

## Fiche annexe sur le pat.

Une partie d'echec peut se terminer par un match nul, on parle alors de pat.

La détection du pat se réalise après chaque coup joué. La fonction de détection est appelée au niveau de l'API (voir annexe sur la gestion des déplacements).

### Condition de pat standard

Il existe plusieurs conditions de pat, la plus commune étant qu'un des joueur ne peut jouer aucune pièce sans se mettre en echec, alors que son roi n'est actuellement pas en echec.

En supposant que la condition d'echec et mat ait été implémenté, cette condition de pat sera implémentée similairement, la différence provenant du fait que le roi courant n'est pas mis en echec.

### Condition de pat avancée

Deux autres conditions de pat existe aux echec.

- Une partie est en pat si durant les 50 derniers coups, aucune capture de pièce n'a eu lieu et aucun pion n'a été déplacé.
- Une partie est en pat si l'échiquier se retrouve à 3 reprises dans une même configuration.

Ces deux conditions sont plus complexes à vérifier. Vous implémenterez deux fonction spécifiques permettant de vérifier ces deux conditions:

*algorithmique\_partie\_detector\_pat\_50\_coups*

*algorithmique\_partie\_detector\_3\_configurations\_identiques*

Vous écrirez les contrats et les algorithmes de ces fonctions.

### Aide:

Pour le cas des 50 coups, on pourra rejouer le mouvement des pièces à partir de l'historique et vérifier si un pion à été joué ou si une pièce à été capturée.

Pour le cas des 3 configurations identiques, on viendra implémenter une fonction qui permet de comparer 2 échiquiers et indique si ils sont identiques ou différent. Cette fonction pourra alors être utilisées afin de comparer les configurations passées entre elles.

### Note:

Ces conditions peuvent être relativement coûteuses en terme de nombre de parcours et de tests. Dans un premier temps, on pourra coder une version simple parcourant à chaque déplacement l'ensemble de l'historique.

Il sera possible de réfléchir par la suite à la mise au points de sauvegarde d'éléments intermédiaires permettant de limiter la complexité algorithmique.