

Monde des cubes

0.1

Generated by Doxygen 1.5.7.1

Mon Apr 27 20:16:17 2009

Contents

1	Class Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	AucuneSurcharge Class Reference	7
4.1.1	Detailed Description	7
4.1.2	Member Function Documentation	7
4.1.2.1	pasSurcharges	7
4.1.2.2	serontPasSurcharges	8
4.1.2.3	verifier	8
4.2	compareEcoAgentID Class Reference	9
4.2.1	Detailed Description	9
4.3	Cube Class Reference	10
4.3.1	Detailed Description	11
4.3.2	Constructor & Destructor Documentation	11
4.3.2.1	Cube	11
4.3.2.2	Cube	11
4.3.2.3	~Cube	11
4.3.3	Member Function Documentation	11
4.3.3.1	agresser	11
4.3.3.2	initialiser	11
4.4	EcoAgent Class Reference	12
4.4.1	Detailed Description	13

4.4.2	Constructor & Destructor Documentation	13
4.4.2.1	EcoAgent	13
4.4.3	Member Function Documentation	13
4.4.3.1	agresser	13
4.4.3.2	getEtat	14
4.4.3.3	getId	14
4.4.3.4	getPositionCourante	14
4.4.3.5	getPositionFinale	14
4.4.3.6	getVisite	14
4.4.3.7	initialiser	15
4.4.3.8	setEtat	15
4.4.3.9	setId	15
4.4.3.10	setPositionCourante	15
4.4.3.11	setPositionFinale	15
4.4.3.12	setVisite	16
4.5	EcoAgentID Class Reference	17
4.5.1	Detailed Description	17
4.5.2	Constructor & Destructor Documentation	18
4.5.2.1	EcoAgentID	18
4.5.2.2	~EcoAgentID	18
4.5.3	Member Function Documentation	18
4.5.3.1	getId	18
4.5.3.2	operator!=	18
4.5.3.3	operator<	18
4.5.3.4	operator==	18
4.6	ExceptionEcoAgentDejaEnregistre Class Reference	19
4.6.1	Detailed Description	19
4.7	PlateformeEcoResolution Class Reference	20
4.7.1	Detailed Description	21
4.7.2	Constructor & Destructor Documentation	21
4.7.2.1	PlateformeEcoResolution	21
4.7.2.2	~PlateformeEcoResolution	21
4.7.3	Member Function Documentation	21
4.7.3.1	addEcoAgent	21
4.7.3.2	addRegle	21
4.7.3.3	getEcoAgent	22

4.7.3.4	getEcoAgentAuDessus	22
4.7.3.5	getEcoAgents	22
4.7.3.6	getEcoAgents	22
4.7.3.7	getRegles	23
4.7.3.8	initialiser	23
4.7.3.9	nombreEcoAgentAuDessus	23
4.7.3.10	resoudre	23
4.7.3.11	sontSatisfaits	23
4.7.3.12	verifierCoherence	24
4.7.4	Member Data Documentation	24
4.7.4.1	ecoagents	24
4.7.4.2	regles	24
4.8	PlateformeMondeDesCubes Class Reference	25
4.8.1	Detailed Description	26
4.8.2	Member Function Documentation	26
4.8.2.1	distanceFinaleATable	26
4.8.2.2	getNombreDeCubes	26
4.8.2.3	getTableID	26
4.8.2.4	initialiser	27
4.8.2.5	obtenirCubePrioritaire	27
4.8.2.6	obtenirGeneur	27
4.8.2.7	resoudre	27
4.8.2.8	setCubeID	27
4.8.2.9	setNombreDeCubes	28
4.8.2.10	setPositionCourante	28
4.8.2.11	setPositionFinale	28
4.8.2.12	setTableIdentifiant	28
4.9	Regle Class Reference	29
4.9.1	Detailed Description	29
4.9.2	Member Function Documentation	29
4.9.2.1	verifier	29
4.10	reliesATable Class Reference	30
4.10.1	Detailed Description	30
4.11	Singleton< T > Class Template Reference	31
4.11.1	Detailed Description	31
4.11.2	Constructor & Destructor Documentation	31

4.11.2.1	Singleton	31
4.11.2.2	~Singleton	31
4.11.3	Member Function Documentation	32
4.11.3.1	getInstance	32
4.11.3.2	kill	32
4.12	Table Class Reference	33
4.12.1	Detailed Description	33
4.12.2	Constructor & Destructor Documentation	34
4.12.2.1	Table	34
4.12.2.2	Table	34
4.12.2.3	~Table	34
4.12.3	Member Function Documentation	34
4.12.3.1	agresser	34
5	File Documentation	35
5.1	trunk/include/aucuneSurcharge.hpp File Reference	35
5.1.1	Detailed Description	35
5.2	trunk/include/compareEcoAgentID.hpp File Reference	36
5.2.1	Detailed Description	36
5.3	trunk/include/cube.hpp File Reference	37
5.3.1	Detailed Description	37
5.4	trunk/include/ecoAgent.hpp File Reference	38
5.4.1	Detailed Description	38
5.5	trunk/include/ecoAgentID.hpp File Reference	39
5.5.1	Detailed Description	39
5.6	trunk/include/etat.hpp File Reference	40
5.6.1	Detailed Description	40
5.7	trunk/include/ExceptionEcoAgentDejaEnregistre.hpp File Reference	41
5.7.1	Detailed Description	41
5.8	trunk/include/plateformeEcoResolution.hpp File Reference	42
5.8.1	Detailed Description	42
5.9	trunk/include/plateformeMondeDesCubes.hpp File Reference	43
5.9.1	Detailed Description	43
5.10	trunk/include/regle.hpp File Reference	44
5.10.1	Detailed Description	44
5.11	trunk/include/reliesATable.hpp File Reference	45
5.11.1	Detailed Description	45

5.12	trunk/include singleton.hpp File Reference	46
5.12.1	Detailed Description	46
5.13	trunk/include/table.hpp File Reference	47
5.13.1	Detailed Description	47

Chapter 1

Class Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

compareEcoAgentID	9
EcoAgent	12
Cube	10
Table	33
EcoAgentID	17
ExceptionEcoAgentDejaEnregistre	19
PlateformeEcoResolution	20
PlateformeMondeDesCubes	25
Regle	29
AucuneSurcharge	7
reliesATable	30
Singleton< T >	31
Singleton< PlateformeMondeDesCubes >	31
PlateformeMondeDesCubes	25

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

AucuneSurcharge (Cette classe contient les fonctions testant si les cubes ne portent pas plus de un cube)	7
compareEcoAgentID (Structure contenant la redefinition de l'operateur de comparaison d'EcoAgentID Structure contenant la redefinition de l'operateur de comparaison d'EcoAgentID pour le transformer en cle unique d'une map)	9
Cube (Classe derivee de la classe EcoAgent designant un Cube)	10
EcoAgent (Classe abstraite qui permet de creer un eco-agent)	12
EcoAgentID (Identifiant unique d'un eco-agent)	17
ExceptionEcoAgentDejaEnregistre (Exception lancee dans le cadre de l'ajout d'un EcoAgent deja enregistre dans une PlateformeEcoResolution)	19
PlateformeEcoResolution (Classe representant une plateforme d'eco-resolution abstraite)	20
PlateformeMondeDesCubes (Classe representant une plateforme d'eco-resolution appliquee au monde des cubes)	25
Regle (Cette classe abstraite est une squelette pour une regle)	29
reliesATable (Cette classe contient les fonctions testant si les cubes sont lies a la table)	30
Singleton< T > (Template de classe permettant de rendre une classe instanciable une seule fois)	31
Table (Classe derivee de la classe EcoAgent designant le Support sur lequel vont etre poses les cubes)	33

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

trunk/include/ aucuneSurcharge.hpp (Verification de la regle : est-ce que chaque cube porte au plus un cube?)	35
trunk/include/ compareEcoAgentID.hpp (Comparaison d'EcoAgentID)	36
trunk/include/ cube.hpp (Implementation du module cube qui est un derive d'un EcoAgent) . . .	37
trunk/include/ ecoAgent.hpp (Mise en place de la classe abstraite EcoAgent)	38
trunk/include/ ecoAgentID.hpp (Implementation de la classe EcoAgentID)	39
trunk/include/ etat.hpp (Enumeration des etats possibles des eco-agents)	40
trunk/include/ ExceptionEcoAgentDejaEnregistre.hpp (Exception lancee lorsqu'on enregistre dans la plateforme un agent qui a deja ete enregistre)	41
trunk/include/ plateformeEcoResolution.hpp (Plateforme abstraite d'eco-resolution)	42
trunk/include/ plateformeMondeDesCubes.hpp (Plateforme d'eco-resolution appliquee au monde des cubes)	43
trunk/include/ regle.hpp (Squelette d'une regle pour une plateforme d'eco-resolution)	44
trunk/include/ reliesATable.hpp (Verification de la regle : est-ce que tous les cubes sont et seront relies directement ou indirectement a la table?)	45
trunk/include/ singleton.hpp (Implementation du design pattern singleton)	46
trunk/include/ table.hpp (Implementation du module table qui est un derive d'un EcoAgent) . . .	47

Chapter 4

Class Documentation

4.1 AucuneSurcharge Class Reference

Cette classe contient les fonctions testant si les cubes ne portent pas plus de un cube.

```
#include <aucuneSurcharge.hpp>
```

Inherits [Regle](#).

Collaboration diagram for AucuneSurcharge:

Public Member Functions

- void [initialiser](#) ()
Suite d'operations realisees pour initialiser la regle.
- bool [verifier](#) ()
Verification de la regle generale : pour l'ensemble des cubes, sont-ils et seront-ils surcharges?
- bool [pasSurcharges](#) ()
Verification d'un element de la regle generale: les cubes sont-ils surcharges?
- bool [serontPasSurcharges](#) ()
Verification d'un element de la regle generale: les cubes seront-ils surcharges?

4.1.1 Detailed Description

Cette classe contient les fonctions testant si les cubes ne portent pas plus de un cube.

4.1.2 Member Function Documentation

4.1.2.1 bool AucuneSurcharge::pasSurcharges ()

Verification d'un element de la regle generale: les cubes sont-ils surcharges?

Returns:

true si le cas est verifiee, false sinon

4.1.2.2 bool AucuneSurcharge::serontPasSurcharges ()

Verification d'un element de la regle generale: les cubes seront-ils surcharges?

Returns:

true si le cas est verifiee, false sinon

4.1.2.3 bool AucuneSurcharge::verifier () [virtual]

Verification de la regle generale : pour l'ensemble des cubes, sont-ils et seront-ils surcharges?

Returns:

true si la regle est verifiee, false sinon

Implements [Regle](#).

The documentation for this class was generated from the following files:

- [trunk/include/aucuneSurcharge.hpp](#)
- [trunk/src/aucuneSurcharge.cpp](#)

4.2 compareEcoAgentID Class Reference

Structure contenant la redefinition de l'operateur de comparaison d'EcoAgentID Structure contenant la redefinition de l'operateur de comparaison d'EcoAgentID pour le transformer en cle unique d'une map.

```
#include <compareEcoAgentID.hpp>
```

4.2.1 Detailed Description

Structure contenant la redefinition de l'operateur de comparaison d'EcoAgentID Structure contenant la redefinition de l'operateur de comparaison d'EcoAgentID pour le transformer en cle unique d'une map.

The documentation for this class was generated from the following file:

- trunk/include/[compareEcoAgentID.hpp](#)

4.3 Cube Class Reference

Classe derivee de la classe [EcoAgent](#) designant un [Cube](#).

```
#include <cube.hpp>
```

Inherits [EcoAgent](#).

Collaboration diagram for Cube:

Public Member Functions

- [Cube](#) ()
Constructeur.
- [Cube](#) (const [EcoAgentID](#) &id)
Constructeur.
- [~Cube](#) ()
Destructeur.
- void [rechercherFuite](#) ()
Suite d'operations realisees par le cube lorsqu'il cherche a fuir.
- void [rechercherSatisfaction](#) ()
Suite d'operations realisees par le cube lorsqu'il cherche a se satisfaire.
- void [agresser](#) ([EcoAgent](#) &a)
Suite d'operations realisees lorsque le cube agresse un autre eco-agent.
- void [estAgresse](#) ()
Suite d'operations realisees par le cube lorsqu'il est agressee.
- void [faireFuite](#) ()
Suite d'operations realisees par le cube lorsqu'il fuit.
- void [faireSatisfaction](#) ()
Suite d'operations realisees par le cube lorsqu'il se satisfait.
- void [initialiser](#) ()
Initialisation de l'etat du cube.
- void [agir](#) ()
Suite d'operations realisees par le cube lorsqu'il agit.

Friends

- ostream & [operator<<](#) (ostream &f, const [Cube](#) &c)
Suite d'operations permettant d'afficher des informations sur le cube.

4.3.1 Detailed Description

Classe derivee de la classe [EcoAgent](#) designant un [Cube](#).

4.3.2 Constructor & Destructor Documentation

4.3.2.1 `Cube::Cube ()`

Constructeur.

Constructeur de la classe [Cube](#) par default. Le cube recevra un [EcoAgentID](#) automatiquement genere.

4.3.2.2 `Cube::Cube (const EcoAgentID & id)`

Constructeur.

Constructeur de la classe [Cube](#) avec un [EcoAgentID](#) specifique

Parameters:

id : identifiant unique que l'agent se verra attribuer

4.3.2.3 `Cube::~~Cube ()`

Destructeur.

Destructeur de la classe [Cube](#)

4.3.3 Member Function Documentation

4.3.3.1 `void Cube::agresser (EcoAgent & a) [virtual]`

Suite d'operations realisees lorsque le cube agresse un autre eco-agent.

Parameters:

a : [EcoAgent](#) a agresser

Implements [EcoAgent](#).

4.3.3.2 `void Cube::initialiser () [virtual]`

Initialisation de l'etat du cube.

Cette methode permet d'initialiser l'etat du cube en prenant en compte sa position courante et sa position finale. Par exemple, si la position courante correspond a la position finale, cette methode initialisera l'Etat a "satisfait"

Implements [EcoAgent](#).

The documentation for this class was generated from the following files:

- [trunk/include/cube.hpp](#)
- [trunk/src/cube.cpp](#)

4.4 EcoAgent Class Reference

Classe abstraite qui permet de creer un eco-agent.

```
#include <ecoAgent.hpp>
```

Inherited by [Cube](#), and [Table](#).

Collaboration diagram for EcoAgent:

Public Member Functions

- [EcoAgent](#) ()
Constructeur de la classe [EcoAgent](#).
- [Etat](#) [getEtat](#) () const
Obtention de l'Etat.
- void [setEtat](#) (const [Etat](#) e)
Changement de l'Etat.
- [EcoAgentID](#) * [getId](#) () const
Obtention de l'EcoAgentID.
- void [setId](#) (const [EcoAgentID](#) &id)
Changement de l'EcoAgentID.
- bool [getVisite](#) () const
Obtention du flag visite.
- void [setVisite](#) (const bool b)
Changement du flag visite.
- void [setPositionCourante](#) (const [EcoAgentID](#) &pos)
Changement de la position courante de l'agent.
- [EcoAgentID](#) * [getPositionCourante](#) () const
Permet d'obtenir la position courante de l'eco-agent.
- void [setPositionFinale](#) (const [EcoAgentID](#) &pos)
Changement de la position finale de l'agent.
- [EcoAgentID](#) * [getPositionFinale](#) () const
Permet d'obtenir la position courante de l'eco-agent.
- virtual void [initialiser](#) ()=0
Initialisation de l'EcoAgent.
- virtual void [rechercherFuite](#) ()=0
Suite d'operations realisees par l'agent lorsqu'il cherche a fuir.

- virtual void [rechercherSatisfaction](#) ()=0
Suite d'operations realisees par l'agent lorsqu'il cherche a se satisfaire.
- virtual void [agresser](#) ([EcoAgent](#) &a)=0
Suite d'operations realisees lorsqu'un agent1 agresse un agent2.
- virtual void [estAgresse](#) ()=0
Suite d'operations realisees par un agent agresse.
- virtual void [faireFuite](#) ()=0
Suite d'operations realisees par l'agent lorsqu'il est agresse.
- virtual void [faireSatisfaction](#) ()=0
Suite d'operations realisees par l'agent lorsqu'il se satisfait.
- virtual void [agir](#) ()=0
Suite d'operations realisees par l'agent lorsqu'il agit.

Friends

- ostream & [operator<<](#) (ostream &, const [EcoAgent](#) &)
Cette methode permet d'afficher des informations sur un [EcoAgent](#).

4.4.1 Detailed Description

Classe abstraite qui permet de creer un eco-agent.

Classe abstraite qui permet de creer un eco-agent. Tout eco-agent doit herite de cette classe.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 [EcoAgent::EcoAgent](#) ()

Constructeur de la classe [EcoAgent](#).

Ce constructeur initialise la valeur des champs par default : initialisation d'un identifiant, positionCourante a NULL, positionFinale a NULL

4.4.3 Member Function Documentation

4.4.3.1 virtual void [EcoAgent::agresser](#) ([EcoAgent](#) & *a*) [pure virtual]

Suite d'operations realisees lorsqu'un agent1 agresse un agent2.

Parameters:

a : [EcoAgent](#) a agresser

Implemented in [Cube](#), and [Table](#).

4.4.3.2 Etat `EcoAgent::getEtat () const`

Obtention de l'Etat.

Methode qui permet d'obtenir l'attribut prive Etat

Returns:

l'Etat de l'objet [EcoAgent](#)

4.4.3.3 `EcoAgentID * EcoAgent::getId () const`

Obtention de l'EcoAgentID.

Methode qui permet d'obtenir l'attribut prive [EcoAgentID](#)

Returns:

l'EcoAgentID de l'objet [EcoAgent](#)

4.4.3.4 `EcoAgentID * EcoAgent::getPositionCourante () const`

Permet d'obtenir la position courante de l'eco-agent.

Methode qui permet d'obtenir l'EcoAgentID de l'eco-agent sur lequel est placé l'eco-agent

Returns:

[EcoAgentID](#) de l'EcoAgent avec lequel cet [EcoAgent](#) est en relation

4.4.3.5 `EcoAgentID * EcoAgent::getPositionFinale () const`

Permet d'obtenir la position courante de l'eco-agent.

Methode qui permet d'obtenir l'EcoAgentID de l'eco-agent sur lequel est placé l'eco-agent

Returns:

[EcoAgentID](#) de l'EcoAgent avec lequel cet [EcoAgent](#) doit se trouver en relation a la fin de la resolution

4.4.3.6 `bool EcoAgent::getVisite () const`

Obtention du flag visite.

Methode qui permet d'obtenir l'attribut prive visite

Returns:

l'EcoAgentID de l'objet [EcoAgent](#)

4.4.3.7 virtual void EcoAgent::initialiser () [pure virtual]

Initialisation de l'EcoAgent.

Suite d'operations realisees pour initialiser un [EcoAgent](#) avant l'eco-resolution

Implemented in [Cube](#), and [Table](#).

4.4.3.8 void EcoAgent::setEtat (const Etat *e*)

Changement de l'Etat.

Methode qui permet de modifier l'attribut prive Etat

Parameters:

e : Etat que recevra l'EcoAgent

4.4.3.9 void EcoAgent::setId (const EcoAgentID & *id*)

Changement de l'EcoAgentID.

Methode qui permet de modifier l'attribut prive [EcoAgentID](#)

Parameters:

id : id que recevra l'EcoAgent

4.4.3.10 void EcoAgent::setPositionCourante (const EcoAgentID & *pos*)

Changement de la position courante de l'agent.

Methode qui permet de changer la position courante de l'eco-agent

Parameters:

pos : [EcoAgentID](#) de la nouvelle position

4.4.3.11 void EcoAgent::setPositionFinale (const EcoAgentID & *pos*)

Changement de la position finale de l'agent.

Methode qui permet de changer la position finale de l'eco-agent, c'est a dire, l'identifiant de l'eco-agent sur lequel l'eco-agent doit se trouver a la fin de la resolution

Parameters:

pos : [EcoAgentID](#) de la nouvelle position

4.4.3.12 void EcoAgent::setVisite (const bool *b*)

Changement du flag visite.

Methode qui permet de modifier l'attribut prive visite

Parameters:

b : valeur du flag

The documentation for this class was generated from the following files:

- [trunk/include/*ecoAgent*.hpp](#)
- [trunk/src/*ecoAgent*.cpp](#)

4.5 EcoAgentID Class Reference

Identifiant unique d'un eco-agent.

```
#include <ecoAgentID.hpp>
```

Public Member Functions

- [EcoAgentID](#) ()
Constructeur.
- [~EcoAgentID](#) ()
Destructeur.
- int [getId](#) () const
Obtention de l'identifiant.
- bool [operator==](#) (const [EcoAgentID](#) &) const
Comparaison de [EcoAgentID](#).
- bool [operator!=](#) (const [EcoAgentID](#) &) const
Comparaison de [EcoAgentID](#).
- bool [operator<](#) (const [EcoAgentID](#) &) const
Comparaison de [EcoAgentID](#).

Static Public Member Functions

- static int [getNombreDeGeneration](#) ()
Obtention du nombre de generation Methode statique qui permet d'obtenir le nombre de generations d'EcoAgentID.

Friends

- ostream & [operator<<](#) (ostream &, const [EcoAgentID](#) &)
Cette methode permet d'afficher des informations sur un [EcoAgent](#).

4.5.1 Detailed Description

Identifiant unique d'un eco-agent.

La classe [EcoAgentID](#) represente un identifiant unique d'un eco-agent. Il permet la generation automatique des identifiants

4.5.2 Constructor & Destructor Documentation

4.5.2.1 `EcoAgentID::EcoAgentID ()`

Constructeur.

Constructeur de la classe [EcoAgentID](#)

4.5.2.2 `EcoAgentID::~~EcoAgentID ()`

Destructeur.

Destructeur de la classe [EcoAgentID](#)

4.5.3 Member Function Documentation

4.5.3.1 `int EcoAgentID::getId () const`

Obtention de l'identifiant.

Methode qui retourne l'identifiant de l'eco-agent

4.5.3.2 `bool EcoAgentID::operator!= (const EcoAgentID & eid) const`

Comparaison de [EcoAgentID](#).

Methode qui permet de comparer deux [EcoAgentID](#)

4.5.3.3 `bool EcoAgentID::operator< (const EcoAgentID & eid) const`

Comparaison de [EcoAgentID](#).

Methode qui permet de comparer deux [EcoAgentID](#)

4.5.3.4 `bool EcoAgentID::operator== (const EcoAgentID & eid) const`

Comparaison de [EcoAgentID](#).

Methode qui permet de comparer deux [EcoAgentID](#)

The documentation for this class was generated from the following files:

- `trunk/include/ecoAgentID.hpp`
- `trunk/src/ecoAgentID.cpp`

4.6 ExceptionEcoAgentDejaEnregistre Class Reference

Exception lancee dans le cadre de l'ajout d'un [EcoAgent](#) deja enregistre dans une [PlateformeEcoResolution](#).

```
#include <ExceptionEcoAgentDejaEnregistre.hpp>
```

4.6.1 Detailed Description

Exception lancee dans le cadre de l'ajout d'un [EcoAgent](#) deja enregistre dans une [PlateformeEcoResolution](#).

The documentation for this class was generated from the following file:

- [trunk/include/ExceptionEcoAgentDejaEnregistre.hpp](#)

4.7 PlateformeEcoResolution Class Reference

classe representant une plateforme d'eco-resolution abstraite

```
#include <plateformeEcoResolution.hpp>
```

Inherited by [PlateformeMondeDesCubes](#).

Public Member Functions

- [PlateformeEcoResolution](#) ()
Constructeur.
- [~PlateformeEcoResolution](#) ()
Destructeur.
- [EcoAgent](#) * [getEcoAgent](#) (const [EcoAgentID](#) &id) const
Obtention d'un eco-agent.
- void [addEcoAgent](#) ([EcoAgent](#) &ea)
Ajout d'un eco-agent.
- void [addRegle](#) ([Regle](#) &r)
Ajout d'une nouvelle regle.
- list< [Regle](#) * > [getRegles](#) ()
Obtention de la liste des regles.
- bool [verifierCoherence](#) () const
Verification du respect des regles apres l'initialisation de la plateforme.
- bool [sontSatisfaits](#) () const
Methode qui verifie si tous les eco-agents sont satisfaits.
- [EcoAgentID](#) * [getEcoAgentAuDessus](#) (const [EcoAgentID](#) &id) const
Methode qui retourne l'EcoAgentID de l'EcoAgent au dessus.
- int [nombreEcoAgentAuDessus](#) (const [EcoAgentID](#) &id) const
Methode qui retourne le nombre d'EcoAgent au dessus de l'EcoAgent avec l'id specifie.
- list< [EcoAgentID](#) * > [getEcoAgents](#) (const [Etat](#) e) const
Methode qui retourne les EcoAgentID des EcoAgent possedant l'etat specifie.
- map< [EcoAgentID](#), [EcoAgent](#) &, [compareEcoAgentID](#) > [getEcoAgents](#) () const
Methode qui retourne tous les eco-agents.
- virtual void [initialiser](#) ()=0
Initialisation de la resolution.
- virtual void [resoudre](#) ()=0
Resolution du probleme par eco-resolution.

Protected Attributes

- map< [EcoAgentID](#), [EcoAgent](#) &, [compareEcoAgentID](#) > [ecoagents](#)
- list< [Regle](#) * > [regles](#)

4.7.1 Detailed Description

classe representant une plateforme d'eco-resolution abstraite

La classe gere les fonctionnalites basiques d'une plateforme d'eco-resolution.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 PlateformeEcoResolution::PlateformeEcoResolution ()

Constructeur.

Constructeur de la classe abstraite [PlateformeEcoResolution](#)

4.7.2.2 PlateformeEcoResolution::~~PlateformeEcoResolution ()

Destructeur.

Destructeur de la classe abstraite [PlateformeEcoResolution](#)

4.7.3 Member Function Documentation

4.7.3.1 void PlateformeEcoResolution::addEcoAgent (EcoAgent & *ea*)

Ajout d'un eco-agent.

Methode qui permet d'ajouter un eco-agent dans la plateforme

Exceptions:

[*ExceptionEcoAgentDejaEnregistre*](#) : lancee lorsqu'on enregistre un [EcoAgent](#) deja enregistre

Parameters:

ea : l'eco-agent a ajouter

4.7.3.2 void PlateformeEcoResolution::addRegle (Regle & *r*)

Ajout d'une nouvelle regle.

Methode qui permet d'ajouter une nouvelle regle dans la plateforme

Parameters:

r : la regle a ajouter

4.7.3.3 `EcoAgent * PlateformeEcoResolution::getEcoAgent (const EcoAgentID & id) const`

Obtention d'un eco-agent.

Methode qui permet d'obtenir un eco-agent de la plateforme a partir de son identifiant

Parameters:

id : id de l'eco-agent voulu

Returns:

un pointeur sur l'eco-agent recherche s'il existe, NULL sinon

4.7.3.4 `EcoAgentID * PlateformeEcoResolution::getEcoAgentAuDessus (const EcoAgentID & id) const`

Methode qui retourne l'EcoAgentID de l'EcoAgent au dessus.

Methode qui retourne l'EcoAgentID de l'EcoAgent dont la position courante est l'EcoAgentID passe en parametre

Parameters:

id : [EcoAgentID](#) dont on cherche l'EcoAgent qui l'a en position courante

Returns:

[EcoAgentID](#) de l'EcoAgent correspondant, NULL s'il n'y en a pas

4.7.3.5 `map< EcoAgentID, EcoAgent &, compareEcoAgentID > PlateformeEcoResolution::getEcoAgents () const`

Methode qui retourne tous les eco-agents.

Methode qui retourne tous les eco-agents. Cette methode est surtout utilisee pour les regles.

Returns:

un conteneur associatif map<EcoAgentID,EcoAgent&>

4.7.3.6 `list< EcoAgentID * > PlateformeEcoResolution::getEcoAgents (const Etat e) const`

Methode qui retourne les [EcoAgentID](#) des [EcoAgent](#) possedant l'etat specifie.

Methode qui retourne les [EcoAgentID](#) des [EcoAgent](#) possedant un etat specifie dans le parametre de la fonction

Parameters:

e : l'etat dans lequel tous les [EcoAgent](#) listes doivent etre

Returns:

la liste des [EcoAgentID](#) des [EcoAgent](#) dans cet Etat

4.7.3.7 list< Regle * > PlateformeEcoResolution::getRegles ()

Obtention de la liste des regles.

Methode qui permet d'obtenir la liste des regles

Returns:

La liste des regles list<Regle*>

4.7.3.8 virtual void PlateformeEcoResolution::initialiser () [pure virtual]

Initialisation de la resolution.

Methode qui permet d'initialiser le probleme avant d'attaquer la resolution

Implemented in [PlateformeMondeDesCubes](#).

4.7.3.9 int PlateformeEcoResolution::nombreEcoAgentAuDessus (const EcoAgentID & id) const

Methode qui retourne le nombre d'EcoAgent au dessus de l'EcoAgent avec l'id specifie.

Methode qui retourne le nombre d'EcoAgent au dessus de l'EcoAgent avec l'id specifie

Parameters:

id : [EcoAgentID](#) dont on cherche le nombre d'EcoAgent superieur

Returns:

un entier superieur ou egal a 0

4.7.3.10 virtual void PlateformeEcoResolution::resoudre () [pure virtual]

Resolution du probleme par eco-resolution.

Methode qui permet de lancer la resolution du probleme par eco-resolution

Implemented in [PlateformeMondeDesCubes](#).

4.7.3.11 bool PlateformeEcoResolution::sontSatisfaits () const

Methode qui verifie si tous les eco-agents sont satisfaits.

Methode qui verifie si tous les eco-agents sont satisfaits. Elle permet d'arreter la resolution.

Returns:

true si tous les eco-agents sont satisfaits, false sinon

4.7.3.12 `bool PlateformeEcoResolution::verifierCoherence () const`

Verification du respect des regles apres l'initialisation de la plateforme.

Methode qui permet de verifier l'ensemble des regles apres l'initialisation de la plateforme

Returns:

true si toutes les regles sont verifiees, false sinon

4.7.4 Member Data Documentation

4.7.4.1 `map<EcoAgentID,EcoAgent&,compareEcoAgentID> PlateformeEcoResolution::ecoagents` `[protected]`

Referencement des eco-agents a partir de leurs identifiants uniques

4.7.4.2 `list<Regle*> PlateformeEcoResolution::regles` `[protected]`

Liste des regles a verifier avant de lancer la resolution

The documentation for this class was generated from the following files:

- `trunk/include/plateformeEcoResolution.hpp`
- `trunk/src/plateformeEcoResolution.cpp`

4.8 PlateformeMondeDesCubes Class Reference

Classe representant une plateforme d'eco-resolution appliquee au monde des cubes.

```
#include <plateformeMondeDesCubes.hpp>
```

Inherits [PlateformeEcoResolution](#), and [Singleton< PlateformeMondeDesCubes >](#).

Collaboration diagram for PlateformeMondeDesCubes:

Public Member Functions

- void [initialiser](#) ()
Initialisation de la resolution du monde des cubes.
- void [resoudre](#) ()
Resolution du probleme du monde des cubes par eco-resolution.
- [EcoAgent](#) * [obtenirCubePrioritaire](#) () const
Obtention de l'EcoAgent qui a la priorite pour agir dans la plateforme d'eco-resolution du monde des cubes.
- void [setNombreDeCubes](#) (int nb)
Determination du nombre de cubes dans la plateforme d'eco-resolution.
- int [getNombreDeCubes](#) () const
Obtention du nombre de cubes dans la plateforme d'eco-resolution.
- void [setTableIdentifiant](#) (const [EcoAgentID](#) &id)
Determination de l'identifiant de la table dans la plateforme.
- [EcoAgentID](#) * [getTableID](#) () const
Obtention de l'identifiant de la table.
- [EcoAgent](#) * [obtenirGeneur](#) (const [EcoAgent](#) ¤tCube)
Obtention du geneur de l'EcoAgent courant.
- void [setPositionFinale](#) (const [EcoAgentID](#) &c, const [EcoAgentID](#) &pos)
Determination de la position finale d'un EcoAgent de la plateforme.
- void [setPositionCourante](#) (const [EcoAgentID](#) &c, const [EcoAgentID](#) &pos)
Determination de la position courante d'un EcoAgent de la plateforme.
- void [setCubeID](#) ([EcoAgent](#) ¤tCube, const [EcoAgentID](#) &id)
Determination de l'identifiant d'un cube.
- int [distanceFinaleATable](#) (const [EcoAgentID](#) &c) const
Determination du nombre d'EcoAgent separant de la table.

Friends

- class [Singleton](#)< [PlateformeMondeDesCubes](#) >
- ostream & [operator](#)<< (ostream &, const [PlateformeMondeDesCubes](#) &)

Redefinition d'opérateur qui permet d'obtenir des informations sur la plateforme Redefinition d'opérateur qui permet d'obtenir des informations sur la plateforme monde des cubes.

4.8.1 Detailed Description

Classe representant une plateforme d'eco-resolution appliquee au monde des cubes.

La classe gere les fonctionnalites de la plateforme d'eco-resolution du monde des cubes.

4.8.2 Member Function Documentation

4.8.2.1 int PlateformeMondeDesCubes::distanceFinaleATable (const EcoAgentID & c) const

Determination du nombre d'EcoAgent separant de la table.

Cette methode permet de determiner le nombre d'EcoAgent separant un [EcoAgent](#) de la table en position finale.

Parameters:

c : identifiant de l'EcoAgent

Returns:

La distance a la table

4.8.2.2 int PlateformeMondeDesCubes::getNombreDeCubes () const

Obtention du nombre de cubes dans la plateforme d'eco-resolution.

Methode qui permet d'obtenir le nombre de cubes de la plateforme d'eco-resolution

Returns:

le nombre de cubes

4.8.2.3 EcoAgentID * PlateformeMondeDesCubes::getTableID () const

Obtention de l'identifiant de la table.

Methode qui permet d'obtenir l'identifiant de la table

Returns:

(l'identifiant de la table (de type [EcoAgentID](#))) ==> un pointeur sur l'identifiant de la table de type EcoAgentID*

4.8.2.4 void PlateformeMondeDesCubes::initialiser () [virtual]

Initialisation de la resolution du monde des cubes.

Methode qui permet d'initialiser le probleme avant d'attaquer la resolution Les cubes sont disposes sur la table selon une position initiale et une position finale est determinee

Implements [PlateformeEcoResolution](#).

4.8.2.5 EcoAgent * PlateformeMondeDesCubes::obtenirCubePrioritaire () const

Obtention de l'EcoAgent qui a la priorite pour agir dans la plateforme d'eco-resolution du monde des cubes.

Methode qui permet d'obtenir l'EcoAgent qui doit agir en priorite dans la plateforme d'eco-resolution du monde des cubes

Returns:

un pointeur sur l'EcoAgent prioritaire s'il existe, NULL sinon

4.8.2.6 EcoAgent * PlateformeMondeDesCubes::obtenirGeneur (const EcoAgent & *currentCube*)

Obtention du geneur de l'EcoAgent courant.

Methode qui permet d'obtenir le geneur de l'EcoAgent courant Cette methode retourne NULL quand la position courante de l'EcoAgent est sa position finale (ou s'il n'y a aucun geneur)

Parameters:

currentCube : l'EcoAgent courant

Returns:

un pointeur sur l'EcoAgent geneur s'il existe, NULL sinon

4.8.2.7 void PlateformeMondeDesCubes::resoudre () [virtual]

Resolution du probleme du monde des cubes par eco-resolution.

Methode qui permet de lancer la resolution du probleme du monde des cubes par eco-resolution

Implements [PlateformeEcoResolution](#).

4.8.2.8 void PlateformeMondeDesCubes::setCubeID (EcoAgent & *currentCube*, const EcoAgentID & *id*)

Determination de l'identifiant d'un cube.

Methode qui permet de fixer l'identifiant d'un cube de la plateforme lorsqu'on l'initialise

Parameters:

currentCube : cube a identifier

id : identifiant du cube

4.8.2.9 void PlateformeMondeDesCubes::setNombreDeCubes (int *nb*)

Determination du nombre de cubes dans la plateforme d'eco-resolution.

Methode qui permet de fixer le nombre de cubes de la plateforme d'eco-resolution

Parameters:

nb : le nombre de cubes

4.8.2.10 void PlateformeMondeDesCubes::setPositionCourante (const EcoAgentID & *c*, const EcoAgentID & *pos*)

Determination de la position courante d'un [EcoAgent](#) de la plateforme.

Methode qui permet de fixer la position courante d'un cube de la plateforme

Parameters:

c : identifiant de l'EcoAgent a positionner

pos : position courante de l'EcoAgent, ie identifiant du cube destinataire

4.8.2.11 void PlateformeMondeDesCubes::setPositionFinale (const EcoAgentID & *c*, const EcoAgentID & *pos*)

Determination de la position finale d'un [EcoAgent](#) de la plateforme.

Methode qui permet de fixer la position finale d'un cube de la plateforme

Parameters:

c : identifiant de l'EcoAgent a positionner

pos : position finale de l'EcoAgent, ie identifiant du cube destinataire

4.8.2.12 void PlateformeMondeDesCubes::setTableIdentifiant (const EcoAgentID & *id*)

Determination de l'identifiant de la table dans la plateforme.

Methode qui permet de fixer l'identifiant de la table de la plateforme d'eco-resolution

Parameters:

id : identifiant de la table

The documentation for this class was generated from the following files:

- trunk/include/[plateformeMondeDesCubes.hpp](#)
- trunk/src/plateformeMondeDesCubes.cpp

4.9 Regle Class Reference

Cette classe abstraite est une squelette pour une regle.

```
#include <regle.hpp>
```

Inherited by [AucuneSurcharge](#), [Reglefalse](#), [Regletrue](#), and [ReliesATable](#).

Public Member Functions

- virtual void [initialiser](#) ()=0
Suite d'operations realisees pour initialiser la regle.
- virtual bool [verifier](#) ()=0
Verification de la regle.

4.9.1 Detailed Description

Cette classe abstraite est une squelette pour une regle.

Cette classe abstraite est une squelette pour une regle qui sera ajoutee dans une [PlateformeEcoResolution](#)

4.9.2 Member Function Documentation

4.9.2.1 virtual bool Regle::verifier () [pure virtual]

Verification de la regle.

Returns:

true si la regle est verifiee, false sinon

Implemented in [AucuneSurcharge](#).

The documentation for this class was generated from the following file:

- trunk/include/[regle.hpp](#)

4.10 reliesATable Class Reference

Cette classe contient les fonctions testant si les cubes sont lies a la table.

```
#include <reliesATable.hpp>
```

4.10.1 Detailed Description

Cette classe contient les fonctions testant si les cubes sont lies a la table.

The documentation for this class was generated from the following file:

- trunk/include/[reliesATable.hpp](#)

4.11 Singleton< T > Class Template Reference

Template de classe permettant de rendre une classe instanciable une seule fois.

```
#include <singleton.hpp>
```

Static Public Member Functions

- static T * [getInstance](#) ()
Obtention de l'instance.
- static void [kill](#) ()
Destruction de l'instance.

Protected Member Functions

- [Singleton](#) ()
Constructeur.
- [~Singleton](#) ()
Destructeur.

4.11.1 Detailed Description

```
template<typename T> class Singleton< T >
```

Template de classe permettant de rendre une classe instanciable une seule fois.

Une classe qui herite de ce template devient un singleton. Elle sera alors instanciable une unique fois.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 `template<typename T> Singleton< T >::Singleton () [inline, protected]`

Constructeur.

Constructeur de la classe [Singleton](#).

4.11.2.2 `template<typename T> Singleton< T >::~~Singleton () [inline, protected]`

Destructeur.

Destructeur de la classe [Singleton](#).

4.11.3 Member Function Documentation

4.11.3.1 `template<typename T> T * Singleton< T >::getInstance ()` `[inline, static]`

Obtention de l'instance.

Methode qui permet d'obtenir l'unique instance en fonctionnement

Returns:

Un pointeur sur l'instance de type T

4.11.3.2 `template<typename T> void Singleton< T >::kill ()` `[inline, static]`

Destruction de l'instance.

Methode qui permet d'obtenir de détruire l'instance

The documentation for this class was generated from the following file:

- `trunk/include/singleton.hpp`

4.12 Table Class Reference

Classe derivee de la classe [EcoAgent](#) designant le Support sur lequel vont etre poses les cubes.

```
#include <table.hpp>
```

Inherits [EcoAgent](#).

Collaboration diagram for Table:

Public Member Functions

- [Table](#) ()
Constructeur.
- [Table](#) ([EcoAgentID](#) &id)
Constructeur.
- [~Table](#) ()
Destructeur.
- void [rechercherFuite](#) ()
Suite d'operations realisees par la table lorsqu'elle cherche a fuir.
- void [rechercherSatisfaction](#) ()
Suite d'operations realisees par la table lorsqu'elle cherche a se satisfaire.
- void [agresser](#) ([EcoAgent](#) &a)
Suite d'operations realisees lorsque la table agresse un autre eco-agent.
- void [estAgresse](#) ()
Suite d'operations realisees par la table lorsqu'elle est agressee.
- void [faireFuite](#) ()
Suite d'operations realisees par la fuite lorsqu'elle fuit.
- void [faireSatisfaction](#) ()
Suite d'operations realisees par la table lorsqu'elle se satisfait.
- void [agir](#) ()
Suite d'operations realisees par la table lorsqu'il agit.
- void [initialiser](#) ()
Initialisation de la table avant l'EcoResolution.

4.12.1 Detailed Description

Classe derivee de la classe [EcoAgent](#) designant le Support sur lequel vont etre poses les cubes.

4.12.2 Constructor & Destructor Documentation

4.12.2.1 Table::Table ()

Constructeur.

Constructeur de la classe [Table](#) par default. Un [EcoAgentID](#) sera automatiquement genere pour la table.

4.12.2.2 Table::Table (EcoAgentID & *id*)

Constructeur.

Constructeur de la classe [Table](#)

Parameters:

id : identifiant unique que l'agent se verra attribuer

4.12.2.3 Table::~~Table ()

Destructeur.

Destructeur de la classe [Table](#)

4.12.3 Member Function Documentation

4.12.3.1 void Table::agresser (EcoAgent & *a*) [virtual]

Suite d'operations realisees lorsque la table agresse un autre eco-agent.

Parameters:

a : [EcoAgent](#) a agresser

Implements [EcoAgent](#).

The documentation for this class was generated from the following files:

- trunk/include/[table.hpp](#)
- trunk/src/[table.cpp](#)

Chapter 5

File Documentation

5.1 trunk/include/aucuneSurcharge.hpp File Reference

verification de la regle : est-ce que chaque cube porte au plus un cube?

```
#include "plateformeMondeDesCubes.hpp"
#include "regle.hpp"
#include "cube.hpp"
#include <map>
#include <iostream>
```

Include dependency graph for aucuneSurcharge.hpp:

Classes

- class [AucuneSurcharge](#)

Cette classe contient les fonctions testant si les cubes ne portent pas plus de un cube.

5.1.1 Detailed Description

verification de la regle : est-ce que chaque cube porte au plus un cube?

Author:

GDD
Pauline Requena
Arnaud Faure

Version:

0.1

Date:

04 avril 2009

5.2 trunk/include/compareEcoAgentID.hpp File Reference

Comparaison d'EcoAgentID.

```
#include "ecoAgentID.hpp"
```

Include dependency graph for compareEcoAgentID.hpp:

This graph shows which files directly or indirectly include this file:

Classes

- class [compareEcoAgentID](#)

Structure contenant la redefinition de l'operateur de comparaison d'EcoAgentID Structure contenant la redefinition de l'operateur de comparaison d'EcoAgentID pour le transformer en cle unique d'une map.

5.2.1 Detailed Description

Comparaison d'EcoAgentID.

Author:

GDD
Pauline Requena
Arnaud Faure

Version:

0.1

Date:

18 avril 2009

5.3 trunk/include/cube.hpp File Reference

Implementation du module cube qui est un derive d'un [EcoAgent](#).

```
#include <iostream>
#include "ecoAgent.hpp"
#include "ecoAgentID.hpp"
#include "plateformeEcoResolution.hpp"
```

Include dependency graph for cube.hpp:

This graph shows which files directly or indirectly include this file:

Classes

- class [Cube](#)
Classe derivee de la classe [EcoAgent](#) designant un [Cube](#).

5.3.1 Detailed Description

Implementation du module cube qui est un derive d'un [EcoAgent](#).

Author:

GDD
Arnaud Faure
Pauline Requena

Version:

0.1

Date:

04 avril 2009

Implementation de la classe [Cube](#) qui est une classe derivee de la classe [EcoAgent](#).

5.4 trunk/include/ecoAgent.hpp File Reference

Mise en place de la classe abstraite [EcoAgent](#).

```
#include <iostream>
#include "ecoAgentID.hpp"
#include "etat.hpp"
```

Include dependency graph for ecoAgent.hpp:

This graph shows which files directly or indirectly include this file:

Classes

- class [EcoAgent](#)
Classe abstraite qui permet de creer un eco-agent.

5.4.1 Detailed Description

Mise en place de la classe abstraite [EcoAgent](#).

Author:

GDD
Arnaud Faure
Pauline Requena

Version:

0.1

Date:

31 mars 2009

Implementation de la classe abstraite [EcoAgent](#) qui regroupe les fonctionnalites communes a tout eco-agent

5.5 trunk/include/ecoAgentID.hpp File Reference

Implementation de la classe [EcoAgentID](#).

```
#include <iostream>
```

Include dependency graph for ecoAgentID.hpp:

This graph shows which files directly or indirectly include this file:

Classes

- class [EcoAgentID](#)
Identifiant unique d'un eco-agent.

5.5.1 Detailed Description

Implementation de la classe [EcoAgentID](#).

Author:

GDD
Arnaud Faure
Pauline Requena

Version:

0.1

Date:

04 avril 2009

Implementation de la classe [EcoAgentID](#) qui est un identifiant unique d'un eco-agent

5.6 trunk/include/etat.hpp File Reference

Enumeration des etats possibles des eco-agents.

```
#include <iostream>
```

Include dependency graph for etat.hpp:

This graph shows which files directly or indirectly include this file:

Enumerations

- enum [Etat](#)

Enumeration repertoriant les etats possibles d'un eco-agent.

5.6.1 Detailed Description

Enumeration des etats possibles des eco-agents.

Author:

GDD
Arnaud Faure
Pauline Requena

Version:

0.1

Date:

04 avril 2009

Implementation de l'enumeration des etats que peuvent prendre les eco-agents

5.7 trunk/include/ExceptionEcoAgentDejaEnregistre.hpp File Reference

Exception lancee lorsqu'on enregistre dans la plateforme un agent qui a deja ete enregistre.

This graph shows which files directly or indirectly include this file:

Classes

- class [ExceptionEcoAgentDejaEnregistre](#)

Exception lancee dans le cadre de l'ajout d'un [EcoAgent](#) deja enregistre dans une [PlateformeEcoResolution](#).

5.7.1 Detailed Description

Exception lancee lorsqu'on enregistre dans la plateforme un agent qui a deja ete enregistre.

Author:

GDD
Pauline Requena
Arnaud Faure

Version:

0.1

Date:

24 avril 2009

5.8 trunk/include/plateformeEcoResolution.hpp File Reference

Plateforme abstraite d'eco-resolution.

```
#include <iostream>
#include <map>
#include <list>
#include "ecoAgentID.hpp"
#include "ecoAgent.hpp"
#include "regle.hpp"
#include "compareEcoAgentID.hpp"
#include "ExceptionEcoAgentDejaEnregistre.hpp"
```

Include dependency graph for plateformeEcoResolution.hpp:

This graph shows which files directly or indirectly include this file:

Classes

- class [PlateformeEcoResolution](#)
classe representant une plateforme d'eco-resolution abstraite

5.8.1 Detailed Description

Plateforme abstraite d'eco-resolution.

Author:

GDD
Pauline Requena
Arnaud Faure

Version:

0.1

Date:

29 mars 2009

5.9 trunk/include/plateformeMondeDesCubes.hpp File Reference

Plateforme d'eco-resolution appliquee au monde des cubes.

```
#include <iostream>
#include "plateformeEcoResolution.hpp"
#include "singleton.hpp"
#include "regle.hpp"
#include "ecoAgent.hpp"
#include "ecoAgentID.hpp"
#include "table.hpp"
#include "cube.hpp"
```

Include dependency graph for plateformeMondeDesCubes.hpp:

This graph shows which files directly or indirectly include this file:

Classes

- class [PlateformeMondeDesCubes](#)

Classe representant une plateforme d'eco-resolution appliquee au monde des cubes.

5.9.1 Detailed Description

Plateforme d'eco-resolution appliquee au monde des cubes.

Author:

GDD
Pauline Requena
Arnaud Faure

Version:

0.1

Date:

02 avril 2009

5.10 trunk/include/regle.hpp File Reference

Squelette d'une regle pour une plateforme d'eco-resolution.

```
#include <iostream>
```

Include dependency graph for regle.hpp:

This graph shows which files directly or indirectly include this file:

Classes

- class [Regle](#)

Cette classe abstraite est une squelette pour une regle.

5.10.1 Detailed Description

Squelette d'une regle pour une plateforme d'eco-resolution.

Author:

GDD
Pauline Requena
Arnaud Faure

Version:

0.1

Date:

04 avril 2009

5.11 trunk/include/reliesATable.hpp File Reference

verification de la regle : est-ce que tous les cubes sont et seront relies directement ou indirectement a la table?

```
#include "plateformeMondeDesCubes.hpp"
#include "regle.hpp"
#include <iostream>
#include <map>
#include "cube.hpp"
```

Include dependency graph for reliesATable.hpp:

5.11.1 Detailed Description

verification de la regle : est-ce que tous les cubes sont et seront relies directement ou indirectement a la table?

Author:

GDD
Pauline Requena
Arnaud Faure

Version:

0.1

Date:

04 avril 2009

5.12 trunk/include/singleton.hpp File Reference

Implementation du design pattern singleton.

```
#include <iostream>
```

Include dependency graph for singleton.hpp:

This graph shows which files directly or indirectly include this file:

Classes

- class [Singleton< T >](#)

Template de classe permettant de rendre une classe instanciable une seule fois.

5.12.1 Detailed Description

Implementation du design pattern singleton.

Author:

GDD
Arnaud Faure
Pauline Requena

Version:

0.1

Date:

29 mars 2009

Implementation du design pattern singleton pour rendre une classe instanciable une unique fois.

5.13 trunk/include/table.hpp File Reference

Implementation du module table qui est un derive d'un [EcoAgent](#).

```
#include <iostream>
#include "ecoAgentID.hpp"
#include "ecoAgent.hpp"
```

Include dependency graph for table.hpp:

This graph shows which files directly or indirectly include this file:

Classes

- class [Table](#)

Classe derivee de la classe [EcoAgent](#) designant le Support sur lequel vont etre poses les cubes.

5.13.1 Detailed Description

Implementation du module table qui est un derive d'un [EcoAgent](#).

Author:

GDD
Arnaud Faure
Pauline Requena

Version:

0.1

Date:

1er avril 2009

Implementation de la classe [Table](#) qui est une classe derivee de la classe [EcoAgent](#).

Index

- ~Cube
 - Cube, [11](#)
- ~EcoAgentID
 - EcoAgentID, [18](#)
- ~PlateformeEcoResolution
 - PlateformeEcoResolution, [21](#)
- ~Singleton
 - Singleton, [31](#)
- ~Table
 - Table, [34](#)
- addEcoAgent
 - PlateformeEcoResolution, [21](#)
- addRegle
 - PlateformeEcoResolution, [21](#)
- agresser
 - Cube, [11](#)
 - EcoAgent, [13](#)
 - Table, [34](#)
- AucuneSurcharge, [7](#)
 - pasSurcharges, [7](#)
 - serontPasSurcharges, [8](#)
 - verifier, [8](#)
- compareEcoAgentID, [9](#)
- Cube, [10](#)
 - ~Cube, [11](#)
 - agresser, [11](#)
 - Cube, [11](#)
 - initialiser, [11](#)
- distanceFinaleATable
 - PlateformeMondeDesCubes, [26](#)
- EcoAgent, [12](#)
 - agresser, [13](#)
 - EcoAgent, [13](#)
 - getEtat, [13](#)
 - getId, [14](#)
 - getPositionCourante, [14](#)
 - getPositionFinale, [14](#)
 - getVisite, [14](#)
 - initialiser, [14](#)
 - setEtat, [15](#)
 - setId, [15](#)
 - setPositionCourante, [15](#)
 - setPositionFinale, [15](#)
 - setVisite, [15](#)
- EcoAgentID, [17](#)
 - ~EcoAgentID, [18](#)
 - EcoAgentID, [18](#)
 - getId, [18](#)
 - operator<, [18](#)
 - operator==, [18](#)
- ecoagents
 - PlateformeEcoResolution, [24](#)
- ExceptionEcoAgentDejaEnregistre, [19](#)
- getEcoAgent
 - PlateformeEcoResolution, [21](#)
- getEcoAgentAuDessus
 - PlateformeEcoResolution, [22](#)
- getEcoAgents
 - PlateformeEcoResolution, [22](#)
- getEtat
 - EcoAgent, [13](#)
- getId
 - EcoAgent, [14](#)
 - EcoAgentID, [18](#)
- getInstance
 - Singleton, [32](#)
- getNombreDeCubes
 - PlateformeMondeDesCubes, [26](#)
- getPositionCourante
 - EcoAgent, [14](#)
- getPositionFinale
 - EcoAgent, [14](#)
- getRegles
 - PlateformeEcoResolution, [22](#)
- getTableID
 - PlateformeMondeDesCubes, [26](#)
- getVisite
 - EcoAgent, [14](#)
- initialiser
 - Cube, [11](#)
 - EcoAgent, [14](#)
 - PlateformeEcoResolution, [23](#)
 - PlateformeMondeDesCubes, [26](#)
- kill

- Singleton, 32
- nombreEcoAgentAuDessus
 - PlateformeEcoResolution, 23
- obtenirCubePrioritaire
 - PlateformeMondeDesCubes, 27
- obtenirGeneur
 - PlateformeMondeDesCubes, 27
- operator<
 - EcoAgentID, 18
- operator==
 - EcoAgentID, 18
- pasSurcharges
 - AucuneSurcharge, 7
- PlateformeEcoResolution, 20
 - ~PlateformeEcoResolution, 21
 - addEcoAgent, 21
 - addRegle, 21
 - ecoagents, 24
 - getEcoAgent, 21
 - getEcoAgentAuDessus, 22
 - getEcoAgents, 22
 - getRegles, 22
 - initialiser, 23
 - nombreEcoAgentAuDessus, 23
 - PlateformeEcoResolution, 21
 - regles, 24
 - resoudre, 23
 - sontSatisfaits, 23
 - verifierCoherence, 23
- PlateformeMondeDesCubes, 25
 - distanceFinaleATable, 26
 - getNombreDeCubes, 26
 - getTableID, 26
 - initialiser, 26
 - obtenirCubePrioritaire, 27
 - obtenirGeneur, 27
 - resoudre, 27
 - setCubeID, 27
 - setNombreDeCubes, 27
 - setPositionCourante, 28
 - setPositionFinale, 28
 - setTableIdentifiant, 28
- Regle, 29
 - verifier, 29
- regles
 - PlateformeEcoResolution, 24
- reliesATable, 30
- resoudre
 - PlateformeEcoResolution, 23
 - PlateformeMondeDesCubes, 27
- serontPasSurcharges
 - AucuneSurcharge, 8
- setCubeID
 - PlateformeMondeDesCubes, 27
- setEtat
 - EcoAgent, 15
- setId
 - EcoAgent, 15
- setNombreDeCubes
 - PlateformeMondeDesCubes, 27
- setPositionCourante
 - EcoAgent, 15
 - PlateformeMondeDesCubes, 28
- setPositionFinale
 - EcoAgent, 15
 - PlateformeMondeDesCubes, 28
- setTableIdentifiant
 - PlateformeMondeDesCubes, 28
- setVisite
 - EcoAgent, 15
- Singleton, 31
 - ~Singleton, 31
 - getInstance, 32
 - kill, 32
 - Singleton, 31
- sontSatisfaits
 - PlateformeEcoResolution, 23
- Table, 33
 - ~Table, 34
 - agresser, 34
 - Table, 34
- trunk/include/aucuneSurcharge.hpp, 35
- trunk/include/compareEcoAgentID.hpp, 36
- trunk/include/cube.hpp, 37
- trunk/include/ecoAgent.hpp, 38
- trunk/include/ecoAgentID.hpp, 39
- trunk/include/etat.hpp, 40
- trunk/include/ExceptionEcoAgentDejaEnregistre.hpp, 41
- trunk/include/plateformeEcoResolution.hpp, 42
- trunk/include/plateformeMondeDesCubes.hpp, 43
- trunk/include/regle.hpp, 44
- trunk/include/reliesATable.hpp, 45
- trunk/include/singleton.hpp, 46
- trunk/include/table.hpp, 47
- verifier
 - AucuneSurcharge, 8
 - Regle, 29
- verifierCoherence
 - PlateformeEcoResolution, 23