Table Of Content

Class Main

```
java.lang.Object
    |
    +--Main
```

< Constructors > < Methods >

public class **Main**
extends java.lang.Object

Constructors

Main

```
public  Main()
```

Methods

main

```
public static void main(java.lang.String[] args)
```

Package Agents

Class Summary

Fire

People

Vision

Agents

Class Fire

```
java.lang.Object
    |
    +--Agents.Fire
```

All Implemented Interfaces:
sim.engine.Steppable

< Constructors > < Methods >

---

public class **Fire**
extends java.lang.Object
implements sim.engine.Steppable

Constructors

Fire

```
public  Fire(sim.util.Int2D hearth)
```

Constructor

Parameters:

hearth - This object's location on the grid

Methods

getHearth

```
public sim.util.Int2D getHearth()
```

This method returns this object's coordinates

Returns:

This object's coordinates

---

step

```
public void step(sim.engine.SimState state)
```

---

toString

`public java.lang.String` **`toString`**`()`

The standard toString() method

Overrides:

toString in class java.lang.Object

Agents

Class People

```
java.lang.Object
    |
    +--Agents.People
```

All Implemented Interfaces:
sim.engine.Steppable, sim.portrayal.Oriented2D

< Fields > < Constructors > < Methods >

---

public class **People**
extends java.lang.Object
implements sim.engine.Steppable, sim.portrayal.Oriented2D

Fields

direction

`public Constants.Direction` **`direction`**

---

earX

`public int` **`earX`**

---

earY

`public int` **`earY`**

---

eyeX

`public int` **`eyeX`**

---

eyeY

```
public int eyeY
```

---

isBlocked

```
public boolean isBlocked
```

---

numOfFails

```
public int numOfFails
```

---

seenDirection

```
public Constants.Direction seenDirection
```

Constructors

People

```
public  People(java.lang.String name,
               java.lang.String earX,
               java.lang.String earY,
               java.lang.String eyeX,
               java.lang.String eyeY)
```

Constructor

Parameters:

name - The agent's name
earX - The agent's left eye X-coordinate
earY - The agent's left eye Y-coordinate
eyeX - The agent's left ear X-coordinate
eyeY - The agent's left ear Y-coordinate

Methods

getAutonomyLevel

```
public int getAutonomyLevel()
```

Getter for the autonomyLevel attribute

Returns:

The agent's autonomy level

getCharismaLevel

`public int` **`getCharismaLevel`**`()`

Getter for the charismaLevel attribute

Returns:

The agent's charisma level

---

getHearingAbility

`public int` **`getHearingAbility`**`()`

Getter for the hearingAbility attribute

Returns:

The agent's hearing ability

---

getHearingField

`public java.util.List` **`getHearingField`**`(AgentDataAccessInterface model)`

This method returns a list of all fields heard by this agent

Parameters:

model - The associated model

Returns:

A list of 2D coordinates

---

getListCoord

`public java.util.List` **`getListCoord`**`()`

This method returns the list of coordinates where is represented this agent on the view

Returns:

A list of coordinates

---

getPanicLevel

`public int` **`getPanicLevel`**`()`

Getter for the panic level attribute

Returns:

The agent's panic level

getScreamingAbility

`public int` **`getScreamingAbility`**`()`

Getter for the screamingAbility attribute

Returns:

The agent's screaming ability

---

getSpeedLevel

`public int` **`getSpeedLevel`**`()`

Getter for the speedLevel attribute

Returns:

The agent's speed level

---

getVisionAbility

`public int` **`getVisionAbility`**`()`

Getter for the visionAbility attribute

Returns:

The agent's vision ability

---

getVisionField

`public java.util.List` **`getVisionField`**`(AgentDataAccessInterface model)`

This method returns a list of all fields seen by this agent

Parameters:

model - The associated model

Returns:

A list of 2D coordinates

---

hearScream

`public void` **`hearScream`**`()`

It defines what this people should do when he's hearing screams

---

isWarned

`public boolean **isWarned**()`

Tells if this people is in warn state or not

Returns:

A boolean telling if the agent is in warning state

---

orientation2D

`public double **orientation2D**()`

---

saySomething

`public void **saySomething**(java.lang.String something)`

This method writes a given message to the standard output, prefixed with this agent's name Useful for debug mode

Parameters:

something - The message to display

---

setStoppable

`public void **setStoppable**(sim.engine.Stoppable stop)`

Setter on the stop attribute

Parameters:

stop - A {@link Stoppable} object associated to this agent

---

step

`public void **step**(sim.engine.SimState state)`

---

toString

`public java.lang.String **toString**()`

Overrides:

toString in class java.lang.Object

Agents

Class Vision

```
java.lang.Object
    |
    +--Agents.Vision
```

< Constructors >

public class **Vision**
extends java.lang.Object

Constructors

Vision

```
public  Vision()
```

Package Components

Class Summary

Door

Exit

Shape

Space

Wall

Components

Class Door

```
java.lang.Object
    |
    +--Shape
         |
         +--Components.Door
```

All Implemented Interfaces:
sim.portrayal.Oriented2D

< Constructors > < Methods >

---

public class **Door**
extends Shape
implements sim.portrayal.Oriented2D

Constructors

Door

```
public  Door(java.lang.String beginX,
             java.lang.String beginY,
             java.lang.String endX,
             java.lang.String endY,
             java.lang.String direction)
```

Constructor

Parameters:

beginX - Where the door begins, in X-coordinate
beginY - Where the door begins, in Y-coordinate
endX - Where the door ends, in X-coordinate
endY - Where the door ends, in Y-coordinate
direction - The pointed direction

Methods

getDoorDirection

```
public Constants.Direction getDoorDirection()
```

Getter for the attribute doorDirection

Returns:

The door pointed direction

---

orientation2D

```
public double orientation2D()
```

---

toString

```
public java.lang.String toString()
```

The standard toString method

Overrides:

toString in class Shape

Components

Class Exit

```
java.lang.Object
    |
    +--Shape
        |
        +--Components.Exit
```

< Constructors >

---

public class **Exit**
extends Shape

Constructors

Exit

```
public  Exit(java.lang.String beginX,
             java.lang.String beginY,
             java.lang.String endX,
             java.lang.String endY)
```

Components

Class Shape

```
java.lang.Object
    |
    +--Components.Shape
```

Direct Known Subclasses:
Door, Exit, Wall

< Constructors > < Methods >

---

public abstract class **Shape**
extends java.lang.Object

Constructors

Shape

```
public  Shape(java.lang.String beginX,
              java.lang.String beginY,
              java.lang.String endX,
              java.lang.String endY)
```

Constructor

Parameters:

beginX - Where the shape begins, in X-coordinate
beginY - Where the shape begins, in Y-coordinate
endX - Where the shape ends, in X-coordinate
endY - Where the shape ends, in Y-coordinate

Methods

getBeginX

```
public int getBeginX()
```

Getter for beginX attribute

Returns:

Where the shape begins, as a x coordinate

---

getBeginY

```
public int getBeginY()
```

Getter for beginY attribute

Returns:

Where the shape begins, as a y coordinate

---

getDirection

```
public Constants.Direction getDirection()
```

---

getEndX

```
public int getEndX()
```

Getter for endX attribute

Returns:

Where the shape ends, as a x coordinate

getEndY

`public int` **`getEndY`**`()`

Getter for endY attribute

Returns:

Where the shape ends, as a y coordinate

---

getListCoord

`public java.util.List` **`getListCoord`**`()`

This method returns the list of coordinates where is represented this shape on the view

Returns:

A list of coordinates

---

toString

`public java.lang.String` **`toString`**`()`

Overrides:

toString in class java.lang.Object

Components

Class Space

```
java.lang.Object
    |
    +--Components.Space
```

< Constructors >

---

public class **Space**
extends java.lang.Object

Constructors

Space

```
public  Space()
```

Components

Class Wall

```
java.lang.Object
    |
    +--Shape
        |
        +--Components.Wall
```

< Constructors >

---

public class **Wall**
extends Shape

Constructors

Wall

```
public  Wall(java.lang.String beginX,
             java.lang.String beginY,
             java.lang.String endX,
             java.lang.String endY)
```

Package Model

Interface Summary

AgentDataAccessInterface

This is an interface that the model should implement It describes every method which can be called by an agent

Class Summary

Model

MyDisplay2D

View

Model

Interface AgentDataAccessInterface

< Methods >

---

public interface **AgentDataAccessInterface**

This is an interface that the model should implement It describes every method which can be called by an agent

Methods

addToGrid

```
public void addToGrid(java.util.List coords,
                      java.lang.Object obj)
```

This method adds the given {@link Object} to the grid, with it given coordinates If there's already a component to one the given coordinates, the existing {@link Object} will be erased

Parameters:

coords - The {@link Object}'s coordinates
obj - The {@link Object} to add

---

addToGridIfEmpty

```
public void addToGridIfEmpty(java.util.List coords,
                             java.lang.Object obj)
```

This method adds the given {@link Object} to the grid, with it given coordinates It won't erase a component if it already exists on a given coordinate

Parameters:

coords - The {@link Object}'s coordinates
obj - The {@link Object} to add

---

canHearTheFire

```
public boolean canHearTheFire(People ppl)
```

This method tells if the given {@link People} can hear a {@link Fire} somewhere

Parameters:

ppl - The observer {@link People}

Returns:

A boolean telling if the given {@link People} can hear a {@link Fire} somewhere

---

canMakeOneStepFront

```
public boolean canMakeOneStepFront(People ppl)
```

This method tells if a given {@link People} can make one step front or not

Parameters:

ppl - The subject {@link People}

Returns:

A boolean telling the given {@link People} can make one step front

---

canMakeOneStepTo

```
public boolean canMakeOneStepTo(Constants.Direction direction,
                                People ppl)
```

This method tells if a given {@link People} can make one step to a given {@link Direction} or not

Parameters:

direction - The direction the agent is supposed to go to
ppl - The subject {@link People}

Returns:

A boolean telling the given {@link People} can make one step to a given {@link Direction}

canSeeAnExit

`public Exit canSeeAnExit(People ppl)`

It checks if the given {@link People} can see an {@link Exit} or not

Parameters:

ppl - The subject of the action

Returns:

Either the seeable {@link Exit} object, either null value if no {@link Exit} can be seen

---

canSeeTheFire

`public boolean canSeeTheFire(People ppl)`

This method tells if the given {@link People} can see a {@link Fire} somewhere

Parameters:

ppl - The observer {@link People}

Returns:

A boolean telling if the given {@link People} can see a {@link Fire} somewhere

---

computeHearingField

`public java.util.List computeHearingField(People ppl)`

This method gives a list of coordinates that are what the given {@link People} can hear

Parameters:

ppl - The given {@link People}

Returns:

A list of coordinates

---

computeVisionField

`public java.util.List computeVisionField(People ppl)`

This method gives a list of coordinates that are what the given {@link People} can see

Parameters:

ppl - The given {@link People}

Returns:

A list of coordinates

---

getAudiblePeople

`public java.util.List` **`getAudiblePeople`**`(People ppl)`

It returns all people that the given {@link People} can hear, according to its abilities

Parameters:

ppl - The given {@link People}

Returns:

A list of all audible {@link People}

---

getClosestAudibleFire

`public Fire` **`getClosestAudibleFire`**`(People ppl)`

It returns the closest {@link Fire} that the given {@link People} can hear

Parameters:

ppl - The observer {@link People}

Returns:

The closest audible {@link Fire}, or null value if there's not

---

getClosestVisibleFire

`public Fire` **`getClosestVisibleFire`**`(People ppl)`

It returns the closest {@link Fire} that the given {@link People} can see

Parameters:

ppl - The observer {@link People}

Returns:

The closest visible {@link Fire}, or null value if there's not

---

getVisibleDoors

`public java.util.List` **`getVisibleDoors`**`(People ppl)`

It returns all people that the given {@link Door} can see, according to its abilities

Parameters:

ppl - The given {@link Door}

Returns:

A list of all seeable {@link Door}

---

getVisiblePeople

```
public java.util.List getVisiblePeople(People ppl)
```

It returns all people that the given {@link People} can see, according to its abilities

Parameters:

ppl - The given {@link People}

Returns:

A list of all seeable {@link People}

---

removeFromGrid

```
public boolean removeFromGrid(java.util.List coords,
                              java.lang.Object obj)
```

This method removes the given {@link Object} from the grid, located at the given coordinates

Parameters:

coords - The given coordinates
obj - The given {@link Object} to remove

Returns:

It returns false if the given {@link Object} was nowhere on the grid, or true

---

someoneScreams

```
public void someoneScreams(People ppl)
```

This method is invoked when a people is screaming Then, it notifies every people who is in the scream's scope

Parameters:

ppl - The people who is screaming

Model

Class Model

```
java.lang.Object
    |
    +--sim.engine.SimState
        |
        +--Model.Model
```

All Implemented Interfaces:

AgentDataAccessInterface, java.io.Serializable

< Constructors > < Methods >

public class **Model**
extends sim.engine.SimState
implements AgentDataAccessInterface

Constructors

Model

```
public  Model(long seed)
```

Methods

addFire

```
public void addFire(sim.util.Int2D hearth)
```

---

addToGrid

```
public void addToGrid(java.util.List coords,
                      java.lang.Object obj)
```

---

addToGridIfEmpty

```
public void addToGridIfEmpty(java.util.List coords,
                             java.lang.Object obj)
```

---

canHearTheFire

```
public boolean canHearTheFire(People ppl)
```

---

canMakeOneStepFront

```
public boolean canMakeOneStepFront(People ppl)
```

---

canMakeOneStepTo

public boolean **canMakeOneStepTo**(Constants.Direction direction,
People ppl)

---

canSeeAnExit

public Exit **canSeeAnExit**(People ppl)

---

canSeeTheFire

public boolean **canSeeTheFire**(People ppl)

---

computeHearingField

public java.util.List **computeHearingField**(People ppl)

---

computeVisionField

public java.util.List **computeVisionField**(People ppl)

---

getAudiblePeople

public java.util.List **getAudiblePeople**(People ppl)

---

getClosestAudibleFire

public Fire **getClosestAudibleFire**(People ppl)

---

getClosestVisibleFire

`public Fire getClosestVisibleFire(People ppl)`

---

getDoorList

`public java.util.List getDoorList()`

Getter for the doorList attribute

Returns:

A list of {@link Door}

---

getExitList

`public java.util.List getExitList()`

Getter for the exitList attribute

Returns:

A list of {@link Exit}

---

getGrid

`public sim.field.grid.ObjectGrid2D getGrid()`

Getter for the grid attribute

Returns:

The grid attribute

---

getPeopleList

`public java.util.List getPeopleList()`

Getter for the peopleList attribute

Returns:

A list of {@link People}

---

getVisibleDoors

```
public java.util.List getVisibleDoors(People ppl)
```

---

getVisiblePeople

```
public java.util.List getVisiblePeople(People ppl)
```

---

getWallList

```
public java.util.List getWallList()
```

Getter for the wallList attribute

Returns:

A list of {@link Wall}

---

removeFromGrid

```
public boolean removeFromGrid(java.util.List coords,
                              java.lang.Object obj)
```

---

removeSpace

```
public void removeSpace()
```

---

someoneScreams

```
public void someoneScreams(People ppl)
```

---

start

```
public void start()
```

Overrides:

start in class sim.engine.SimState

Model

Class MyDisplay2D

```
java.lang.Object
    |
    +--java.awt.Component
        |
        +--java.awt.Container
            |
            +--javax.swing.JComponent
                |
                +--sim.display.Display2D
                    |
                    +--Model.MyDisplay2D
```

All Implemented Interfaces:

java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.swing.TransferHandler.HasGetTransferHandler, sim.display.Manipulating2D, sim.engine.Steppable

---

< Fields > < Constructors >

---

public class **MyDisplay2D**
extends sim.display.Display2D

Fields

simulation

```
public final sim.display.GUIState simulation
```

Constructors

MyDisplay2D

```
public  MyDisplay2D(double width,
                    double height,
                    sim.display.GUIState simulation)
```

Model

Class View

```
java.lang.Object
    |
    +--sim.display.GUIState
        |
        +--Model.View
```

< Fields > < Constructors > < Methods >

---

public class **View**
extends sim.display.GUIState

Fields

display

`public MyDisplay2D` **`display`**

---

displayFrame

`public javax.swing.JFrame` **`displayFrame`**

---

gridPortrayal

`public sim.portrayal.grid.ObjectGridPortrayal2D` **`gridPortrayal`**

Constructors

View

`public` **`View`**`(sim.engine.SimState state)`

Methods

getSimulationInspectedObject

`public java.lang.Object` **`getSimulationInspectedObject`**`()`

Overrides:

getSimulationInspectedObject in class sim.display.GUIState

---

init

`public void` **`init`**`(sim.display.Controller controller)`

Overrides:

init in class sim.display.GUIState

---

start

`public void` **`start`**`()`

Overrides:

start in class sim.display.GUIState

Package Util

Class Summary

Actions

Actions.Action

ClickablePortrayal

Constants

Constants.Direction

LogConsole

ReadXml

Utils

This class contains some useful methods All methods are declared as static as the Utils class isn't supposed to be implemented

---

Util

Class Actions

```
java.lang.Object
    |
    +--Util.Actions
```

< Constructors >

---

public class **Actions**
extends java.lang.Object

Constructors

Actions

```
public  Actions()
```

Util

Class Actions.Action

```
java.lang.Object
    |
    +--java.lang.Enum
        |
        +--Util.Actions.Action
```

All Implemented Interfaces:
java.io.Serializable, java.lang.Comparable

< Fields > < Methods >

---

public static final class **Actions.Action**
extends java.lang.Enum

Fields

ADD

`public static final Actions.Action` **`ADD`**

DRAW

`public static final Actions.Action` **`DRAW`**

FILE

`public static final Actions.Action` **`FILE`**

READ

`public static final Actions.Action` **`READ`**

Methods

valueOf

`public static Actions.Action` **`valueOf`**`(java.lang.String name)`

---

values

```
public static Util.Actions.Action[] values()
```

Util

Class ClickablePortrayal

```
java.lang.Object
    |
    +--sim.portrayal.SimplePortrayal2D
        |
        +--sim.portrayal.simple.RectanglePortrayal2D
            |
            +--Util.ClickablePortrayal
```

All Implemented Interfaces:
sim.portrayal.Portrayal2D

< Constructors > < Methods >

---

public class **ClickablePortrayal**
extends sim.portrayal.simple.RectanglePortrayal2D

Constructors

ClickablePortrayal

```
public  ClickablePortrayal(java.awt.Color color)
```

Methods

handleMouseEvent

```
public boolean handleMouseEvent(sim.display.GUIState guistate,
                                sim.display.Manipulating2D manipulating,
                                sim.portrayal.LocationWrapper wrapper,
                                java.awt.event.MouseEvent event,
                                sim.portrayal.DrawInfo2D
fieldPortrayalDrawInfo,
                                int type)
```

Overrides:

handleMouseEvent in class sim.portrayal.SimplePortrayal2D

Util

Class Constants

```
java.lang.Object
    |
    +--Util.Constants
```

< Fields > < Constructors >

---

public class **Constants**
extends java.lang.Object

Fields

AGENT_HIGH_SPEED

`public static final int` **`AGENT_HIGH_SPEED`**

AGENT_NORMAL_SPEED

`public static final int` **`AGENT_NORMAL_SPEED`**

AGENT_SLOW_SPEED

`public static final int` **`AGENT_SLOW_SPEED`**

AGENT_VERY_HIGH_SPEED

`public static final int` **`AGENT_VERY_HIGH_SPEED`**

FRAME_HEIGHT

`public static final int` **`FRAME_HEIGHT`**

FRAME_WIDTH

`public static final int` **`FRAME_WIDTH`**

GRID_HEIGHT

public static final int **GRID_HEIGHT**

GRID_TO_FRAME

public static final int **GRID_TO_FRAME**

GRID_WIDTH

public static final int **GRID_WIDTH**

MAX_ABILITY

public static final int **MAX_ABILITY**

MAX_PANIC

public static final int **MAX_PANIC**

MIN_ABILITY

public static final int **MIN_ABILITY**

NUM_STEP_FIRE_SPREAD

public static final int **NUM_STEP_FIRE_SPREAD**

STRONG_PANIC

public static final int **STRONG_PANIC**

Constructors

Constants

public **Constants**()

Util

Class Constants.Direction

```
java.lang.Object
    |
    +--java.lang.Enum
        |
        +--Util.Constants.Direction
```

All Implemented Interfaces:
java.io.Serializable, java.lang.Comparable

< Fields > < Methods >

---

public static final class **Constants.Direction**
extends java.lang.Enum

Fields

EAST

`public static final Constants.Direction` **`EAST`**

NORTH

`public static final Constants.Direction` **`NORTH`**

SOUTH

`public static final Constants.Direction` **`SOUTH`**

UNKNOWN

`public static final Constants.Direction` **`UNKNOWN`**

WEST

`public static final Constants.Direction` **`WEST`**

Methods

valueOf

```
public static Constants.Direction valueOf(java.lang.String name)
```

---

values

```
public static Util.Constants.Direction[] values()
```

Util

Class LogConsole

```
java.lang.Object
    |
    +--Util.LogConsole
```

< Constructors > < Methods >

---

public class **LogConsole**
extends java.lang.Object

Constructors

LogConsole

```
public  LogConsole()
```

Methods

print

```
public static void print(java.lang.String message,
                         java.lang.String type,
                         java.lang.String resource)
```

Util

Class ReadXml

```
java.lang.Object
    |
    +--Util.ReadXml
```

< Constructors > < Methods >

---

public class **ReadXml**
extends java.lang.Object

Constructors

ReadXml

```
public  ReadXml()
```

Methods

getDoorList

```
public static java.util.List getDoorList()
```

---

getExitList

```
public static java.util.List getExitList()
```

---

getPeopleList

```
public static java.util.List getPeopleList()
```

---

getWallList

```
public static java.util.List getWallList()
```

readXmlFile

```
public static void readXmlFile(java.lang.String filename)
```

Util

Class Utils

```
java.lang.Object
    |
    +--Util.Utils
```

< Constructors > < Methods >

---

public class **Utils**
extends java.lang.Object

This class contains some useful methods All methods are declared as static as the {@link Utils} class isn't supposed to be implemented

Constructors

Utils

```
public  Utils()
```

Methods

areDirectionsOpposite

```
public static boolean areDirectionsOpposite(Constants.Direction d1,
                                            Constants.Direction d2)
```

This method tells if the two given {@link Direction} are opposite

Parameters:

d1 - The first {@link Direction}
d2 - The second {@link Direction}

Returns:

A boolean telling if the two given directions are opposite

---

getDirectionFromCoordinates

```
public static Constants.Direction getDirectionFromCoordinates(People people,
                                                               sim.util.Int2D
c)
```

It tells in which direction is a given position in comparison with a given {@link People}

Parameters:

people - The watcher
c - The targeted position

Returns:

The direction where is the given coordinate from the {@link People}'s point of view

---

getOppositeDirection

```
public static Constants.Direction getOppositeDirection(Constants.Direction d)
```

This method gives the opposite {@link Direction} based on a given {@link Direction}

Parameters:

d - The based {@link Direction}

Returns:

The d's opposite {@link Direction}

---

getRandomMasonValue

```
public static int getRandomMasonValue(AgentDataAccessInterface s,
                                      int val1,
                                      int val2)
```

It simply generates a random value according to Mason's way of working

Parameters:

s - The {@link AgentDataAccessInterface} to which the simulation is linked
val1 - The min or the max
val2 - The min or the max

Returns:

A random number between val1 and val2, both included

---

getRandomMasonValue

```
public static int getRandomMasonValue(sim.engine.SimState s,
                                       int val1,
                                       int val2)
```

It simply generates a random value according to Mason's way of working

Parameters:

s - The {@link SimState} to which the simulation is linked
val1 - The min or the max
val2 - The min or the max

Returns:

A random number between val1 and val2, both included

---

isCoordInGrid

```
public static boolean isCoordInGrid(int x,
                                    int y)
```

This method tells if a given coordinate is on the grid or outside

Parameters:

x - The x-coordinate
y - The y-coordinate

Returns:

A boolean telling if a given coordinate is on the grid

---

isCoordInGrid

```
public static boolean isCoordInGrid(sim.util.Int2D coord)
```

This method tells if a given coordinate is on the grid or outside

Parameters:

coord - The given coordinate

Returns:

A boolean telling if a given coordinate is on the grid

INDEX