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\_survolée, O\_survolée)

pion\_ajoute(A\_arrivée, O\_arrivée)

Retourne 0