

Solitaire

Algorithme de déplacement d'un pion

Fonction

Déplacement d'un pion.

Entrées

- Damier avant le déplacement, passé en référence. Pour les besoins de l'exercice, on considèrera que ses cases appartiennent à un repère orthonormé donc les axes partent du point d'origine $(0, 0)$ vers des valeurs positives en abscisse et en ordonnée ;
- Coordonnées de la case de départ ($A_départ, O_départ$) ;
- Coordonnées de la case d'arrivée ($A_arrivée, O_arrivée$).

Sorties

- Damier actualisé (récupération de la référence) ;
- Statut du déplacement :
 - 0 en cas de réussite ;
 - -1 s'il y a un problème avec la case de départ ;
 - -2 s'il y a un problème avec la case d'arrivée ;
 - -3 si la distance entre les deux cases n'est pas égale à 2 ;
 - -4 si la case survolée n'existe pas.

Pré-conditions

- Le mouvement est possible (la distance entre les deux cases est bien de 2) ;
- La case de départ existe et n'est pas vide ;
- La case d'arrivée existe et est vide ;
- La case survolée existe.

Post-conditions

Le mouvement a bien eu lieu (le pion sur la case départ a bien été déplacé vers la case d'arrivée).

Algorithme

```
// Tests sur la validité des cases de départ et d'arrivée
Si non(case_existe(A_départ, O_départ)) OU non(pion_présent(A_arrivée, O_arrivée))
    Retourne -1
FinSi

Si non(case_existe(A_arrivée, O_arrivée)) OU pion_présent(A_arrivée, O_arrivée)
    Retourne -2
FinSi
```

```
// Test de la distance entre les cases de départ et d'arrivée
Diff_A = max(A_départ, A_arrivée) - min(A_départ, A_arrivée)
Diff_O = max(O_départ, O_arrivée) - min(O_départ, O_arrivée)

Si non(Diff_A == 2) ET non(Diff_O == 2)
    Retourne -3
FinSi

// Détermination des coordonnées de la case survolée et test de son existence
A_survolée = max(A_départ, A_arrivée) -1
O_survolée = max(O_départ, O_arrivée) -1

Si non(case_existe(A_survolée, O_survolée))
    Retourne -4
FinSi

// Suppression du pion survolé s'il existe
Si pion_présent(A_survolée, O_survolée)
    pion_retire(A_survolée, O_survolée)
FinSi

// Suppression du pion de départ et ajout de celui d'arrivée
pion_retire(A_départ, O_départ)
pion_ajoute(A_arrivée, O_arrivée)

Retourne 0
```