MoteurJeu

// const

**int** getMaxPasJeu();

**int** getPasJeuCourant();

// [invariants]

// 0 <= getPasjeuCourrant(M) <= getMaxPasJeu(M)

// \pre init(m) require m>=0

// \post getMaxPasJeu(init(m)) = m

// \post getPasJeuCourrant(init(m)) = 0;

**void** init(**int** m);

// \pre pasJeu(M, cmd) require getPasJeuCourant(M) <= getMaxPasJeu(M)

// \post getPasjeuCourrant(pasJeu(M, cmd)) = getPasjeuCourrant(M)+1

**void** pasJeu(Commande cmd);

Classe de Test – AbstractTest

PRECONDITIONS

1. Objectif 1 : init
   1. cas positif

condition initiale : aucune

opération : L1.1=init(5)

oracle : pas de message d’erreur

* 1. cas négatif

condition initiale : aucune

opération : L1.1=init(-1)

oracle : nombre de pas de jeu négatif

// on ne peut pas tester pasJeu

POST-CONDITION

1. Objectif 2 : init

Condition initiale : aucune

opération : L2.1 = init(5)

oracle : invariants vrais et getMaxPasJeu(L2.1) = 5 et getPasJeuCourrant(L2.1) = 0

1. Objectif 3 : pasJeu

Condition initiale : I3.1 = init(5)

Opération : L3.1 = pasJeu(init(5), COMMANDE.RIEN)

Oracle : invariants vrais et getPasJeuCourrant(L3.1) = getPasJeuCourrant(I3.1) + 1

INVARIANTS

Objectif 4 : getPasJeuCourrant

Condition initiale : I4.1 = pasjeu(init(5), COMMANDE.RIEN)

Opération : L4.1 = pasjeu(I4.1, COMMANDE.RIEN)

Oracle : invariants vrais