

Fiche annexe sur l'echec et mat.

Une partie d'echec se gagne lorsque le roi adverse en mise en echec par une pièce, et qu'il n'existe aucun mouvement lui permettant d'éviter d'être placé en echec.

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

Pour détecter si une pièce est en echec et mat, il est nécessaire de savoir si, pour un jeu donné, il est possible ou non d'effectuer au moins un déplacement empêchant la mise en echec.

Pour cela, il est demandé d'implémenter la fonction

algorithmique_partie_detector_deplacement_est_possible dans le fichier *algorithmique_partie*.

Le contrat de cette fonction sera le suivant:

```
/**
 * Fonction detecter_deplacement_est_possible:
 * *****
 *   Detecte si il existe au moins un deplacement possible pour le joueur
courant.
 *   Tous les deplacements sont prises en comptes et ne doivent pas mettre le
joueur
 *   courant en echec.
 *
 *   Necessite:
 *   - Un pointeur constant vers un echiquier (pointeur non NULL).
 *   - Un pointeur constant vers un historique de la partie (pointeur non
NULL).
 *   Garantie:
 *   - Renvoie 1 si au moins un deplacement est possible: il n'y a pas echec
et mat.
 *   - Renvoie 0 si aucun deplacement n'est possible: si echec et mat, soit
pat.
 */
```

L'algorithme de cette fonction sera le suivant:

```
// Pour toutes les pièces du jeu courant
// Pour toutes les cases (x,y) de déplacement potentiel
//   Faire une copie de l'échiquier
//   Tenter de déplacer la pièce en (x,y) (avec prise en compte des coups
speciaux)
//       Si le déplacement s'est réalisé
//           Renvoie 1
//
//renvoie 0
```

Une fois cette fonction réalisée, on implémentera la fonction

algorithmique_partie_detector_echec_et_mat dans le fichier *algorithmique_partie*.

Le contrat et la signature de cette fonction seront les suivants:

```
/**
 * Fonction detecter_deplacement_est_possible:
```

```
* *****
*   Detecte si il existe au moins un deplacement possible pour le joueur
courant.
*   Tous les deplacements sont prises en comptes et ne doivent pas mettre le
joueur
*   courant en echec.
*
*   Necessite:
*   - Un pointeur constant vers un echiquier (pointeur non NULL).
*   - Un pointeur constant vers un historique de la partie (pointeur non
NULL).
*   Garantie:
*   - Renvoie 1 si au moins un deplacement est possible: il n'y a pas echec
et mat.
*   - Renvoie 0 si aucun deplacement n'est possible: si echec et mat, soit
pat.
*/
int algorithmique_partie_detector_deplacement_est_possible(const echiquier*
echiquier_courant,const historique_partie* historique);
```