandi la classe Echiquier avec deux méthodes :

- `echec()` qui retourne vrai si le roi courant est en échec. Pour cela on vérifie pour chaque déplacement du joueur adverse si une de ses pièces peut prendre le roi. A partir du moment nous avons introduit la notion d'échec, nous avons rajouté ce fait dans la validité du déplacement d'une pièce. En effet une pièce ne peut pas se déplacer si elle met ou laisse son roi en échec.
- `echecEtMat()` qui retourne vrai si le roi courant est échec & mat. Pour cela, on vérifie s'il n'y a plus aucun déplacement valide pour sauver le roi.