

Test Level

Cas de test : Level::testInit1_0

Conditions initiales : \emptyset

Operations : init(25,20)

Oracle :

Pas de ContractError

getWidth() == 25

getHeight() == 20

isEditing() == true

$\forall (x,y), \text{getNature}(x,y) == \text{Nature.EMPTY}$

Cas de test : Level::testInit1_1

Conditions initiales : \emptyset

Operations : init(0,25)

Oracle :

ContractError pour init

Cas de test : Level::testInit1_2

Conditions initiales : \emptyset

Operations : init(25,0)

Oracle :

ContractError pour init

Cas de test : Level::testGetNature2_0

Conditions initiales : init(10,10)

Operations : getNature(3,3)

Oracle :

Pas de contractError

Cas de test : Level::testGetNature2_1

Conditions initiales : init(10,10)

Operations : getNature(-2,5)

Oracle :

ContractError pour getNature

Cas de test : Level::testGetNature2_2

Conditions initiales : init(10,10)

Operations : getNature(15,5)

Oracle :

ContractError pour getNature

Cas de test : Level::testGetNature2_3

Conditions initiales : init(10,10)

Operations : getNature(3,-2)

Oracle :

ContractError pour getNature

Cas de test : Level::testGetNature2_3

Conditions initiales : init(10,10)

Operations : getNature(3,12)

Oracle :

ContractError pour getNature

Cas de test : Level::testIsExit3_0

Conditions initiales : init(10,10)

Operations : isExit(3,3)

Oracle :

Pas de ContractError

Cas de test : Level::testIsExit3_1

Conditions initiales : init(10,10)

Operations : isExit(-2,5)

Oracle :

ContractError pour isExit

Cas de test : Level::testIsExit3_2

Conditions initiales : init(10,10)

Operations : isExit(15,5)

Oracle :

ContractError pour isExit

Cas de test : Level::testIsExit3_3

Conditions initiales : init(10,10)

Operations : isExit(3,-2)

Oracle :

ContractError pour isExit

Cas de test : Level::testIsExit3_4

Conditions initiales : init(10,10)

Operations : isExit(3,12)

Oracle :

ContractError pour isExit

Cas de test : Level::testIsEntrance4_0

Conditions initiales : init(10,10)

Operations : isExit(3,3)

Oracle :

Pas de ContractError

Cas de test : Level::testIsEntrance4_1

Conditions initiales : init(10,10)

Operations : isExit(-2,5)

Oracle :

ContractError pour isEntrance

Cas de test : Level::testIsEntrance4_2

Conditions initiales : init(10,10)

Operations : isExit(15,5)

Oracle :

ContractError pour isEntrance

Cas de test : Level::testIsEntrance4_3

Conditions initiales : init(10,10)

Operations : isExit(3,-2)

Oracle :

ContractError pour isEntrance

Cas de test : Level::testIsEntrance4_4

Conditions initiales : init(10,10)

Operations : isExit(3,12)

Oracle :

ContractError pour isEntrance

Cas de test : Level::testgoEditing5_0

Conditions initiales :

```
init(10,10)
 $\forall i = 0, \forall j$  setNature(i, j, Nature.METAL)
 $\forall i = 9, \forall j$  setNature(i, j, Nature.METAL)
 $\forall j = 0, \forall i$  setNature(i, j, Nature.METAL)
 $\forall j = 9, \forall i$  setNature(i, j, Nature.METAL)
setEntrance(8, 5)
setNature(5, 6, Nature.METAL)
setExit(5, 5)
goPlay()
```

Operations : goEditing()

Oracle :

```
Pas de ContractError
isEditing() = true
```

Cas de test : Level::testgoEditing5_1

Conditions initiales :

```
init(10,10)
```

Operations : goEditing()

Oracle :

```
ContractError pour goEditing
```

Cas de test : Level::testSetNature6_0

Conditions initiales :

```
init(10,10)
```

Operations : setNature(5, 5, Nature.DIRT)

Oracle :

```
Pas de ContractError
getNature(5, 5) = Nature.DIRT
```

Cas de test : Level::testSetNature6_1

Conditions initiales :

```
init(10,10)
```

Operations : setNature(-2, 5, Nature.DIRT)

Oracle :

```
ContractError pour setNature
```

Cas de test : Level::testSetNature6_2

Conditions initiales :

init(10,10)

Operations : setNature(15, 5, Nature.DIRT)

Oracle :

ContractError pour setNature

Cas de test : Level::testSetNature6_3

Conditions initiales :

init(10,10)

Operations : setNature(5, -5, Nature.DIRT)

Oracle :

ContractError pour setNature

Cas de test : Level::testSetNature6_4

Conditions initiales :

init(10,10)

Operations : setNature(5, 15, Nature.DIRT)

Oracle :

ContractError pour setNature

Cas de test : Level::testSetEntrance7_0

Conditions initiales :

init(10,10)

Operations : setEntrance(5, 5)

Oracle :

Pas de ContractError

isEntrance(5, 5) = true

Cas de test : Level::testSetEntrance7_1

Conditions initiales :

init(10,10)

setNature(5, 4, Nature.DIRT)

Operations : setEntrance(5, 5)

Oracle :

ContractError pour setEntrance

Cas de test : Level::testSetEntrance7_2

Conditions initiales :

init(10,10)

setNature(5, 6, Nature.METAL)

Operations : setEntrance(5, 5)

Oracle :

ContractError pour setEntrance

Cas de test : Level::testSetExit8_0

Conditions initiales :

init(10,10)

setNature(5, 6, Nature.METAL)

Operations : setExit(5, 5)

Oracle :

Pas de ContractError

isExit(5, 5) = true

Cas de test : Level::testSetExit8_1

Conditions initiales :

init(10,10)

setNature(5, 6, Nature.DIRT)

Operations : setExit(5, 5)

Oracle :

ContractError pour setExit

Cas de test : Level::testSetExit8_2

Conditions initiales :

init(10,10)

setNature(5, 4, Nature.METAL)

setNature(5, 6, Nature.METAL)

Operations : setExit(5, 5)

Oracle :

ContractError pour setExit

Cas de test : Level::testGoPlay9_0

Conditions initiales :

init(10,10)
 $\forall i = 0, \forall j$ setNature(i, j, Nature.METAL)
 $\forall i = 9, \forall j$ setNature(i, j, Nature.METAL)
 $\forall j = 0, \forall i$ setNature(i, j, Nature.METAL)
 $\forall j = 9, \forall i$ setNature(i, j, Nature.METAL)
setEntrance(8, 5)
setNature(5, 6, Nature.METAL)
setExit(5, 5)

Operations : goPlay()

Oracle :

Pas de ContractError
isEditing() = false
 $\exists !(i,j)$ tq isEntrance(i, j) = true
 $\exists !(i',j')$ tq isExit(i', j') = true avec $(i,j) \neq (i',j')$

Cas de test : Level::testGoPlay9_1

Conditions initiales :

init(10,10)
 $\forall i = 0, \forall j$ setNature(i, j, Nature.METAL)
 $\forall i = 9, \forall j$ setNature(i, j, Nature.METAL)
 $\forall j = 0, \forall i$ setNature(i, j, Nature.METAL)
 $\forall j = 9, \forall i$ setNature(i, j, Nature.METAL)
setNature(5, 6, Nature.METAL)
setExit(5, 5)

Operations : goPlay()

Oracle :

ContractError pour goPlay

Cas de test : Level::testGoPlay9_1

Conditions initiales :

init(10,10)
 $\forall i = 0, \forall j$ setNature(i, j, Nature.METAL)
 $\forall i = 9, \forall j$ setNature(i, j, Nature.METAL)
 $\forall j = 0, \forall i$ setNature(i, j, Nature.METAL)
 $\forall j = 9, \forall i$ setNature(i, j, Nature.METAL)
setEntrance(5, 5)

Operations : goPlay()

Oracle :

ContractError pour goPlay

Cas de test : Level::testGoPlay9_3

Conditions initiales :

```
init(10,10)
setEntrance(8, 5)
setNature(5, 6, Nature.METAL)
setExit(5, 5)
```

Operations : goPlay()

Oracle :

ContractError pour goPlay

Cas de test : Level::testRemove10_0

Conditions initiales :

```
init(10,10)
 $\forall i = 0, \forall j$  setNature(i, j, Nature.METAL)
 $\forall i = 9, \forall j$  setNature(i, j, Nature.METAL)
 $\forall j = 0, \forall i$  setNature(i, j, Nature.METAL)
 $\forall j = 9, \forall i$  setNature(i, j, Nature.METAL)
setEntrance(8, 5)
setNature(5, 6, Nature.METAL)
setExit(5, 5)
setNature(7, 7, Nature.DIRT)
goPlay()
```

Operations : remove(7,7)

Oracle :

Pas de ContractError
getNature(7, 7) = Nature.EMPTY

Cas de test : Level::testRemove10_1

Conditions initiales :

```
init(10,10)
setNature(7, 7, Nature.DIRT)
```

Operations : remove(7,7)

Oracle :

ContractError pour remove

Cas de test : Level::testRemove10_2

Conditions initiales :

```
init(10,10)
 $\forall i = 0, \forall j$  setNature(i, j, Nature.METAL)
 $\forall i = 9, \forall j$  setNature(i, j, Nature.METAL)
 $\forall j = 0, \forall i$  setNature(i, j, Nature.METAL)
 $\forall j = 9, \forall i$  setNature(i, j, Nature.METAL)
setEntrance(8, 5)
setNature(5, 6, Nature.METAL)
setExit(5, 5)
setNature(7, 7, Nature.METAL)
goPlay()
```

Operations : remove(7,7)

Oracle :

ContractError pour remove

Cas de test : Level::testBuild11_0

Conditions initiales :

```
init(10,10)
 $\forall i = 0, \forall j$  setNature(i, j, Nature.METAL)
 $\forall i = 9, \forall j$  setNature(i, j, Nature.METAL)
 $\forall j = 0, \forall i$  setNature(i, j, Nature.METAL)
 $\forall j = 9, \forall i$  setNature(i, j, Nature.METAL)
setEntrance(8, 5)
setNature(5, 6, Nature.METAL)
setExit(5, 5)
goPlay()
```

Operations : build(7,7)

Oracle :

Pas de ContractError
getNature(7, 7) = Nature.DIRT

Cas de test : Level::testBuild11_1

Conditions initiales :

```
init(10,10)
```

Operations : build(7,7)

Oracle :

ContractError pour build

Cas de test : Level::testBuild11_2

Conditions initiales :

```
init(10,10)
 $\forall i = 0, \forall j$  setNature(i, j, Nature.METAL)
 $\forall i = 9, \forall j$  setNature(i, j, Nature.METAL)
 $\forall j = 0, \forall i$  setNature(i, j, Nature.METAL)
 $\forall j = 9, \forall i$  setNature(i, j, Nature.METAL)
setEntrance(8, 5)
setNature(5, 6, Nature.METAL)
setExit(5, 5)
setNature(7, 7, Nature.METAL)
goPlay()
```

Operations : build(7,7)

Oracle :

ContractError pour build

Cas de test : Level::testBuild11_3

Conditions initiales :

```
init(10,10)
 $\forall i = 0, \forall j$  setNature(i, j, Nature.METAL)
 $\forall i = 9, \forall j$  setNature(i, j, Nature.METAL)
 $\forall j = 0, \forall i$  setNature(i, j, Nature.METAL)
 $\forall j = 9, \forall i$  setNature(i, j, Nature.METAL)
setEntrance(8, 5)
setNature(5, 6, Nature.METAL)
setExit(5, 5)
goPlay()
```

Operations : build(8,5)

Oracle :

ContractError pour build

Cas de test : Level::testBuild11_4

Conditions initiales :

```
init(10,10)
 $\forall i = 0, \forall j$  setNature(i, j, Nature.METAL)
 $\forall i = 9, \forall j$  setNature(i, j, Nature.METAL)
 $\forall j = 0, \forall i$  setNature(i, j, Nature.METAL)
 $\forall j = 9, \forall i$  setNature(i, j, Nature.METAL)
setEntrance(8, 5)
setNature(5, 6, Nature.METAL)
setExit(5, 5)
goPlay()
```

Operations : build(5,5)

Oracle :

ContractError pour build