

Sur les cases de l'échiquier on peut mettre une pièce... ou pas. Donc pour des raisons pratiques, on mettra toujours un objet-pièce sur chaque case de l'échiquier mais cet objet pourra être vide :

```
self.cases = [  
  
    Piece('TOUR','noir'),Piece('CAVALIER','noir'),Piece('FOU','noir'),Piece('DAME','noir'),Piece('ROI','noir'),  
    Piece('FOU','noir'),Piece('CAVALIER','noir'),Piece('TOUR','noir'),  
  
    Piece('PION','noir'),Piece('PION','noir'),Piece('PION','noir'),Piece('PION','noir'),Piece('PION','noir'),  
    Piece('PION','noir'),Piece('PION','noir'),Piece('PION','noir'),  
  
        Piece(),Piece(),Piece(),Piece(),Piece(),Piece(),Piece(),Piece(),  
        Piece(),Piece(),Piece(),Piece(),Piece(),Piece(),Piece(),Piece(),  
        Piece(),Piece(),Piece(),Piece(),Piece(),Piece(),Piece(),Piece(),  
        Piece(),Piece(),Piece(),Piece(),Piece(),Piece(),Piece(),Piece(),  
  
    Piece('PION','blanc'),Piece('PION','blanc'),Piece('PION','blanc'),Piece('PION','blanc'),Piece('PION','blanc'),  
    Piece('PION','blanc'),Piece('PION','blanc'),Piece('PION','blanc'),  
  
    Piece('TOUR','blanc'),Piece('CAVALIER','blanc'),Piece('FOU','blanc'),Piece('DAME','blanc'),Piece('ROI','blanc'),  
    Piece('FOU','blanc'),Piece('CAVALIER','blanc'),Piece('TOUR','blanc')  
]
```

Outre ses cases, l'échiquier possède les attributs suivants :

- le camp qui a le trait (aux noirs ou blancs de jouer);
- la case d'une prise en passant;
- le nombre de coups joués depuis le début de la partie;
- un historique des coups joués;
- les droits au roque.

On va créer dans cet objet 'Echiquier' les méthodes suivantes :

- générer la liste de tous les coups pour un camp donné;
- possibilité de définir une position des pièces au format FEN;
- possibilité d'exporter cette position avec la notation FEN;
- déplacer une pièce selon les règles du jeu;
- annuler le dernier coup joué;
- changer le joueur qui a le trait;
- savoir si un roi est en échec;
- savoir si une pièce est attaquée;
- afficher l'échiquier en mode console;
- afficher l'historique des coups joués;
- donner une note à une position.

```
nomPiece=(VIDE,'ROI','DAME','TOUR','CAVALIER','FOU','PION')  
valeurPiece=(0,0,9,5,3,3,1)
```

Avec le code ci-dessus, une case vide et le roi ont 0 point, la dame 9, la tour 5, cavalier et fou 3 et enfin le pion : 1 point.

Une pièce a les attributs suivants :

- un nom (roi, dame, tour...);
- une couleur (blanc, noir);
- une valeur (définie en points ci-dessus).

## Déplacement des pièces

Prenons un cas concret :

On a une tour en case a4 (index 32) (voir image plus haut des cases numérotées).

On va la déplacer d'une case à gauche. Elle se retrouve donc en case  $32-1=31$  soit h5 ! impossible !

Nous utiliserons donc la méthode de Robert Hyatt appelée 'mailbox' qui permet d'éviter ce débordement. Il s'agit de deux tableaux de 64 et 120 éléments, ressemblant à une boîte à lettres selon son créateur, d'où son nom :

```
tab120 = (  
    -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,  
    -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,  
    -1, 0, 1, 2, 3, 4, 5, 6, 7, -1,  
    -1, 8, 9, 10, 11, 12, 13, 14, 15, -1,  
    -1, 16, 17, 18, 19, 20, 21, 22, 23, -1,  
    -1, 24, 25, 26, 27, 28, 29, 30, 31, -1,  
    -1, 32, 33, 34, 35, 36, 37, 38, 39, -1,  
    -1, 40, 41, 42, 43, 44, 45, 46, 47, -1,  
    -1, 48, 49, 50, 51, 52, 53, 54, 55, -1,  
    -1, 56, 57, 58, 59, 60, 61, 62, 63, -1,  
    -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,  
    -1, -1, -1, -1, -1, -1, -1, -1, -1, -1  
)  
  
tab64 = (
```

```
21, 22, 23, 24, 25, 26, 27, 28,  
31, 32, 33, 34, 35, 36, 37, 38,  
41, 42, 43, 44, 45, 46, 47, 48,  
51, 52, 53, 54, 55, 56, 57, 58,  
61, 62, 63, 64, 65, 66, 67, 68,  
71, 72, 73, 74, 75, 76, 77, 78,  
81, 82, 83, 84, 85, 86, 87, 88,  
91, 92, 93, 94, 95, 96, 97, 98  
)
```

Avec notre tour en a4 : dans le tableau 'tab64' à l'index 32, nous obtenons la valeur 61.

On la déplace d'une case à gauche, on obtient 60.

Regardons le 60ème élément dans le tableau 'tab120' : cela donne -1.

Il s'agit de la valeur définie pour indiquer une sortie de plateau.

Toujours d'après ce tableau 'tab64' on peut définir les 'vecteurs' suivants pour les déplacements des pièces :

```
deplacements_tour=(-10,10,-1,1)  
deplacements_fou=(-11,-9,11,9)  
deplacements_cavalier=(-12,-21,-19,-8,12,21,19,8)
```

Et cerise sur le gâteau ça suffit ! inutile de spécifier des vecteurs pour la dame et le roi car ceux-ci se déplacent à la fois comme le fou et la tour, mais d'une seule case à la fois pour le roi.

<http://fr.jeffprod.com/blog/2014/comment-programmer-un-jeu-dechecs.html>

## Piece.java

```
import java.util.*;
1
2 public abstract class Piece {
3     public String nom;
4     public int couleur;
5     public Position position;
6
7     public Piece() {
8         nom="Inconnue";
9         couleur=0;
10    }
11
12    public Piece(String nom, int couleur, int x, int y)
13    {
14        this.nom=nom;
15        this.couleur=couleur;
16        this.position = new Position(x,y) ;
17    }
18
19    public String getNom() {
20        return nom;
21    }
22
23    public int getCouleur() {
24        return couleur;
25    }
26
27    public abstract String toString();
28
29    public Position getPosition() {
30        return position;
31    }
}
```

```
public void initialiserPositions() {
```

```
    Fou fou = new Fou("F",1,1,3);
// 1: le nom (string), 2: la couleur (1 noir et 0 blanc), 3: x, 4: y
    echi.getListPiece().add(fou);
    System.out.println(echi);
}
```

Le logiciel se compose de 9 modules

- initialisation
- affichage
- nouveaujeu
- droits
- echecsblancs

- echecsnoirs
- chargement
- sauvegarde
- aide
- option

<http://codes-sources.commentcamarche.net/source/view/30753/975517#browser>