Service: Player **include** Entity

Observators: LastCom: [Player] → Option[Command]

Content: [Player] × int × int → Option[Mob]

pre Content(P,x,y) **requires** x ∈ $\{-1, 0, 1\}$ and y ∈ $\{-1, +3\}$

Nature: [Player] × int × int → Cell

pre Nature(P,x,y) **requires** x ∈ $\{-1, 0, 1\}$ and y ∈ $\{-1, +3\}$

Viewable: [Player] × int × int → Cell

pre Nature(P,x,y) **requires** x ∈ $\{-1, 0, 1\}$ and y ∈ $\{-1, +3\}$

Ressource : [Player] → Ressource

Key: [Player] → boolean

Win: [Player] → boolean

pre Win(P) **requires** Ressource(P) = TREASOR

Dead: [Player] → boolean

pre Dead(P) **requires** Hp(P) ≤ 0

Operators:

openDoor:[Player] → [Player]

pre OpenDoor(P) **require** Key(P) = true **and**

Face(P) = N **implies** Environement::CellNature(Envi(P), Col(p), Row(p) - 1) ∈ $\{DWC\}$
and Environement::CellContent(Envi(P), Col(p), Row(p) - 1) = NO

Face(P) = E **implies** Environement::CellNature(Envi(P), Col(p) + 1, Row(p)) ∈ $\{DNC\}$
and Environement::CellContent(Envi(P), Col(p) + 1, Row(p)) = NO

Face(P) = S **implies** Environement::CellNature(Envi(P), Col(p), Row(p) + 1) ∈ $\{DWC\}$
and Environement::CellContent(Envi(P), Col(p), Row(p) + 1) = NO

Face(P) = W **implies** Environement::CellNature(Envi(P), Col(p) - 1, Row(p)) ∈ $\{DNC\}$
and Environement::CellContent(Envi(P), Col(p) - 1, Row(p)) = NO

CloseDoor:[Player] → [Player]

pre CloseDoor(P) **require**

Face(P) = N **implies** Environement::CellNature(Envi(P), Col(p), Row(p) - 1) ∈ $\{DWO\}$
and Environement::CellContent(Envi(P), Col(p), Row(p) - 1) = NO

Face(P) = E **implies** Environement::CellNature(Envi(P), Col(p) + 1, Row(p)) ∈ $\{DNO\}$
and Environement::CellContent(Envi(P), Col(p) + 1, Row(p)) = NO

Face(P) = S **implies** Environement::CellNature(Envi(P), Col(p), Row(p) + 1) ∈ $\{DWC\}$
and Environement::CellContent(Envi(P), Col(p), Row(p) + 1) = NO

Face(P) = W **implies** Environement::CellNature(Envi(P), Col(p) - 1, Row(p)) ∈ $\{DNC\}$
and Environement::CellContent(Envi(P), Col(p) - 1, Row(p)) = NO

Observations:

[**Invariants**]: Face(P) = N
implies Content(P,u,v) = Environment:CellContent(Envi(P),Col(P)+u,Row(P)+v)
Face(P) = N
implies Nature(P,u,v) = Environment:CellNature(Envi(P),Col(P)+u,Row(P)+v)
Face(P) = S
implies Content(P,u,v) = Environment:CellContent(Envi(P),Col(P)-u,Row(P)-v)
Face(P) = S
implies Nature(P,u,v) = Environment:CellNature(Envi(P),Col(P)-u,Row(P)-v)
Face(P) = E
implies Content(P,u,v) = Environment:CellContent(Envi(P),Col(P)+v,Row(P)-u)
Face(P) = E
implies Nature(P,u,v) = Environment:CellNature(Envi(P),Col(P)+v,Row(P)-u)
Face(P) = W
implies Content(P,u,v) = Environment:CellContent(Envi(P),Col(P)-v,Row(P)+u)
Face(P) = W
implies Nature(P,u,v) = Environment:CellNature(Envi(P),Col(P)-v,Row(P)+u)
forall u,v in [-1,1] × [-1,1], not Viewable(P,u,v)
Viewable(P,-1,2)= Nature(P,-1,1) $\notin \{WALL, DWC, DNC\}$
Viewable(P,0,2)= Nature(P,0,1) $\notin \{WALL, DWC, DNC\}$
Viewable(P,1,2)= Nature(P,1,1) $\notin \{WALL, DWC, DNC\}$
Viewable(P,-1,3)= Nature(P,-1,2) $\notin \{WALL, DWC, DNC\}$ and Viewable(P,-1,2)
Viewable(P,0,3)= Nature(P,0,2) $\notin \{WALL, DWC, DNC\}$ and Viewable(P,0,2)
Viewable(P,1,3)= Nature(P,1,2) $\notin \{WALL, DWC, DNC\}$ and Viewable(P,1,2)

[**openDoor**]: Key(OpenDoor(P)) = true
Face(P) = N **implies** Environement::CellNature(Envi(P), Col(p), Row(p) - 1) $\in \{DWO\}$
Face(P) = E **implies** Environement::CellNature(Envi(P), Col(p) + 1, Row(p)) $\in \{DNO\}$
Face(P) = S **implies** Environement::CellNature(Envi(P), Col(p), Row(p) + 1) $\in \{DWO\}$
Face(P) = W **implies** Environement::CellNature(Envi(P), Col(p) - 1, Row(p)) $\in \{DNO\}$

[**ClooseDoor**]: Key(OpenDoor(P)) = Key(p)
Face(P) = N **implies** Environement::CellNature(Envi(P), Col(p), Row(p) - 1) $\in \{DWC\}$
Face(P) = E **implies** Environement::CellNature(Envi(P), Col(p) + 1, Row(p)) $\in \{DNC\}$
Face(P) = S **implies** Environement::CellNature(Envi(P), Col(p), Row(p) + 1) $\in \{DWC\}$
Face(P) = W **implies** Environement::CellNature(Envi(P), Col(p) - 1, Row(p)) $\in \{DNC\}$

[**step**]: LastCom(P)=FF **implies** step(P) = Forward(P)
LastCom(P)=BB **implies** step(P) = Backward(P)
LastCom(P)=LL **implies** step(P) = StrafeLeft(P)
LastCom(P)=RR **implies** step(P) = StrafeRight(P)
LastCom(P)=TL **implies** step(P) = TurnLeft(P)
LastCom(P)=TR **implies** step(P) = TurnRight(P)
LastCom(P)=C **implies** step(P) = Attack(P)
LastCom(P)=CLOSE **implies** step(P) = ColseDoor(P)
LastCom(P)=OPEN **implies** step(P) = OpenDoor(P)