Ecran

Service: Screen

Observators:
const Height: [Screen] → int
const Width: [Screen] → int
CellNature: [Screen] × int × int → Cell
pre CellNature(S,x,y) **requires** 0 ≤ y < Height(S) **and** 0 ≤ x < Width(S)

Constructors:
init: int × int → [Screen]
pre init(h,w) **requires** 0 < h **and** 0 < w

Operators:
Dig: [Screen] × int × int → [Screen]
pre Dig(S,x,y) **requires** CellNature(S,x,y) = **PLT**
Fill: [Screen] × int × int → [Screen]
pre Dig(S,x,y) **requires** CellNature(S,x,y) = **HOL**

Observations:

[init]:
Height(init(h,w)) = h
Width(init(h,w)) = w
forall (x, y) **in** [0;Width(S)[× [0;Height(S)[, CellNature(init(h,w),x,y) = **EMP**

[Dig]:
CellNature(Dig(S,x,y)),x,y = **HOL**
forall (x, y) **in** [0;Width(S)[× [0;Height(S)[,
(x ≠ u **or** y ≠ v) **implies** CellNature(Dig(S,u,v)),x,y) = CellNature(x,y)

[Fill]:
CellNature(Fill(S,x,y),x,y) = **PLT**
forall (x, y) **in** [0;Width(S)[× [0;Height(S)[,
(x ≠ u or y ≠ v) **implies** CellNature(Fill(S,u,v)),x,y) = CellNature(x,y)

Ecran éditable

Service: EditableScreen **includes** Screen

Observators: Playable: [EditableScreen] → bool

Operators: SetNature: [EditableScreen] × int × int × Cell → [EditableScreen]

pre SetNature(S,x,y,C) **requires** 0 ≤ y < Height(S) and 0 ≤ x < Width(S)

Observations:

[invariant]: Playable(S) **min**

forall (x, y) **in** [0;Width(S)[× [0;Height(S)[, CellNature(S,x,y) ≠ **HOL**

and forall x **in** [0;Width(S)[, CellNature(S,x,0) = **MTL**

[SetNature]: CellNature(SetNature(S,x,y,C)),x,y = C

forall (x, y) **in** [0;Width(S)[× [0;Height(S)[,

(x ≠ u or y ≠ v) **implies**

CellNature(SetNature(S,u,v,C)),x,y) = CellNature(x,y)

Environment

Service: Environment **includes** Screen

Observators: CellContent: int × int → Set{Character + Item}

pre CellContent(E,x,y) **requires** 0 ≤ y < Height(S) and 0 ≤ x < Width(S)

toString : [Environnement] → String

Constructors: init: EditableScreen → Environment

Observations:

[invariant] : **forall** (x; y) **in** [0;Width(E)[× [0;Height(E)[,

forall Character c1, c2 **in** CellContent(E,x,y)², c1 = c2

forall (x; y) **in** [0;Width(E)[× [0;Height(E)[,

CellNature(E,x,y) **in** {**MTL**, **PLR, TLP**} **implies** CellContent(x,y) = Ø

forall (x; y) **in** [0;Width(E)[× [0;Height(E)[,

exists Treasure t **in** CellContent(E,x,y)

implies (CellNature(E,x,y) = **EMP and**

CellNature(E,x,y-1) in {**PLT**, **MTL,TLP**})

[init]: **forall** (x; y) **in** [0;Width(E)[× [0;Height(E)[,

CellNature(init(S),x,y) = EditableScreen::CellNature(S,x,y)

Character

Service: Character

Observators: **const** Envi: [Character] → Environment
Hgt: [Character] → int
Wdt: [Character] → int
Id : [Character] → int

Constructors: init: Screen × int × int × int → [Character]
pre init(S,x,y) **requires** Environment::CellNature(S,x,y) = **EMP**

Operators: SetPos : [Character] × int × int → [Character]
pre setPos(C,x,y) **requires**
0 ≤ x < Width(Envi(C)) **and**
0 ≤ y < Height(Envi(C)) **and**
Environment::CellNature(Wdt(C),Hgt(C)) not in {MTL, PLT ,TLP}
GoLeft: [Character] → [Character]
GoRight: [Character] → [Character]
GoUp: [Character] → [Character]
GoDown: [Character] → [Character]

Observations:

[invariant]: Environment::CellNature(Envi(C),Wdt(C),Hgt(C)) **in** {EMP, HOL, LAD, HDR}

[Init]: GetWdt(Init(screen, x,y,id)) = x
GetHgt(Init(screen, x,y,id)) = y
id > -1 **implies** GetId(Init(screen,x,y,id)) = id

[GoLeft]: Hgt(GoLeft(C)) = Hgt(C)
Wdt(C) = 0 **implies** Wdt(GoLeft(C)) = Wdt(C)
Environment::CellNature(Envi(C),Wdt(C)-1,Hgt(C)) **in** {**MTL**, **PLT**, **TLP** }
implies Wdt(GoLeft(C)) = Wdt(C)
Environment::CellNature(Envi(C),Wdt(C),Hgt(C)) **not in** {**LAD**, **HDR**}
and Environment::CellNature(Envi(C),Wdt(C),Hgt(C)-1) **not in** {**PLT**, **MTL**, **LAD**,**TLP**}
and not exists Character c **in** Environment::CellContent(Envi(C),Wdt(C),Hgt(C)-1)
implies Wdt(GoLeft(C)) = Wdt(C)
exists Character c **in** Environment::CellContent(Envi(C),Wdt(C)-1,Hgt(C))
implies Wdt(GoLeft(C)) = Wdt(C)
(Wdt(C) ≠ 0) **and** Environment::CellNature(Envi(C),Wdt(C)-1,Hgt(C)) **not in** {**MTL**, **PLT**, **TLP** }
and (Environment::CellNature(Envi(C),Wdt(C),Hgt(C)) in {**LAD**, **HDR**}
or Environment::CellNature(Envi(C),Wdt(C),Hgt(C)-1)
in {**PLT**, **MTL**, **LAD**, **TLP**}
or exists Character c **in** Environment::CellContent(Envi(C),Wdt(C),Hgt(C)-1))
and (Id(C) = -1
or (**not exists** Character c **in** Environment::CellContent(Envi(C),Wdt(C)-1,Hgt(C))))
implies Wdt(GoLeft(C)) = Wdt(C)-1

[GoRight]: Hgt(GoRight(C)) = Hgt(C)
Wdt(C) = Width(Envi(C))-1 **implies** Wdt(GoRight(C)) = Wdt(C)
Environment::CellNature(Envi(C),Wdt(C)+1,Hgt(C)) **in** {**MTL**, **PLT**, **TLP** }
implies Wdt(GoRight(C)) = Wdt(C)
Environment::CellNature(Envi(C),Wdt(C),Hgt(C)) **not in** {**LAD**, **HDR**}
and Environment::CellNature(Envi(C),Wdt(C),Hgt(C)-1) **not in** {**PLT**, **MTL**, **LAD**,**TLP**}
and not exists Character c **in** Environment::CellContent(Envi(C),Wdt(C),Hgt(C)-1)
implies Wdt(GoRight(C)) = Wdt(C)
exists Character c **in** Environment::CellContent(Envi(C),Wdt(C)+1,Hgt(C))
implies Wdt(GoRight(C)) = Wdt(C)
(Wdt(C) ≠ Width(Envi(C))-1)
and Environment::CellNature(Envi(C),Wdt(C)+1,Hgt(C)) **not in** {**MTL**, **PLT**, **TLP** }
and (Environment::CellNature(Envi(C),Wdt(C),Hgt(C)) in {**LAD**, **HDR**}
or Environment::CellNature(Envi(C),Wdt(C),Hgt(C)-1)
in {**PLT**, **MTL**, **LAD**, **TLP**}
or exists Character c **in** Environment::CellContent(Envi(C),Wdt(C),Hgt(C)-1))
and (Id(C) = -1
or (**not exists** Character c **in** Environment::CellContent(Envi(C),Wdt(C)+1,Hgt(C))))
implies Wdt(GoRight(C)) = Wdt(C)+1

[GoUp]: Wdt(GoUp(C)) = Wdt(C)
Hgt(C) = Environment::Height(Envi(C))-1 **implies** Hgt(GoUp(C)) = Hgt(C)
Environment::CellNature(Envi(C),Wdt(C)+1,Hgt(C)) **in** {**MTL**, **PLT**, **TLP** }
implies Hgt(GoUp(C)) = Hgt(C)
Environment::CellNature(Envi(C),Wdt(C),Hgt(C)) **not in** {**LAD**}
implies Hgt(GoUp(C)) = Hgt(C)
exists Guard g **in** Environment::CellContent(Envi(C),Wdt(C),Hgt(C)+1) **and** Id(C) < -1
implies Wdt(GoUp(C)) = Wdt(C)
(Hgt(C) ≠ Environment::Height(Envi(C))-1)
and Environment::CellNature(Envi(C),Wdt(C),Hgt(C)+1) **not in** {**MTL**, **PLT**, **TLP** }
and Environment::CellNature(Envi(C),Wdt(C),Hgt(C)) in {**LAD**}
and (Id(C) = -1
or (**not exists** Character c **in** Environment::CellContent(Envi(C),Wdt(C),Hgt(C)+1)))
implies Hgt(GoUp(C)) = Hgt(C)+1

[GoDown]: Wdt(GoUp(C)) = Wdt(C)
Hgt(C) = 1 **implies** Hgt(GoDown(C)) = Hgt(C)
Environment::CellNature(Envi(C),Wdt(C)+1,Hgt(C)) **in** {**MTL**, **PLT**, **TLP** }
implies Hgt(GoDown(C)) = Hgt(C)
exists Guard g **in** Environment::CellContent(Envi(C),Wdt(C),Hgt(C)-1) **and** Id(C) < -1
implies Wdt(GoDown(C)) = Wdt(C)
(Hgt(C) ≠ 1)
and Environment::CellNature(Envi(C),Wdt(C),Hgt(C)+1) **not in** {**MTL**, **PLT**, **TLP** }
and Environment::CellNature(Envi(C),Wdt(C),Hgt(C)) in {**LAD**}
and (Id(C) = -1
or (**not exists** Character c **in** Environment::CellContent(Envi(C),Wdt(C),Hgt(C)+1)))
implies Hgt(GoDown(C)) = Hgt(C)-1

[setPos]: Hgt(SetHgt(C,y)) = Hgt(C) **and** Wdt(SetWdt(C,x)) = Wdt(C)//remplace celle de setHgt et setWdt

Guard

Service: Guard **includes** Character

Observators: **const** Id: [Guard] → int
Target: [Guard] → Player
Behaviour : [Guard] → Command
TimeInHole: [Guard] → int
Engine: [Guard] → Engine
HasItem: [Guard] → boolean
Treasure : [Guard] → Item

Predicates:

willFall : [Guard] → boolean
willFall(g) **defined by**
Environment::CellNature(Envi(E),Wdt(g),Hgt(g)) **not in** {**LAD**,**HDR**,**HOL**} **and**
Environment::CellNature(Envi(E),Wdt(g),Hgt(g)-1) **in** {**HDR**,**HOL**,**EMP**} **and**
not exists Guard gbis **in** Environment::CellContent(Envi(E),Wdt(g),Hgt(g)-1)

willWaitInHole : [Guard] → boolean
willWaitInHole(g) **defined by**
Environment::CellNature(Envi(E),Wdt(g),Hgt(g)) = **HOL and**
Guard::TimeInHole(g) < 10

willClimbLeft : [Guard] → boolean
willClimbLeft(g) **defined by**
Environment::CellNature(Envi(E),Wdt(g),Hgt(g)) = **HOL and**
Guard::TimeInHole(g) = 10 **and**
Guard::Behaviour(g) = **LEFT**

willClimbRight : [Guard] → boolean
willClimbRight(g) **defined by**
Environment::CellNature(Envi(E),Wdt(g),Hgt(g)) = **HOL and**
Guard::TimeInHole(g) = 10 **and**
Guard::Behaviour(g) = **RIGHT**

willClimbNeutral : [Guard] → boolean
willClimbNeutral(g) **defined by**
Environment::CellNature(Envi(E),Wdt(g),Hgt(g)) = **HOL and**
Guard::TimeInHole(g) = 10 **and**
Guard::Behaviour(g) = **NEUTRAL**

willFollowBehaviour : [Guard] → boolean
not willFall and not willWaitInHole and not willClimbLeft and not willClimbRight and not willClimbNeutral

Constructors: init: Screen × int × int → [Character]
pre init(S,x,y) **requires** Environment::CellNature(S,x,y) = **EMP**

Operators: WaitInHole: [Guard] → [Guard]
pre WaitInHole(G) **requires** Environment::CellNature(Envi(G),Wdt(G),Hgt(G)) = **HOL**
SetTreasure: [Guard] × Item → [Guard]
pre SetTreasure(G,t) **requires not** HasItem(G)

ClimbLeft: [Guard] → [Guard]
pre ClimbLeft(g) **requires** Environment::CellNature(Envi(G),Wdt(G),Hgt(G)) = **HOL**
ClimbRight: [Guard] → [Guard]
pre ClimbRight(g) **requires** Environment::CellNature(Envi(G),Wdt(G),Hgt(G)) = **HOL**
Step: [Guard] → [Guard]

Observations:
[Invariants]

//UP
Environment::CellNature(Envi(G),Wdt(G),Hgt(G)) **in** {**LAD**} **and**
Hgt(G) < Hgt(Target(G)) **and**
Environment::CellNature(Envi(G),Wdt(G),Hgt(G)-1) **not in** {**PLT**,**MTL**,**TLP**} **and**
not exists Guard g **in** Environnement::getCellContent(Envi(G),Wdt(G),Hgt(G)-1) **and**
|Environment::Hgt(Target(G)) - Hgt(G)| > |Environment::Wdt(Target(G)) - Wdt(G))|) **implies**
Behaviour(G) = Command::**UP**
//DOWN
Environment::CellNature(Envi(G),Wdt(G),Hgt(G)) **in** {**LAD**} **and**
Hgt(G) > Hgt(Target(G)) **and**
Environment::CellNature(Envi(G),Wdt(G),Hgt(G)-1) **not in** {**PLT**,**MTL**,**TLP**} **and**
not exists Guard g **in** Environnement::getCellContent(Envi(G),Wdt(G),Hgt(G)-1) **and**
|Environment::Hgt(Target(G)) - Hgt(G)| > |Environment::Wdt(Target(G)) - Wdt(G))|) **implies**
Behaviour(G) = Command::**DOWN**
//LEFT
(Environment::CellNature(Envi(G),Wdt(G),Hgt(G)) **in** {**LAD**} **and**
Wdt(G) < Wdt(Target(G)) **and**
Environment::CellNature(Envi(G),Wdt(G),Hgt(G)-1) **not in** {**PLT**,**MTL**,**TLP**} **and**
not exists Guard g **in** Environnement::getCellContent(Envi(G),Wdt(G),Hgt(G)-1) **and**
|Environment::Hgt(Target(G)) - Hgt(G)| < |Environment::Wdt(Target(G)) - Wdt(G))|)
or
((Environment::CellNature(Envi(G),Wdt(G),Hgt(G)) **in** {**HOL**} **or**
(Environment::CellNature(Envi(G),Wdt(G),Hgt(G)-1) **in** {**PLT**,**MTL**,**TLP**} **and**
Environment::CellNature(Envi(G),Wdt(G),Hgt(G)) **not in** {**LAD**}) **or**
(**exists** Guard g **in** Environnement::getCellContent(Envi(G),Wdt(G),Hgt(G)-1) **and**
Environment::CellNature(Envi(G),Wdt(G),Hgt(G)) **not in** {**LAD**})) **and**
Wdt(G) > Wdt(Target(G)) **implies** Behaviour(G) = Command::**LEFT**
//RIGHT
(Environment::CellNature(Envi(G),Wdt(G),Hgt(G)) **in** {**LAD**} **and**
Wdt(G) > Wdt(Target(G)) **and**
Environment::CellNature(Envi(G),Wdt(G),Hgt(G)-1) **not in** {**PLT**,**MTL**,**TLP**} **and**
not exists Guard g **in** Environnement::getCellContent(Envi(G),Wdt(G),Hgt(G)-1) **and**
|Environment::Hgt(Target(G)) - Hgt(G)| < |Environment::Wdt(Target(G)) - Wdt(G))|)
or
((Environment::CellNature(Envi(G),Wdt(G),Hgt(G)) **in** {**HOL**} **or**
(Environment::CellNature(Envi(G),Wdt(G),Hgt(G)-1) **in** {**PLT**,**MTL**,**TLP**} **and**
Environment::CellNature(Envi(G),Wdt(G),Hgt(G)) **not in** {**LAD**}) **or**
(**exists** Guard g **in** Environnement::getCellContent(Envi(G),Wdt(G),Hgt(G)-1) **and**
Environment::CellNature(Envi(G),Wdt(G),Hgt(G)) **not in** {**LAD**})) **and**
Wdt(G) < Wdt(Target(G)) **implies** Behaviour(G) = Command::**RIGHT**

Environment::CellNature(Envi(G),Wdt(G),Hgt(G)) **in** {**HDR,EMP**} **and**
Environment::CellNature(Envi(G),Wdt(G),Hgt(G)-1) **in** {**PLT**,**MTL**,**TLP**} **or**
exists Guard g **in** Environnement::getCellContent(Envi(G),Wdt(G),Hgt(G)-1) **and**
Wdt(G) == Wdt(Target(G)) **implies** Behaviour(G) = Command::**NEUTRAL**

Environment::CellNature(Envi(G),Wdt(G),Hgt(G)) **in** {**HDR**} **and**
(Environment::CellNature(Envi(G),Wdt(G),Hgt(G)-1) **in** {**PLT**,**MTL**,**TLP**} **or**
exists Guard g **in** Environnement::getCellContent(Envi(G),Wdt(G),Hgt(G)-1) **or**
|Environment::Hgt(Target(G)) - Hgt(G)| < |Environment::Wdt(Target(G)) - Wdt(G))|) **implies**
(Wdt(G) > Wdt(Target(G)) **implies** Behaviour(G) = Command::**LEFT) or**
(Wdt(G) < Wdt(Target(G)) **implies** Behaviour(G) = Command::**RIGHT) or**
(Wdt(G) = Wdt(Target(G)) **implies** Behaviour(G) = Command::**NEUTRAL))**
or
(Environment::CellNature(Envi(G),Wdt(G),Hgt(G)-1) **not in** {**PLT**,**MTL**,**TLP**} **and**
not exists Guard g **in** Environnement::getCellContent(Envi(G),Wdt(G),Hgt(G)-1) **and**
|Environment::Hgt(Target(G)) - Hgt(G)| < |Environment::Wdt(Target(G)) - Wdt(G))| **implies**
[(Wdt(G) > Wdt(Target(G)) **implies** Behaviour(G) = Command::**LEFT) or**
(Wdt(G) < Wdt(Target(G)) **implies** Behaviour(G) = Command::**RIGHT) or**
(Wdt(G) = Wdt(Target(G)) **implies** Behaviour(G) = Command::**NEUTRAL)]**
or
[(Hgt(G) > Hgt(Target(G)) **implies** Behaviour(G) = Command::**DOWN) or**
(Hgt(G) < Hgt(Target(G)) **implies** Behaviour(G) = Command::**UP) or**
(Hgt(G) = Hgt(Target(G)) **implies** Behaviour(G) = Command::**NEUTRAL)]))**

exists Guard g **in** Environnement::CellContent(Envi(G),Wdt(G),Hgt(G)) **implies** g = G

[Init]

TimeInHole(init(e,h,w,t)) = 0
Target(init(e,h,w,t)) = t

[WaitInHole]

TimeInHole(WaitInHole(G)) = TimeInHole(G)+1

[ClimLeft]

Environment::CellNature(Envi(G),Wdt(G),Hgt(G)+1) **not in** {**PLT**,**MTL**,**TLP**} **and**
Environment::CellNature(Envi(G),Wdt(G)-1,Hgt(G)+1) **not in** {**PLT**,**MTL**,**TLP**} **and**
not exists Character c **in** Environnement::getCellContent(Envi(G),Wdt(G)-1,Hgt(G)+1)
implies Wdt(ClimbLeft(G)) = Wdt(G)-1 **and** Hgt(ClimbLeft(G)) = Hgt(G)+1

[ClimRight]

Environment::CellNature(Envi(G),Wdt(G),Hgt(G)+1) **not in** {**PLT**,**MTL**,**TLP**} **and**
Environment::CellNature(Envi(G),Wdt(G)+1,Hgt(G)+1) **not in** {**PLT**,**MTL**,**TLP**} **and**
not exists Character c **in** Environnement::getCellContent(Envi(G),Wdt(G)+1,Hgt(G)+1)
implies Wdt(ClimbRight(G)) = Wdt(G)+1 **and** Hgt(ClimbRight(G)) = Hgt(G)+1

[SetTreasure]

HasTreasure(SetTreasure(G,t)) = true

[Step]

WillFall(G) **implies** GetBehaviour(G) = Command::DOWN **and**
Wdt(Step(G)) = Wdt(G) **and** Hgt(Step(G)) = Hgt(G)-1 **and**
TimeInHole(Step(G)) = TimeInHole(G)

WillClimbLeft(G) **implies** GetBehaviour(G) = Command::LEFT **and**
((Wdt(Step(G)) = Wdt(G)-1 **and** Hgt(Step(G))= Hgt(G)+1) **or**
(Wdt(Step(G)) ≠ Wdt(G) **and** Hgt(Step(G)) ≠ Hgt(G))) **and**
TimeInHole(Step(G)) = TimeInHole(G)
WillClimbRight(G) **implies** GetBehaviour(G) = Command::RIGHT **and**
((Wdt(Step(G)) = Wdt(G)+1 **and** Hgt(Step(G))= Hgt(G)-11) **or**
(Wdt(Step(G)) ≠ Wdt(G) **and** Hgt(Step(G)) ≠ Hgt(G))) **and**
TimeInHole(Step(G)) = TimeInHole(G)
WillClimbNeutral(G) **implies** GetBehaviour(G) = Command::NEUTRAL **and**
Wdt(Step(G)) = Wdt(G) **and** Hgt(Step(G)) = Hgt(G) **and**
TimeInHole(Step(G)) = TimeInHole(G)
WillWaitInHole(G) **implies** GetBehaviour(G) = Command::NEUTRAL **and**
Wdt(Step(G)) = Wdt(G) **and** Hgt(Step(G)) = Hgt(G) **and**
TimeInHole(Step(G)) = TimeInHole(G)+1

Player

Service: Player **includes** Character, use **Coord**

Observators: Engine : [Player] → [Engine]
HasGauntlet : [Player] → boolean
Gauntlet : [Player] → Item
ClonePlayer : [Player] → [Player]

Constructors: init: Engine × Coord → [Character]
pre init(E,c) **requires**
Environment::CellNature(Envi(E),GetX(c),GetY(c)) = **EMP**
and e ≠ Ø **and** c ≠ Ø **and**
0 ≤ GetX(c) < Width(Envi(C)) **and**
0 ≤ GetY(c) < Height(Envi(C))

Operators: Step : [Player] → [Player]
HitLeft : [Player] → [Player]
pre HitLeft(P) **requires** HasGauntlet(P)
HitRight : [Player] → [Player]
pre HitRight(P) **requires** HasGauntlet(P)
SetGauntlet : [Player] × Item → [Player]
pre setGauntlet(g) **requires** Item:Nature(g) = ItemType::Gauntlet **and**
g ≠ Ø **and (not** hasGauntlet(P))

Observations:

[Invariant]
exists Character c in Environment::CellContent(Envi(P),Wdt(P),Hgt(P)) **implies** c = P

[Init]
GetWdt(Init(E,c)) = GetX(c) **and** GetHgt(Init(E,c)) = GetY(c) **and** Engine(P) = E

[HitLeft]
HasGauntlet(HitLeft(P)) = **false**

[HitRight]
HasGauntlet(HitRight(P)) = **false**

[SetGauntlet]
HasGauntlet(SetGauntlet(P,g)) = **true**
GetGauntlet(SetGauntlet(P,g)) = **g**

[Step] clone **defined by** ClonePlayer(P) **in**
Wdt(Step(clone)) = Wdt(Step(P)) **and** Hgt(Step(clone)) = Hgt(Step(P))

Environment::CellNature(Envi(P),Wdt(P),Hgt(P)) **not in** {**LAD**,**HDR**} **and**
Environment::CellNature(Envi(P),Wdt(P),Hgt(P)-1) **not in** {**PLT**,**MTL**,**TLP**,**LAD**} **and**
not exists Guard g **in** Environnement::getCellContent(Envi(P),Wdt(P),Hgt(P)-1) **implies**
Hgt(Step(P)) = Hgt(P)-1

command **defined by** Engine::NextCommand(Engine(P)) **in**
command = Command::DIGL **and**
Environment::CellNature(Envi(P),Wdt(P)-1,Hgt(P)-1) **in** {**PLT**} **implies**
Environment::CellNature(Envi(Step(clone)),Wdt(P)-1,Hgt(P)-1) = Cell::**HOL and**
Environment::CellNature(Envi(Step(P)),Wdt(P)-1,Hgt(P)-1) = Cell::**HOL**

Environment::CellNature(Envi(P),Wdt(P),Hgt(P)-1) **in** {**PLT**,**MTL**,**TLP**,**LAD**} **or**
exists Guard g **in** Environnement::getCellContent(Envi(P),Wdt(P),Hgt(P)-1) **and**
Environment::CellNature(Envi(P),Wdt(P)-1,Hgt(P)) **in** {**EMP,LAD,HDR,HOL**} **and**
Environment::CellNature(Envi(P),Wdt(P)-1,Hgt(P)-1) **in** {**PLT**} **and**
Environment::CellNature(Envi(P),Wdt(P)-1,Hgt(P)-1) **not in** {**HOL**} **implies**
Environment::CellNature(Envi(Step(P)),Wdt(P)-1,Hgt(P)-1) = Cell::**HOL**

command **defined by** Engine::NextCommand(Engine(P)) **in**
command = Command::DIGR **and**
Environment::CellNature(Envi(P),Wdt(P)+1,Hgt(P)-1) **in** {**PLT**} **implies**
Environment::CellNature(Envi(Step(clone)),Wdt(P)+1,Hgt(P)-1) = Cell::**HOL and**
Environment::CellNature(Envi(Step(P)),Wdt(P)+1,Hgt(P)-1) = Cell::**HOL**

Environment::CellNature(Envi(P),Wdt(P),Hgt(P)-1) **in** {**PLT**,**MTL**,**TLP**,**LAD**} **or**
exists Guard g **in** Environnement::getCellContent(Envi(P),Wdt(P),Hgt(P)-1) **and**
Environment::CellNature(Envi(P),Wdt(P)+1,Hgt(P)) **in** {**EMP,LAD,HDR,HOL**} **and**
Environment::CellNature(Envi(P),Wdt(P)+1,Hgt(P)-1) **in** {**PLT**} **and**
Environment::CellNature(Envi(P),Wdt(P)+1,Hgt(P)-1) **not in** {**HOL**} **implies**
Environment::CellNature(Envi(Step(P)),Wdt(P)+1,Hgt(P)-1) = Cell::**HOL**

Engine

Service: Engine

Observators:
Envi : [Engine] → [Environment]
Player : [Engine] → [Player]
Guards : [Engine] → **listOf** [Guard]
Treasures : [Engine] → **listOf** Item
Status : [Engine] → GameState
Holes: [Engine] → **listOf** Hole
Commands: [Engine] → **listOf** Command
NextCommand: [Engine] → Command
Teleporteurs: [Engine] → **listOf** Teleporteur
Gauntlet: [Engine] → Item

Constructor: Init: EditableScreen × Coord × **listOf** Coord × **listOf** Item × **listOf** Teleporteur × Item → [Engine]

pre init(es,p,guards,treasures,tp,gant) **requires**
EditableScreen::Playable(es)
and
Coord::X(p) ≠ Item::Col(gant) **or** Coord ::Y(p) ≠ Item::Hgt(gant)
and
forall Coord co **in** guards
Coord::X(co) ≠ Coord::X(p) **or** Coord::Y(co) ≠ Coord::Y(p)
Coord::X(co) ≠ Item::Col(gant) **or** Coord::Y(co) ≠ Item::Hgt(gant)
forall Coord cobis in guards\co
Coord::X(co) ≠ Coord::X(cobis) **or** Coord::Y(co) ≠ Coord::Y(cobis)
forall Item t **in** treasures
Coord::X(co) ≠ Item::Col(t) **or** Coord::GetY(co) ≠ Item::Hgt(t)
and
forall Item t **in** treasures
0 ≤ Item::Col(t) < Environment::Width(Envi(E)) **and** 0 ≤ Item::Hgt(t) < Environment::Height(Envi(E))
Item::Col(t) ≠ Coord::X(p) **or** Item::Hgt(t) ≠ Coord::Y(p)
Environment::CellNature(Envi(E),Item::Col(t),Item::Hgt(t)) = **EMP and** Environment::CellNature(Envi(E),Item::Col(t),Item::Hgt(t)-1) **in** {**MTL**,**PLT**}
Item::Col(t) ≠ Item::Col(gant) **or** Item::Hgt(t) ≠ Item::Hgt(gant)
forall Item tbis **in** treasures\t
Item::Col(t) ≠ Item::Col(tbis) **or** Item::Hgt(t) ≠ Item::Hgt(tbis)
and
forall Teleporteur t **in** tp
Environment::CellNature(Envi(E),Coord::X(Teleporteur::PosA),Coord::Y(Teleporteur::PosA)) = **PLT**
Environment::CellNature(Envi(E),Coord::X(Teleporteur::PosB),Coord::Y(Teleporteur::PosB)) = **PLT**

Coord::X(Teleporteur::PosA) = Coord::X(p) **implies** Coord::Y(Teleporteur::PosA) ≠ Coord::Y(p)-1
Coord::X(Teleporteur::PosB) = Coord::X(p) **implies** Coord::Y(Teleporteur::PosB) ≠ Coord::Y(p)-1

Operators: Step: [Engine] → [Engine]
AddCommand: [Engine] × Command → [Engine]

Observations:

[Invariant]:

Player(E) ∈ Environment::CellContent(Envi(E),Character::Wdt(Player(E)),Character::Hgt(Player(E))
forall Guard g **in** Guards(E)
g ∈ Environment::CellContent(Envi(E),Character::Wdt(g),Character::Hgt(g))
forall Item t **in** Treasures(E)
t ∈ Environment::CellContent(Envi(E),Item::Col(t),Item::Hgt(t))
forall Hole h **in** Holes(E)
Environment::CellNature(Envi(E),Hole::X(h),Hole::Y(h) = **HOL**

[Init]:

Status(init(es,p,guards,treasures,tp,gant)) = Playing
Score(init(es,p,guards,treasures,tp,gant)) = 0
Holes (init(es,p,guards,treasures,tp,gant)) = **empty**
Commands(init(es,p,guards,treasures,tp,gant)) = **empty**
Player(init(es,p,guards,treasures,tp,gant)) ∈
Environment::CellContent(Envi(init(es,p,guards,treasures,tp,gant)),Character::Wdt(Coord::X(p),Coord::Y(p)))
forall Coord gcoord **in** guards
Exists Guard g **in**
Environment::CellContent(Envi(init(es,p,guards,treasures,tp,gant)),Character::Wdt(Coord::X(gcoord),Coord::Y(gcoord)))
forall Item t **in** treasures
Exists Item i **in**
Environment::CellContent(Envi(init(es,p,guards,treasures,tp,gant)),Item::Wdt(Coord::Col(t),Coord::Hgt(t)))
and i = t
forall Teleporteur t **in** teleporteurs
Environment::CellNature(Envi(E),Coord::X(Teleporteur::PosA),Coord::Y(Teleporteur::PosA)) = **TLP and** Environment::CellNature(Envi(E),Coord::X(Teleporteur::PosB),Coord::Y(Teleporteur::PosB)) = **TLP**

[Step]:

NextCommand(E) = **HITR and** Player::HasGauntlet(Player(E))
implies
((**exists** Guard g **in** Guards(E) **and** Character::Wdt(g) > Character::Wdt(Player(E)) **and** Character::Hgt(g) = Character::Hgt(Player(E)) **and** (**forall** Guard gbis **in** Guards(E)\g Character::Wdt(gbis) > Character::Wdt(Player(E)) **and** Character::Hgt(gbis) = Character::Hgt(Player(E)) **and** Character::Wdt(g) < Character::Wdt(gbis)) **and (forall** Integer i **in**]Character::Wdt(Player());Character::Wdt(g)[Environment::CellNature(Envi(E),i,Character::Hgt(Player(E)) **not in {PLT,MLT,TLP}**) **implies** g ∉ Guards(Step(E)))

NextCommand(E) = **HITL and** Player::HasGauntlet(Player(E))
implies
((**exists** Guard g **in** Guards(E) **and** Character::Wdt(g) < Character::Wdt(Player(E)) **and** Character::Hgt(g) = Character::Hgt(Player(E)) **and** (**forall** Guard gbis **in** Guards(E)\g Character::Wdt(gbis) < Character::Wdt(Player(E)) **and** Character::Hgt(gbis) = Character::Hgt(Player(E)) **and** Character::Wdt(g) > Character::Wdt(gbis)) **and (forall** Integer i **in**]Character::Wdt(g);Character::Wdt(Player())[Environment::CellNature(Envi(E),i,Character::Hgt(Player(E)) **not in {PLT,MLT,TLP}**) **implies** g ∉ Guards(Step(E)))

forall Item t : Treasures(E)
(Item::Col(t) = Character::Wdt(Player(Step(E))) **and** Item::Hgt(t) = Character::Hgt(Player(Step(E))))
or
(**exists** Guard g **in** Guards(Step(E))
Item::Col(t) = Character::Wdt(g) **and** Item::Hgt(t) = Character::Hgt(g))
implies t ∉ Treasures(Step(E))
exists Guard g **in** Guards((E)) (Character::Wdt(g) = Character::Wdt(Player(E)) **and** Character::Hgt(g) = Character::Hgt(Player(E)) **implies** Status(Step(E)) = Loss
forall Hole h : Holes(E)
Hole::T(h) = 14
implies (h ∉ Holes(Step(E))
and
exists Guards g **in** Environment::CellContent(Envi(E),Character::Wdt(g),Character:Hgt(g))
implies g ∉ Guards(Step(E))
and
exists Player p **in**
Environment::CellContent(Envi(E),Character::Wdt(Player(E)),Character:Hgt(Player(E))) **implies** Status(Step(E)) = Loss)
Hole::T(h) < 14
implies ((h ∈ Holes(Step(E))
and
exists Hole hole **in** Holes(Step(E)) h = hole **implies** Hole::T(hole) = Hole::T(h)+1
list::size(Commands(Step(E))) = list::size(Commands(E))

[AddCommand]
list::size(Commands(AddCommand(E,c))) = list::size(Commands(E))+1