

## Service : Lemming ajouter require gameEng

**Use:** GameEng, Level

**type:** bool, int, enum Direction {DROITIER, GAUCHER}

**enum:** Type {MARCHEUR, TOMBEUR, CREUSEUR, BASHER, MINER, STOPPEUR}

### Observers :

getDirection : [Lemming] → Direction  
getType : [Lemming] → Type  
getX : [Lemming] → int  
getY : [Lemming] → int  
tombeDepuis : [Lemming] → int  
    **pre** tombeDepuis(L) require getType(L) = TOMBEUR  
exploseurDepuis : [Lemming] → int  
    **pre** exploseurDepuis(L) require isExploseur(L)  
getId : [Lemming] → int  
isGrimpeur : [Lemming] → bool  
isFlotteur : [Lemming] → bool  
isBuilder : [Lemming] → bool  
isExploseur : [Lemming] → bool  
getGameEng : [Lemming] → GameEng  
isCurrentlyBuilding : [Lemming] → bool  
    **pre** isCurrentlyBuilding(L) require isBuilder(L)  
isCurrentlyClimbing : [Lemming] → bool  
    **pre** isCurrentlyClimbing(L) require isGrimpeur(L)  
getNombreToursBuilder : [Lemming] → int  
    **pre** getNombreToursBuilder(L) require isBuilder(L)  
getNombresDallesPosees : [Lemming] → int  
    **pre** getNombresDallesPosees(L) require isBuilder(L)  
nbCreuseTunnel : [Lemming] → int  
    **pre** nbCreuseTunnel(L) require getType(L) = BASHER

### Constructors:

init : int → [Lemming]

### Operators :

devientTombeur : [Lemming] → [Lemming]  
    **pre** devientTombeur(L) require getType(L) ≠ TOMBEUR  
devientMarcheur : [Lemming] → [Lemming]  
    **pre** devientMarcheur(L) require getType(L) ≠ MARCHEUR  
devientGaucher : [Lemming] → [Lemming]  
    **pre** devientGaucher(L) require getDirection(L) ≠ GAUCHER  
devientDroitier : [Lemming] → [Lemming]  
    **pre** devientDroitier(L) require getDirection(L) ≠ DROITIER

```

devientCreuseur : [Lemming] → [Lemming]
    pre devientCreuseur(L) require getType(L) ≠ CREUSEUR
devientGrimpeur : [Lemming] → [Lemming]
    pre devientGrimpeur(L) require !isGrimpeur(L)
        ∧ ( getType(L) = MARCHEUR
            ∨ getType(L) = MARCHEUR
            ∨ getType(L) = MARCHEUR )
devientFlotteur : [Lemming] → [Lemming]
    pre devientFlotteur(L) require !isFlotteur(L)
devientBuilder : [Lemming] → [Lemming]
    pre devientBuilder(L) require !isBuilder(L)
devientMiner : [Lemming] → [Lemming]
    pre devientMiner(L) require getType(L) ≠ MINER
devientBasher : [Lemming] → [Lemming]
    pre devientBasher(L) require getType(L) ≠ BASHER
devientExploseur : [Lemming] → [Lemming]
    pre devientExploseur(L) require !isExploseur(L)
devientStoppeur : [Lemming] → [Lemming]
    pre devientStoppeur(L) require getType(L) ≠ TOMBEUR
        ∧ getType(L) ≠ STOPPEUR

const intervalle_dalle: [Lemming] -> int
const maxDalles : [Lemming] -> int
const maxCreuse: [Lemming] -> int
step : [Lemming] → [Lemming]
    pre step(L) require !GameEng::gameOver(getGameEng(L))
        ∧ getId(L) ∈ GameEng::getLemmingActifs(getGameEng(L))

```

## Observations :

[invariants]

```

0 ≤ getX(L) < Level::getWidth(GameEng::getLevel(getGameEng(L)))
0 ≤ getY(L) < Level::getHeight(GameEng::getLevel(getGameEng(L)))
getId(L) > 0

```

[init]

```

getDirection(init(id)) = DROITIER
getType(init(id)) = TOMBEUR
getId(init(id)) = id
isGrimpeur(init(id)) = false
isExploseur(init(id)) = false
isExploseur(init(id)) = false
isFlotteur(init(id)) = false
isBuilder(init(id)) = false
tombeDepuis(init(id)) = 0
isCurrentlyBuilding(init(id)) = false
isCurrentlyClimbing(init(id)) = false
nbCreuseTunnel(init(id)) = 0

```

```

    getNombreToursBuilder(init(id)) = 0
    getNombreDallesPosees(init(id)) = 0
[devientTombreur]
    getType(devientTombreur(L)) = TOMBEUR

[devientMarcheur]
    getType(devientMarcheur(L)) = MARCHEUR
    tombeDepuis(devientMarcheur(L)) = 0
[devientGaucher]
    getDirection(devientGaucher(L)) = GAUCHER
[devientDroitier]
    getDirection(devientDroitier(L)) = DROITIER
[devientCreuseur]
    getType(devientCreuseur(L)) = CREUSEUR
[devientGrimpeur]
    isGrimpeur(devientGrimpeur(L)) = true
[devientFlotteur]
    isFlotteur(devientFlotteur(L)) = true
[devientBuilder]
    isBuilder(devientBuilder(L)) = true
[devientMiner]
    getType(devientMiner(L)) = MINER
[devientBasher]
    getType(devientBasher(L)) = BASHER
    nbCreuseTunnel(devientBasher(L)) = 0
[devientExploseur]
    isExploseur(devientExploseur(L)) = true
[devientStoppeur]
    getType(devientStoppeur(L)) = STOPPEUR
[intervalle_dalle] :
    intervalle_dalle(L) = 2
[maxDalles]
    maxDalles(L) = 12
[maxCreuse]
    maxCreuse(L) = 20
[step]
    • isExploseur(L) ∧ exploseurDepuis(L) < 4 ⇒ exploseurDepuis(step(L)) = exploseurDepuis(L) + 1
    • isExploseur(L) ∧ exploseurDepuis(L) = 4 ⇒
        ∧ GameEng::getNombreActifs(getGameEng(step(L))) = GameEng::getNombreActifs(getGameEng(L)) -1
        ∧ GameEng::getNombreMorts(getGameEng(step(L))) = GameEng::getNombreMorts(getGameEng(L))+1
        ∧ ∀ i ∈ [getX(L)-2, getX(L)+2], ∀ j ∈ [getY(L)-2, getY(L)+2]
        i ≥ 0 ∧ i ≤ Level::getWidth(GameEng::getLevel(getGameEng(L)))
        j ≥ 0 ∧ j ≤ Level::getHeight(GameEng::getLevel(getGameEng(L)))
        (i ≠ getX(L)-2 ∧ j ≠ getY(L)-2)
        (i ≠ getX(L)-2 ∧ j ≠ getY(L)+1)
        (i ≠ getX(L)+2 ∧ j ≠ getY(L)-2)
        (i ≠ getX(L)+2 ∧ j ≠ getY(L)+2)

```

Level::getNature(GameEng::getLevel(getGameEng(L), i, j) = DIRT  $\Rightarrow$   
 Level::getNature(GameEng::getLevel(getGameEng(step(L)), i, j) = EMPTY

- $\text{getType}(L) = \text{BASHER} \wedge \neg \text{GameEng::isObstacle}(\text{getX}(L), \text{getY}(L)) \Rightarrow$   
 $\text{getType}(\text{step}(L)) = \text{TOMBEUR} \wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L) \wedge \text{getY}(\text{step}(L)) = \text{getY}(L)$
- $\text{getType}(L) = \text{BASHER} \wedge \text{getDirection}(L) = \text{DROITIER}$   
 $\wedge \text{GameEng::isObstacle}(\text{getGameEng}(L), \text{getX}(L), \text{getY}(L)+1)$   
 $\wedge \neg \text{GameEng::isObstacle}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L))$   
 $\wedge \neg \text{GameEng::isObstacle}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L)-1)$   
 $\wedge \neg \text{GameEng::isObstacle}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L)-2)$   
 $\wedge \text{Level::getNature}(\text{GameEng::getLevel}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L)) \neq \text{METAL}$   
 $\wedge \text{Level::getNature}(\text{GameEng::getLevel}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L)-1) \neq \text{METAL}$   
 $\wedge \text{Level::getNature}(\text{GameEng::getLevel}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L)-2) \neq \text{METAL}$   
 $\wedge \text{nbCreuseTunnel}(L) < \text{maxCreuse}(L)$   
 $\Rightarrow \text{getType}(\text{step}(L)) = \text{getType}(L)$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L) + 1$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L)$   
 $\wedge \neg \text{GameEng::isObstacle}(\text{getGameEng}(\text{step}(L)), \text{getX}(L)+1, \text{getY}(L))$   
 $\wedge \neg \text{GameEng::isObstacle}(\text{getGameEng}(\text{step}(L)), \text{getX}(L)+1, \text{getY}(L)-1)$   
 $\wedge \neg \text{GameEng::isObstacle}(\text{getGameEng}(\text{step}(L)), \text{getX}(L)+1, \text{getY}(L)-2)$
- $\text{getType}(L) = \text{BASHER} \wedge \text{getDirection}(L) = \text{GAUCHER}$   
 $\wedge \text{GameEng::isObstacle}(\text{getGameEng}(L), \text{getX}(L), \text{getY}(L)+1)$   
 $\wedge \neg \text{GameEng::isObstacle}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L))$   
 $\wedge \neg \text{GameEng::isObstacle}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L)-1)$   
 $\wedge \neg \text{GameEng::isObstacle}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L)-2)$   
 $\wedge \text{Level::getNature}(\text{GameEng::getLevel}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L)) \neq \text{METAL}$   
 $\wedge \text{Level::getNature}(\text{GameEng::getLevel}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L)-1) \neq \text{METAL}$   
 $\wedge \text{Level::getNature}(\text{GameEng::getLevel}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L)-2) \neq \text{METAL}$   
 $\wedge \text{nbCreuseTunnel}(L) < \text{maxCreuse}(L)$   
 $\Rightarrow \text{getType}(\text{step}(L)) = \text{getType}(L)$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L) - 1$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L)$   
 $\wedge \neg \text{GameEng::isObstacle}(\text{getGameEng}(\text{step}(L)), \text{getX}(L)-1, \text{getY}(L))$   
 $\wedge \neg \text{GameEng::isObstacle}(\text{getGameEng}(\text{step}(L)), \text{getX}(L)-1, \text{getY}(L)-1)$   
 $\wedge \neg \text{GameEng::isObstacle}(\text{getGameEng}(\text{step}(L)), \text{getX}(L)-1, \text{getY}(L)-2)$
- $\text{getType}(L) = \text{BASHER} \wedge \text{getDirection}(L) = \text{DROITIER}$   
 $\wedge \text{GameEng::isObstacle}(\text{getGameEng}(L), \text{getX}(L), \text{getY}(L)+1)$   
 $\wedge (\text{GameEng::isObstacle}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L))$   
 $\vee \text{GameEng::isObstacle}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L)-1)$   
 $\vee \text{GameEng::isObstacle}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L)-2)$   
 $\vee \text{Level::getNature}(\text{GameEng::getLevel}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L)) = \text{METAL}$   
 $\vee \text{Level::getNature}(\text{GameEng::getLevel}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L)-1) = \text{METAL}$   
 $\vee \text{Level::getNature}(\text{GameEng::getLevel}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L)-2) = \text{METAL}$   
 $\vee \text{nbCreuseTunnel}(L) \geq \text{maxCreuse}(L))$   
 $\Rightarrow \text{getType}(\text{step}(L)) = \text{MARCHEUR}$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L)$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L)$

- $\text{getType}(L) = \text{BASHER} \wedge \text{getDirection}(L) = \text{GAUCHER}$   
 $\wedge \text{GameEng}::\text{isObstacle}(\text{getGameEng}(L), \text{getX}(L), \text{getY}(L)+1)$   
 $\wedge (\text{GameEng}::\text{isObstacle}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L))$   
 $\quad \vee \text{GameEng}::\text{isObstacle}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L)-1)$   
 $\quad \vee \text{GameEng}::\text{isObstacle}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L)-2)$   
 $\quad \vee \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L)) = \text{METAL}$   
 $\quad \vee \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L)-1) = \text{METAL}$   
 $\quad \vee \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L)-2) = \text{METAL}$   
 $\quad \vee \text{nbCreuseTunnel}(L) \geq \text{maxCreuse}(L))$   
 $\Rightarrow \text{getType}(\text{step}(L)) = \text{MARCHEUR}$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L)$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L)$
- $\text{getType}(L) = \text{MINER} \wedge \text{!GameEng}::\text{isObstacle}(\text{getX}(L), \text{getY}(L)) \Rightarrow$   
 $\quad \text{getType}(\text{step}(L)) = \text{TOMBEUR} \wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\quad \wedge \text{getX}(\text{step}(L)) = \text{getX}(L) \wedge \text{getY}(\text{step}(L)) = \text{getY}(L)$
- $\text{getType}(L) = \text{MINER} \wedge \text{getDirection}(L) = \text{DROITIER}$   
 $\wedge \text{GameEng}::\text{isObstacle}(\text{getGameEng}(L), \text{getX}(L), \text{getY}(L)+1)$   
 $\wedge \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L)) \neq \text{METAL}$   
 $\wedge \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)+2, \text{getY}(L)) \neq \text{METAL}$   
 $\wedge \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L)+1) \neq \text{METAL}$   
 $\wedge \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)+2, \text{getY}(L)+1) \neq \text{METAL}$   
 $\wedge \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)+3, \text{getY}(L)+1) \neq \text{METAL}$   
 $\Rightarrow \text{getType}(\text{step}(L)) = \text{getType}(L)$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L) + 1$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L) + 1$   
 $\wedge \text{!GameEng}::\text{isObstacle}(\text{getGameEng}(\text{step}(L)), \text{getX}(L)+1, \text{getY}(L))$   
 $\wedge \text{!GameEng}::\text{isObstacle}(\text{getGameEng}(\text{step}(L)), \text{getX}(L)+1, \text{getY}(L)-1)$   
 $\wedge \text{!GameEng}::\text{isObstacle}(\text{getGameEng}(\text{step}(L)), \text{getX}(L)+1, \text{getY}(L)-2)$
- $\text{getType}(L) = \text{MINER} \wedge \text{getDirection}(L) = \text{GAUCHER}$   
 $\wedge \text{GameEng}::\text{isObstacle}(\text{getGameEng}(L), \text{getX}(L), \text{getY}(L)+1)$   
 $\wedge \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L)) \neq \text{METAL}$   
 $\wedge \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)-2, \text{getY}(L)) \neq \text{METAL}$   
 $\wedge \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L)+1) \neq \text{METAL}$   
 $\wedge \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)-2, \text{getY}(L)+1) \neq \text{METAL}$   
 $\wedge \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)-3, \text{getY}(L)+1) \neq \text{METAL}$   
 $\Rightarrow \text{getType}(\text{step}(L)) = \text{getType}(L)$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L) - 1$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L) + 1$   
 $\wedge \text{!GameEng}::\text{isObstacle}(\text{getGameEng}(\text{step}(L)), \text{getX}(L)-1, \text{getY}(L))$   
 $\wedge \text{!GameEng}::\text{isObstacle}(\text{getGameEng}(\text{step}(L)), \text{getX}(L)-1, \text{getY}(L)-1)$   
 $\wedge \text{!GameEng}::\text{isObstacle}(\text{getGameEng}(\text{step}(L)), \text{getX}(L)-1, \text{getY}(L)-2)$

- [illegible]

- [illegible]

- $\bullet$   $\text{getType}(L) = \text{MARCHEUR} \wedge \text{getDirection}(L) = \text{GAUCHER} \wedge \text{isGrimpeur}(L)$   
 $\wedge \text{GameEng}::\text{isObstacle}(\text{getGameEng}(L), \text{getX}(L), \text{getY}(L)-2)$   
 $\wedge \text{GameEng}::\text{isObstacle}(\text{getGameEng}(L), \text{getX}(L), \text{getY}(L)+1)$   
 $\wedge \text{GameEng}::\text{isObstacle}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L))$   
 $\Rightarrow \text{getType}(\text{step}(L)) = \text{getType}(L)$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{DROITIER}$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L)$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L)$
- $\bullet$   $\text{getType}(L) = \text{MARCHEUR} \wedge \text{getDirection}(L) = \text{DROITIER} \wedge \text{isBuilder}(L) \wedge \text{isCurrentlyBuilding}(L)$   
 $\wedge \text{getNombreToursBuilder}(L) < \text{intervalle\_dalle}(L)$   
 $\Rightarrow \text{getNombreToursBuilder}(\text{step}(L)) = \text{getNombreToursBuilder}(L) + 1$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L)$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L)$   
 $\wedge \text{getType}(\text{step}(L)) = \text{getType}(L)$
- $\bullet$   $\text{getType}(L) = \text{MARCHEUR} \wedge \text{getDirection}(L) = \text{DROITIER} \wedge \text{isBuilder}(L) \wedge \text{isCurrentlyBuilding}(L)$   
 $\wedge \text{getNombreToursBuilder}(L) = \text{intervalle\_dalle}(L)$   
 $\wedge \text{!GameEng}::\text{isObstacle2}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L))$   
 $\wedge \text{!GameEng}::\text{isObstacle2}(\text{getGameEng}(L), \text{getX}(L)+2, \text{getY}(L))$   
 $\wedge \text{!GameEng}::\text{isObstacle2}(\text{getGameEng}(L), \text{getX}(L)+3, \text{getY}(L))$   
 $\wedge \text{GameEng}::\text{isLibre}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L)-2)$   
 $\wedge \text{GameEng}::\text{isLibre}(\text{getGameEng}(L), \text{getX}(L)+2, \text{getY}(L)-2)$   
 $\wedge \text{GameEng}::\text{isLibre}(\text{getGameEng}(L), \text{getX}(L)+3, \text{getY}(L)-2)$   
 $\wedge \text{getNombreDallesPosees}(L) < \text{maxDalles}(L)$   
 $\Rightarrow \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L)) = \text{DIRT}$   
 $\wedge \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)+2, \text{getY}(L)) = \text{DIRT}$   
 $\wedge \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)+3, \text{getY}(L)) = \text{DIRT}$   
 $\wedge \text{getNombreDallesPosees}(\text{step}(L)) < \text{getNombreDallesPosees}(L) + 1$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L) + 2$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L) - 1$   
 $\wedge \text{getNombreToursBuilder}(L) = 0$   
 $\wedge \text{getType}(\text{step}(L)) = \text{getType}(L)$
- $\bullet$   $\text{getType}(L) = \text{MARCHEUR} \wedge \text{getDirection}(L) = \text{GAUCHER} \wedge \text{isBuilder}(L) \wedge \text{isCurrentlyBuilding}(L)$   
 $\wedge \text{getNombreToursBuilder}(L) = \text{intervalle\_dalle}(L)$   
 $\wedge \text{!GameEng}::\text{isObstacle2}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L))$   
 $\wedge \text{!GameEng}::\text{isObstacle2}(\text{getGameEng}(L), \text{getX}(L)-2, \text{getY}(L))$   
 $\wedge \text{!GameEng}::\text{isObstacle2}(\text{getGameEng}(L), \text{getX}(L)-3, \text{getY}(L))$   
 $\wedge \text{GameEng}::\text{isLibre}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L)-2)$   
 $\wedge \text{GameEng}::\text{isLibre}(\text{getGameEng}(L), \text{getX}(L)-2, \text{getY}(L)-2)$   
 $\wedge \text{GameEng}::\text{isLibre}(\text{getGameEng}(L), \text{getX}(L)-3, \text{getY}(L)-2)$   
 $\wedge \text{getNombreDallesPosees}(L) < \text{maxDalles}(L)$   
 $\Rightarrow \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L)) = \text{DIRT}$   
 $\wedge \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)+2, \text{getY}(L)) = \text{DIRT}$   
 $\wedge \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L)+3, \text{getY}(L)) = \text{DIRT}$   
 $\wedge \text{getNombreDallesPosees}(\text{step}(L)) < \text{getNombreDallesPosees}(L) + 1$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L) - 2$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L) - 1$   
 $\wedge \text{getNombreToursBuilder}(L) = 0$   
 $\wedge \text{getType}(\text{step}(L)) = \text{getType}(L)$



- $\bullet$   $\text{getType}(L) = \text{MARCHEUR} \wedge \text{getDirection}(L) = \text{GAUCHER} \wedge \text{isBuilder}(L) \wedge \text{isCurrentlyBuilding}(L)$   
 $\wedge \text{getNombreToursBuilder}(L) = \text{intervalle\_dalle}(L)$   
 $(\text{GameEng}::\text{isObstacle2}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L))$   
 $\vee \text{GameEng}::\text{isObstacle2}(\text{getGameEng}(L), \text{getX}(L)-2, \text{getY}(L))$   
 $\vee \text{GameEng}::\text{isObstacle2}(\text{getGameEng}(L), \text{getX}(L)-3, \text{getY}(L))$   
 $\vee \text{GameEng}::\text{isLibre}(\text{getGameEng}(L), \text{getX}(L)-1, \text{getY}(L)-2)$   
 $\vee \text{GameEng}::\text{isLibre}(\text{getGameEng}(L), \text{getX}(L)-2, \text{getY}(L)-2)$   
 $\vee \text{GameEng}::\text{isLibre}(\text{getGameEng}(L), \text{getX}(L)-3, \text{getY}(L)-2)$   
 $\vee \text{getNombreDallesPosees}(L) \geq \text{maxDalles}(L))$   
 $\Rightarrow \text{getNombreDallesPosees}(\text{step}(L)) = \text{getNombreDallesPosees}(L)$   
 $\wedge \text{getType}(\text{step}(L)) = \text{getType}(L)$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L)$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L)$   
 $\wedge \text{getNombreToursBuilder}(L) = 0$   
 $\wedge \text{isCurrentlyBuilding}(L) = \text{false}$
- $\bullet$   $\text{getType}(L) = \text{MARCHEUR} \wedge \text{getDirection}(L) = \text{DROITIER} \wedge \text{isBuilder}(L) \wedge \text{isCurrentlyBuilding}(L)$   
 $\wedge \text{getNombreToursBuilder}(L) = \text{intervalle\_dalle}(L)$   
 $(\text{GameEng}::\text{isObstacle2}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L))$   
 $\vee \text{GameEng}::\text{isObstacle2}(\text{getGameEng}(L), \text{getX}(L)+2, \text{getY}(L))$   
 $\vee \text{GameEng}::\text{isObstacle2}(\text{getGameEng}(L), \text{getX}(L)+3, \text{getY}(L))$   
 $\vee \text{GameEng}::\text{isLibre}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L)-2)$   
 $\vee \text{GameEng}::\text{isLibre}(\text{getGameEng}(L), \text{getX}(L)+2, \text{getY}(L)-2)$   
 $\vee \text{GameEng}::\text{isLibre}(\text{getGameEng}(L), \text{getX}(L)+3, \text{getY}(L)-2)$   
 $\vee \text{getNombreDallesPosees}(L) \geq \text{maxDalles}(L))$   
 $\Rightarrow \text{getNombreDallesPosees}(\text{step}(L)) = \text{getNombreDallesPosees}(L)$   
 $\wedge \text{getType}(\text{step}(L)) = \text{getType}(L)$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L)$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L)$   
 $\wedge \text{getNombreToursBuilder}(\text{step}(L)) = 0$   
 $\wedge \text{isCurrentlyBuilding}(\text{step}(L)) = \text{false}$
- $\bullet$   $\text{getType}(L) = \text{MARCHEUR} \wedge \text{getDirection}(L) = \text{DROITIER}$   
 $\wedge \text{GameEng}::\text{isObstacle}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L)-1)$   
 $\Rightarrow \wedge \text{getType}(\text{step}(L)) = \text{getType}(L)$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{GAUCHER}$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L)$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L)$
- $\bullet$   $\text{getType}(L) = \text{MARCHEUR} \wedge \text{getDirection}(L) = \text{DROITIER}$   
 $\wedge \text{GameEng}::\text{isObstacle}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L))$   
 $\wedge \text{GameEng}::\text{isObstacle}(\text{getGameEng}(L), \text{getX}(L)+1, \text{getY}(L)-2)$   
 $\Rightarrow \wedge \text{getType}(\text{step}(L)) = \text{getType}(L)$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{GAUCHER}$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L)$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L)$

- $$\begin{aligned} & \bullet \text{ getType(L) = MARCHEUR } \wedge \text{ getDirection(L) = DROITIER} \\ & \quad \wedge \text{ GameEng::isObstacle(getGameEng(L), getX(L)+1, getY(L))} \\ & \quad \wedge \text{ !GameEng::isObstacle(getGameEng(L), getX(L)+1, getY(L)-1)} \\ & \quad \wedge \text{ !GameEng::isObstacle(getGameEng(L), getX(L)+1, getY(L)-2)} \\ & \Rightarrow \wedge \text{ getType(step(L)) = getType(L)} \\ & \quad \wedge \text{ getDirection(step(L)) = getDirection(L)} \\ & \quad \wedge \text{ getX(step(L)) = getX(L) + 1} \\ & \quad \wedge \text{ getY(step(L)) = getY(L) - 1} \end{aligned}$$
- $$\begin{aligned} & \bullet \text{ getType(L) = MARCHEUR } \wedge \text{ getDirection(L) = DROITIER} \\ & \quad \wedge \text{ GameEng::isObstacle(getGameEng(L), getX(L)+1, getY(L))} \\ & \quad \wedge \text{ !GameEng::isObstacle(getGameEng(L), getX(L)+1, getY(L)-1)} \\ & \Rightarrow \wedge \text{ getType(step(L)) = getType(L)} \\ & \quad \wedge \text{ getDirection(step(L)) = getDirection(L)} \\ & \quad \wedge \text{ getX(step(L)) = getX(L) + 1} \\ & \quad \wedge \text{ getY(step(L)) = getY(L)} \end{aligned}$$

===
- $$\begin{aligned} & \bullet \text{ getType(L) = MARCHEUR } \wedge \text{ getDirection(L) = GAUCHER} \\ & \quad \wedge \text{ GameEng::isObstacle(getGameEng(L), getX(L)-1, getY(L)-1)} \\ & \Rightarrow \wedge \text{ getType(step(L)) = getType(L)} \\ & \quad \wedge \text{ getDirection(step(L)) = DROITIER} \\ & \quad \wedge \text{ getX(step(L)) = getX(L)} \\ & \quad \wedge \text{ getY(step(L)) = getY(L)} \end{aligned}$$
- $$\begin{aligned} & \bullet \text{ getType(L) = MARCHEUR } \wedge \text{ getDirection(L) = GAUCHER} \\ & \quad \wedge \text{ GameEng::isObstacle(getGameEng(L), getX(L)-1, getY(L))} \\ & \quad \wedge \text{ GameEng::isObstacle(getGameEng(L), getX(L)-1, getY(L)-2)} \\ & \Rightarrow \wedge \text{ getType(step(L)) = getType(L)} \\ & \quad \wedge \text{ getDirection(step(L)) = DROITIER} \\ & \quad \wedge \text{ getX(step(L)) = getX(L)} \\ & \quad \wedge \text{ getY(step(L)) = getY(L)} \end{aligned}$$
- $$\begin{aligned} & \bullet \text{ getType(L) = MARCHEUR } \wedge \text{ getDirection(L) = GAUCHER} \\ & \quad \wedge \text{ GameEng::isObstacle(getGameEng(L), getX(L)-1, getY(L))} \\ & \quad \wedge \text{ !GameEng::isObstacle(getGameEng(L), getX(L)-1, getY(L)-1)} \\ & \quad \wedge \text{ !GameEng::isObstacle(getGameEng(L), getX(L)-1, getY(L)-2)} \\ & \Rightarrow \wedge \text{ getType(step(L)) = getType(L)} \\ & \quad \wedge \text{ getDirection(step(L)) = getDirection(L)} \\ & \quad \wedge \text{ getX(step(L)) = getX(L) - 1} \\ & \quad \wedge \text{ getY(step(L)) = getY(L) - 1} \end{aligned}$$
- $$\begin{aligned} & \bullet \text{ getType(L) = MARCHEUR } \wedge \text{ getDirection(L) = GAUCHER} \\ & \quad \wedge \text{ GameEng::isObstacle(getGameEng(L), getX(L)-1, getY(L))} \\ & \quad \wedge \text{ !GameEng::isObstacle(getGameEng(L), getX(L)-1, getY(L)-1)} \\ & \Rightarrow \wedge \text{ getType(step(L)) = getType(L)} \\ & \quad \wedge \text{ getDirection(step(L)) = getDirection(L)} \\ & \quad \wedge \text{ getX(step(L)) = getX(L) - 1} \\ & \quad \wedge \text{ getY(step(L)) = getY(L)} \end{aligned}$$

- $\bullet$   $\text{getType}(L) = \text{TOMBEUR} \wedge \text{isFlotteur}(L)$   
 $\wedge \text{!GameEng::isObstacle}(\text{getGameEng}(L), \text{getX}(L), \text{getY}(L)+1)$   
 $\wedge \text{tombeDepuis}(L) \bmod 2 = 0$   
 $\Rightarrow \wedge \text{getType}(\text{step}(L)) = \text{getType}(L)$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\wedge \text{tombeDepuis}(\text{step}(L)) = \text{tombeDepuis}(L) + 1$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L)$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L) + 1$
- $\bullet$   $\text{getType}(L) = \text{TOMBEUR} \wedge \text{isFlotteur}(L)$   
 $\wedge \text{!GameEng::isObstacle}(\text{getGameEng}(L), \text{getX}(L), \text{getY}(L)+1)$   
 $\wedge \text{tombeDepuis}(L) \bmod 2 = 1$   
 $\Rightarrow \wedge \text{getType}(\text{step}(L)) = \text{getType}(L)$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\wedge \text{tombeDepuis}(\text{step}(L)) = \text{tombeDepuis}(L) + 1$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L)$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L)$
- $\bullet$   $\text{getType}(L) = \text{TOMBEUR} \wedge \text{isFlotteur}(L)$   
 $\wedge \text{GameEng::isObstacle}(\text{getGameEng}(L), \text{getX}(L), \text{getY}(L)+1)$   
 $\Rightarrow \wedge \text{getType}(\text{step}(L)) = \text{MARCHEUR}$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L)$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L)$
- $\bullet$   $\text{getType}(L) = \text{TOMBEUR}$   
 $\wedge \text{GameEng::isObstacle}(\text{getGameEng}(L), \text{getX}(L), \text{getY}(L)+1)$   
 $\wedge \text{tombeDepuis}(L) < 8$   
 $\Rightarrow \wedge \text{getType}(\text{step}(L)) = \text{MARCHEUR}$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\wedge \text{tombeDepuis}(\text{step}(L)) = 0$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L)$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L) + 1$
- $\bullet$   $\text{getType}(L) = \text{TOMBEUR}$   
 $\wedge \text{GameEng::isObstacle}(\text{getGameEng}(L), \text{getX}(L), \text{getY}(L)+1)$   
 $\wedge \text{tombeDepuis}(L) \geq 8$   
 $\Rightarrow \text{GameEng::getNombreMorts}(\text{getGameEng}(\text{step}(L))) = \text{GameEng::getNombreMorts}(\text{getGameEng}(L)) + 1$   
 $\wedge \text{getId}(L) \notin \text{GameEng::getLemmingsActifs}(\text{getGameEng}(\text{step}(L)))$
- $\bullet$   $\text{getType}(L) = \text{TOMBEUR}$   
 $\wedge \text{!GameEng::isObstacle}(\text{getGameEng}(L), \text{getX}(L), \text{getY}(L)+1)$   
 $\wedge \text{tombeDepuis}(L) < 8$   
 $\Rightarrow \wedge \text{getType}(\text{step}(L)) = \text{getType}(L)$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\wedge \text{tombeDepuis}(\text{step}(L)) = \text{tombeDepuis}(L) + 1$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L)$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L) + 1$

- [illegible]

- $\text{getType}(L) = \text{CREUSEUR}$   
 $\wedge \text{Level}::\text{getNature}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(L), \text{getX}(L), \text{getY}(L)+1) = \text{DIRT}$   
 $\Rightarrow \wedge \text{getType}(\text{step}(L)) = \text{getType}(L)$   
 $\wedge \text{getDirection}(\text{step}(L)) = \text{getDirection}(L)$   
 $\wedge \text{getX}(\text{step}(L)) = \text{getX}(L)$   
 $\wedge \text{getY}(\text{step}(L)) = \text{getY}(L) + 1$   
 $\wedge \text{!GameEng}::\text{isObstacle}(\text{getGameEng}(L), \text{getX}(L), \text{getY}(L)+1)$
- $\text{Level}::\text{isExit}(\text{GameEng}::\text{getLevel}(\text{getGameEng}(\text{step}(L)), \text{getX}(\text{step}(L)), \text{getY}(\text{step}(L)))) = \text{true}$   
 $\Rightarrow \text{GameEng}::\text{getNombreSauves}(\text{getGameEng}(\text{step}(L))) = \text{GameEng}::\text{getNombreSauves}(\text{getGameEng}(L)) + 1$   
 $\wedge \text{getId}(L) \notin \text{GameEng}::\text{getLemmingsActifs}(\text{getGameEng}(\text{step}(L)))$