Service: Joueur

use: Level, GameEng, Lemming
type: Lemming, int, boolean, string

Observateurs:

getNbJetons: [Joueur] x string → int
        pre getNbJetons(s) require s = "DIGGER" ∨ s= "CLIMBER" ∨ s = "BUILDER"
                          ∨ s = "FLOATER" ∨ s = BOMBER" ∨ s = "STOPPER"
                          ∨ s = "BASHER" ∨ s = "MINER"

Constructeurs:

init: int →[Joueur]
        pre init(n) require n > 0

Operateurs:

select: [Joueur] x int x int → Lemming
        pre select(J, x, y) require ¬GameEng::gameOver(getGameEng(J))
                        ∧ x ∈ [0;
Level::getWidth(GameEng::getLevel(getGameEng(J))[
                        ∧ y ∈ [0;
Level::getHeight(GameEng::getLevel(getGameEng(J))[

changeClasse: [Joueur] x Lemming x string →[Joueur]
        pre changeClasse(J, L, s) require s = "DIGGER" ∨ s= "CLIMBER" ∨ s = "BUILDER"
                        ∨ s = "FLOATER" ∨ s = BOMBER" ∨ s = "STOPPER"
                        ∨ s = "BASHER" ∨ s = "MINER"
                        ∧ getNbJetons(J, s) > 0

changeSpawnSpeed: [Joueur] x int →[Joueur]
        pre changeSpawnSpeed(J, s) require s > 0
                                  ∧ GameEng::gameOver(getGameEng(J)) = false

annihilation: [Joueur] →[Joueur]
        pre annihilation(J) require ¬GameEng::gameOver(getGameEng(J))

Observations

[Invariants]
        getNbJetons("DIGGER") ≥ 0
        getNbJetons("CLIMBER") ≥ 0
        getNbJetons("BUILDER") ≥ 0
        getNbJetons("FLOATER") ≥ 0
        getNbJetons("BOMBER") ≥ 0
        getNbJetons("STOPPER") ≥ 0

getNbJetons("BASHER") ≥ 0
getNbJetons("MINER") ≥ 0

[init]

getNbJetons(init(n), "DIGGER") = n
getNbJetons(init(n), "CLIMBER") = n
getNbJetons(init(n), "BUILDER") = n
getNbJetons(init(n), "FLOATER") = n
getNbJetons(init(n), "BOMBER") = n
getNbJetons(init(n), "STOPPER") = n
getNbJetons(init(n), "BASHER") = n
getNbJetons(init(n), "MINER") = n

[changeClasse]

getNbJetons(changeClasse(J, L, s), s) = getNbJetons(J, s)-1
(s = "CLIMBER" ⇒ Lemming::isGrimpeur(L)
s = "FLOATER" ⇒ Lemming::isFlotteur(L)
s = "BUILDER" ⇒ Lemming::isBuilder(L)
s = "BOMBER" ⇒ Lemming::isExploseur(L)) v
(s = "DIGGER" ⇒ Lemming::getType(L) = CREUSEUR
s = "STOPPER" ⇒ Lemming::getType(L) = STOPPEUR
s = "BASHER" ⇒ Lemming::getType(L) = BASHER
s = "MINER" ⇒ Lemming::getType(L) = MINER)

[spawnSpeed]

GameEng::getSpawnSpeed(getGameEng(spawnSpeed(J, s))) = s

[annihilation]

∀ i ∈ GameEng::getLemmingsActifs(getGameEng(annihilation(J)),
    Lemming::isExploseur(GameEng::getLemming(getGameEng(annihilation(J)),
i))