

Monde des cubes

0.1

Generated by Doxygen 1.5.7.1

Tue Jun 2 14:02:39 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	12
4.4	EcoAgent Class Reference	13
4.4.1	Detailed Description	14

4.4.2	Constructor & Destructor Documentation	14
4.4.2.1	EcoAgent	14
4.4.3	Member Function Documentation	15
4.4.3.1	agresser	15
4.4.3.2	getEtat	15
4.4.3.3	getId	15
4.4.3.4	getPositionCourante	15
4.4.3.5	getPositionFinale	15
4.4.3.6	getVisite	16
4.4.3.7	initialiser	16
4.4.3.8	setEtat	16
4.4.3.9	setId	16
4.4.3.10	setPositionCourante	16
4.4.3.11	setPositionFinale	17
4.4.3.12	setVisite	17
4.5	EcoAgentID Class Reference	18
4.5.1	Detailed Description	18
4.5.2	Constructor & Destructor Documentation	19
4.5.2.1	EcoAgentID	19
4.5.2.2	EcoAgentID	19
4.5.2.3	~EcoAgentID	19
4.5.3	Member Function Documentation	19
4.5.3.1	getId	19
4.5.3.2	operator!=	19
4.5.3.3	operator<	19
4.5.3.4	operator==	19
4.6	ExceptionCubeNonRelie Class Reference	20
4.6.1	Detailed Description	20
4.7	ExceptionCubeSeraNonRelie Class Reference	21
4.7.1	Detailed Description	21
4.8	ExceptionEcoAgentDejaEnregistre Class Reference	22
4.8.1	Detailed Description	22
4.9	ExceptionIIExisteraUneBoucle Class Reference	23
4.9.1	Detailed Description	23
4.10	ExceptionIIExisteUneBoucle Class Reference	24
4.10.1	Detailed Description	24

4.11	ExceptionUnCubeEstSurcharge Class Reference	25
4.11.1	Detailed Description	25
4.12	ExceptionUnCubeSeraSurcharge Class Reference	26
4.12.1	Detailed Description	26
4.13	PlateformeEcoResolution Class Reference	27
4.13.1	Detailed Description	28
4.13.2	Constructor & Destructor Documentation	28
4.13.2.1	PlateformeEcoResolution	28
4.13.2.2	~PlateformeEcoResolution	28
4.13.3	Member Function Documentation	28
4.13.3.1	addEcoAgent	28
4.13.3.2	addRegle	28
4.13.3.3	getEcoAgent	29
4.13.3.4	getEcoAgentAuDessus	29
4.13.3.5	getEcoAgents	29
4.13.3.6	getEcoAgents	29
4.13.3.7	getRegles	30
4.13.3.8	initialiser	30
4.13.3.9	nombreEcoAgentAuDessus	30
4.13.3.10	resoudre	30
4.13.3.11	sontSatisfaits	30
4.13.3.12	verifierCoherence	31
4.13.4	Member Data Documentation	31
4.13.4.1	ecoagents	31
4.13.4.2	regles	31
4.14	PlateformeMondeDesCubes Class Reference	32
4.14.1	Detailed Description	33
4.14.2	Member Function Documentation	33
4.14.2.1	addEcoAgent	33
4.14.2.2	distanceATable	33
4.14.2.3	distanceFinaleATable	34
4.14.2.4	getNombreDeCubes	34
4.14.2.5	getTableID	34
4.14.2.6	initialiser	34
4.14.2.7	obtenirCubePrioritaire	34
4.14.2.8	obtenirGeneur	35

4.14.2.9	resoudre	35
4.14.2.10	setCubeID	35
4.14.2.11	setPositionCourante	35
4.14.2.12	setPositionFinale	35
4.14.2.13	setTableIdentifiant	36
4.14.2.14	toString	36
4.14.3	Friends And Related Function Documentation	36
4.14.3.1	operator<<	36
4.15	Regle Class Reference	37
4.15.1	Detailed Description	37
4.15.2	Member Function Documentation	37
4.15.2.1	verifier	37
4.16	reliesATable Class Reference	38
4.16.1	Detailed Description	38
4.17	Singleton< T > Class Template Reference	39
4.17.1	Detailed Description	39
4.17.2	Constructor & Destructor Documentation	39
4.17.2.1	Singleton	39
4.17.2.2	~Singleton	39
4.17.3	Member Function Documentation	40
4.17.3.1	getInstance	40
4.17.3.2	kill	40
4.18	Table Class Reference	41
4.18.1	Detailed Description	42
4.18.2	Constructor & Destructor Documentation	42
4.18.2.1	Table	42
4.18.2.2	Table	42
4.18.2.3	~Table	42
4.18.3	Member Function Documentation	42
4.18.3.1	agresser	42
5	File Documentation	43
5.1	trunk/include/aucuneSurcharge.hpp File Reference	43
5.1.1	Detailed Description	43
5.2	trunk/include/compareEcoAgentID.hpp File Reference	44
5.2.1	Detailed Description	44
5.3	trunk/include/cube.hpp File Reference	45

5.3.1	Detailed Description	45
5.4	trunk/include/ecoAgent.hpp File Reference	46
5.4.1	Detailed Description	46
5.5	trunk/include/ecoAgentID.hpp File Reference	47
5.5.1	Detailed Description	47
5.6	trunk/include/etat.hpp File Reference	48
5.6.1	Detailed Description	48
5.7	trunk/include/Exceptions.hpp File Reference	49
5.7.1	Detailed Description	49
5.8	trunk/include/plateformeEcoResolution.hpp File Reference	50
5.8.1	Detailed Description	50
5.9	trunk/include/plateformeMondeDesCubes.hpp File Reference	51
5.9.1	Detailed Description	51
5.10	trunk/include/regle.hpp File Reference	52
5.10.1	Detailed Description	52
5.11	trunk/include/reliesATable.hpp File Reference	53
5.11.1	Detailed Description	53
5.12	trunk/include/singleton.hpp File Reference	54
5.12.1	Detailed Description	54
5.13	trunk/include/table.hpp File Reference	55
5.13.1	Detailed Description	55

Chapter 1

Class Index

1.1 Class Hierarchy

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

compareEcoAgentID	9
EcoAgent	13
Cube	10
Table	41
EcoAgentID	18
ExceptionCubeNonRelie	20
ExceptionCubeSeraNonRelie	21
ExceptionEcoAgentDejaEnregistre	22
ExceptionIIExisteraUneBoucle	23
ExceptionIIExisteUneBoucle	24
ExceptionUnCubeEstSurcharge	25
ExceptionUnCubeSeraSurcharge	26
PlateformeEcoResolution	27
PlateformeMondeDesCubes	32
Regle	37
AucuneSurcharge	7
reliesATable	38
Singleton< T >	39
Singleton< PlateformeMondeDesCubes >	39
PlateformeMondeDesCubes	32

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 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)	13
EcoAgentID (Identifiant unique d'un EcoAgent)	18
ExceptionCubeNonRelie (Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes sont relies a la table")	20
ExceptionCubeSeraNonRelie (Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes seront relies a la table")	21
ExceptionEcoAgentDejaEnregistre (Exception lancee dans le cadre de l'ajout d'un EcoAgent deja enregistre dans une PlateformeEcoResolution)	22
ExceptionIlexisteraUneBoucle (Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes seront relies a la table")	23
ExceptionIlexisteUneBoucle (Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes sont relies a la table")	24
ExceptionUnCubeEstSurcharge (Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes sont surcharges")	25
ExceptionUnCubeSeraSurcharge (Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes seront surcharges")	26
PlateformeEcoResolution (Classe representant une plateforme d'eco-resolution abstraite)	27
PlateformeMondeDesCubes (Classe representant une plateforme d'eco-resolution appliquee au monde des cubes)	32
Regle (Cette classe abstraite est une squelette pour une regle)	37
reliesATable (Cette classe contient les fonctions testant si les cubes sont lies a la table)	38
Singleton< T > (Template de classe permettant de rendre une classe instanciable une seule fois)	39
Table (Classe derivee de la classe EcoAgent designant le support sur lequel vont etre poses les cubes)	41

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?)	43
trunk/include/ compareEcoAgentID.hpp (Comparaison d'EcoAgentID)	44
trunk/include/ cube.hpp (Implementation du module cube qui est un derive d'un EcoAgent) . . .	45
trunk/include/ ecoAgent.hpp (Mise en place de la classe abstraite EcoAgent)	46
trunk/include/ ecoAgentID.hpp (Implementation de la classe EcoAgentID)	47
trunk/include/ etat.hpp (Enumeration des etats possibles des eco-agents)	48
trunk/include/ Exceptions.hpp (Liste des Exceptions possibles)	49
trunk/include/ plateformeEcoResolution.hpp (Plateforme abstraite d'eco-resolution)	50
trunk/include/ plateformeMondeDesCubes.hpp (Plateforme d'eco-resolution appliquee au monde des cubes)	51
trunk/include/ regle.hpp (Squelette d'une regle pour une plateforme d'eco-resolution)	52
trunk/include/ reliesATable.hpp (Cette regle verifie que tous les cubes sont relies a la table a l'initialisation de la resolution ainsi que dans l'etat final de la resolution)	53
trunk/include/ singleton.hpp (Implementation du design pattern singleton)	54
trunk/include/ table.hpp (Implementation du module table qui est un derive d'un EcoAgent) . . .	55

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, une exception sinon

Exceptions:

[ExceptionUnCubeEstSurcharge](#) : lancee si au moins un cube est surcharge a l'etat initial de la resolution

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, une exception sinon

Exceptions:

[ExceptionUnCubeSeraSurcharge](#) : lancee si au moins un cube est surcharge a l'etat final de la resolution

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, une exception sinon

Exceptions:

[ExceptionUnCubeEstSurcharge](#) : lancee si au moins un cube est surcharge a l'etat initial de la resolution

[ExceptionUnCubeSeraSurcharge](#) : lancee si au moins un cube est surcharge a l'etat final de la resolution

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 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 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 Le comportement de recherche de fuite est le suivant : le cube determine s'il a un geneur (un [EcoAgent](#) au dessus de lui ou au dessus de sa position finale). S'il possede un geneur, il l'agresse, sinon il fuit.
- void [rechercherSatisfaction](#) ()
Suite d'operations realisees par le cube lorsqu'il cherche a se satisfaire Le comportement de recherche de satisfaction est le suivant : le cube determine s'il a un geneur (un [EcoAgent](#) au dessus de lui ou au dessus de sa position finale). S'il possede un geneur, il l'agresse, sinon il se satisfait.
- 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 Lorsque le cube est agresse, il passe son Etat en RECHERCHEFUITE.
- void [faireFuite](#) ()
Suite d'operations realisees par le cube lorsqu'il fuit Si le cube est agresse par le cube qui est sa position finale ou alors que sa position finale n'est pas satisfaite alors le cube fuit sur la table. Sinon le cube fuit en allant directement sur le cube qui est sa position finale.
- void [faireSatisfaction](#) ()
Suite d'operations realisees par le cube lorsqu'il se satisfait Le cube se deplace sur sa position finale et passe sont Etat a SATISFAIT.
- void [initialiser](#) ()
Initialisation de l'etat du cube.
- void [agir](#) ()

Suite d'operations realisees par le cube lorsqu'il agit Le cube determine comment il doit agir en fonction de son Etat. S'il est en RECHERCHESATISFACTION il execute [rechercherSatisfaction\(\)](#), s'il est en RECHERCHEFUITE il execute [rechercherFuite\(\)](#).

Friends

- ostream & [operator<<](#) (ostream &f, const [Cube](#) &c)

Suite d'operations permettant d'afficher des informations sur le cube Affichage d'information sur le cube comme son [EcoAgentID](#), sont Etat et ses positions initiale et finale.

4.3.1 Detailed Description

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

Un [Cube](#) s'execute generalement dans une [PlateformeMondeDesCubes](#) afin de resoudre le probleme du monde des cubes. Un cube peut etre pose sur une [Table](#), un autre [Cube](#) et plus generalement un autre [EcoAgent](#) en rapport avec le probleme du monde des cubes. Cette classe determine les comportements d'un [Cube](#) quand il doit agir dans le cadre d'une eco-resolution (fuite, satisfaction, agression).

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](#).
- virtual [~EcoAgent](#) ()
Destructeur de la classe [EcoAgent](#).
- [EcoAgent](#) (const [EcoAgent](#) &)
Constructeur par recopie.
- [EcoAgent](#) & [operator=](#) (const [EcoAgent](#) &)
Surcharge de l'affectation.
- [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 [EcoAgent](#).

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

Public Member Functions

- [EcoAgentID](#) ()
Constructeur.
- [EcoAgentID](#) (const int id)
Constructeur avec un identifiant specifie Ce constructeur cree un [EcoAgentID](#) avec l'identifiant donne en parametre.
- virtual [~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 [EcoAgent](#).

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 (const int *id*)

Constructeur avec un identifiant specifie Ce constructeur cree un [EcoAgentID](#) avec l'identifiant donne en parametre.

Parameters:

id : valeur de l'identifiant de l'EcoAgentID cree

4.5.2.3 EcoAgentID::~~EcoAgentID () [virtual]

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 ExceptionCubeNonRelie Class Reference

Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes sont relies a la table".

```
#include <Exceptions.hpp>
```

4.6.1 Detailed Description

Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes sont relies a la table".

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

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

4.7 ExceptionCubeSeraNonRelie Class Reference

Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes seront relies a la table".

```
#include <Exceptions.hpp>
```

4.7.1 Detailed Description

Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes seront relies a la table".

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

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

4.8 ExceptionEcoAgentDejaEnregistre Class Reference

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

```
#include <Exceptions.hpp>
```

4.8.1 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/Exceptions.hpp](#)

4.9 ExceptionIIExisteraUneBoucle Class Reference

Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes seront relies a la table".

```
#include <Exceptions.hpp>
```

4.9.1 Detailed Description

Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes seront relies a la table".

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

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

4.10 ExceptionIIExisteUneBoucle Class Reference

Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes sont relies a la table".

```
#include <Exceptions.hpp>
```

4.10.1 Detailed Description

Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes sont relies a la table".

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

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

4.11 ExceptionUnCubeEstSurcharge Class Reference

Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes sont surcharges".

```
#include <Exceptions.hpp>
```

4.11.1 Detailed Description

Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes sont surcharges".

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

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

4.12 ExceptionUnCubeSeraSurcharge Class Reference

Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes seront surcharges".

```
#include <Exceptions.hpp>
```

4.12.1 Detailed Description

Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes seront surcharges".

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

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

4.13 PlateformeEcoResolution Class Reference

classe representant une plateforme d'eco-resolution abstraite

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

Inherited by [PlateformeMondeDesCubes](#).

Public Member Functions

- [PlateformeEcoResolution](#) ()
Constructeur.
- virtual [~PlateformeEcoResolution](#) ()
Destructeur.
- [EcoAgent](#) * [getEcoAgent](#) (const [EcoAgentID](#) &id) const
Obtention d'un [EcoAgent](#).
- virtual void [addEcoAgent](#) ([EcoAgent](#) &ea)
Ajout d'un [EcoAgent](#).
- 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 EcoAgents sont satisfaits i.e. si la resolution est terminee.
- [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](#)

4.13.1 Detailed Description

classe representant une plateforme d'eco-resolution abstraite

La classe gere les fonctionnalites basiques d'une plateforme d'eco-resolution. Elle permet l'ajout d'EcoAgent et les references grace a leur identifiant [EcoAgentID](#). Elle integre un ensemble de fonctionnalites "basiques" qui permettent par exemple d'obtenir un [EcoAgent](#) reference a partir de son ID, d'ajouter des regles et de les verifier. Bien evidemment, la plateforme permet de lancer la resolution en elle-meme.

4.13.2 Constructor & Destructor Documentation

4.13.2.1 PlateformeEcoResolution::PlateformeEcoResolution ()

Constructeur.

Constructeur de la classe abstraite [PlateformeEcoResolution](#)

4.13.2.2 PlateformeEcoResolution::~~PlateformeEcoResolution () [virtual]

Destructeur.

Destructeur de la classe abstraite [PlateformeEcoResolution](#)

4.13.3 Member Function Documentation

4.13.3.1 void PlateformeEcoResolution::addEcoAgent (EcoAgent & *ea*) [virtual]

Ajout d'un [EcoAgent](#).

Methode qui permet d'ajouter un [EcoAgent](#) dans la plateforme. Il est alors reference par son [EcoAgentID](#).

Exceptions:

ExceptionEcoAgentDejaEnregistre : lancee lorsqu'on enregistre un [EcoAgent](#) deja enregistre, autrement dit lorsqu'un [EcoAgent](#) avec le meme [EcoAgentID](#) a deja ete ajoute.

Parameters:

ea : l'EcoAgent a ajouter

Reimplemented in [PlateformeMondeDesCubes](#).

4.13.3.2 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

4.13.3.3 EcoAgent * PlateformeEcoResolution::getEcoAgent (const EcoAgentID & id) const

Obtention d'un [EcoAgent](#).

Methode qui permet d'obtenir un [EcoAgent](#) de la plateforme a partir de son identifiant

Parameters:

id : id de l'EcoAgent voulu

Returns:

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

4.13.3.4 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

4.13.3.5 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&>

4.13.3.6 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'état dans lequel tous les [EcoAgent](#) listes doivent être

Returns:

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

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

Obtention de la liste des règles.

Méthode qui permet d'obtenir la liste des règles

Returns:

La liste des règles list<Regle*>

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

Initialisation de la résolution.

Méthode qui permet d'initialiser le problème avant d'attaquer la résolution

Implemented in [PlateformeMondeDesCubes](#).

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

Méthode qui retourne le nombre d'EcoAgent au dessus de l'EcoAgent avec l'id spécifiée.

Méthode qui retourne le nombre d'EcoAgent au dessus de l'EcoAgent avec l'id spécifiée

Parameters:

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

Returns:

un entier supérieur ou égal à 0

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

Résolution du problème par eco-résolution.

Méthode qui permet de réaliser une étape de la résolution du problème par eco-résolution

Implemented in [PlateformeMondeDesCubes](#).

4.13.3.11 bool PlateformeEcoResolution::sontSatisfaits () const

Méthode qui vérifie si tous les EcoAgents sont satisfaits i.e. si la résolution est terminée.

Méthode qui vérifie si tous les eco-agents sont satisfaits. Elle permet de déterminer quand la résolution est terminée.

Returns:

true si tous les EcoAgents sont satisfaits, false sinon

4.13.3.12 bool PlateformeEcoResolution::verifierCoherence () const

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

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

Returns:

true si toutes les regles sont verifiees, false sinon

4.13.4 Member Data Documentation**4.13.4.1 map<EcoAgentID,EcoAgent&,compareEcoAgentID> PlateformeEcoResolution::ecoagents [protected]**

Referencement des eco-agents a partir de leurs identifiants uniques

4.13.4.2 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

4.14 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.
- int [getNombreDeCubes](#) () const
Obtention du nombre de cubes present 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 [distanceATable](#) (const [EcoAgentID](#) &c) const
Determination du nombre d'EcoAgent separant de la table.
- int [distanceFinaleATable](#) (const [EcoAgentID](#) &c) const
Determination du nombre d'EcoAgent separant de la table en position finale.
- void [addEcoAgent](#) ([EcoAgent](#) &ea)
Ajout d'un EcoAgent en verifiant que l'EcoAgentID est different de celui de la Table.

- string [toString](#) () const

Methode qui permet d'obtenir un affichage minimal de l'agencement des cubes.

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.

4.14.1 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.

4.14.2 Member Function Documentation

4.14.2.1 void [PlateformeMondeDesCubes::addEcoAgent](#) ([EcoAgent](#) & *ea*) [virtual]

Ajout d'un [EcoAgent](#) en verifiant que l'EcoAgentID est different de celui de la [Table](#).

Methode qui permet d'ajouter un eco-agent dans la plateforme tout en verifiant que l'identifiant de celui-ci est different de l'identifiant de la table

Exceptions:

[ExceptionEcoAgentDejaEnregistre](#) : lancee lorsqu'on enregistre un [EcoAgent](#) deja enregistre, autrement dit lorsqu'un [EcoAgent](#) avec le meme [EcoAgentID](#) a deja ete ajoute.

Parameters:

ea : l'EcoAgent a ajouter

Reimplemented from [PlateformeEcoResolution](#).

4.14.2.2 int [PlateformeMondeDesCubes::distanceATable](#) (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 courante.

Parameters:

c : identifiant de l'EcoAgent

Returns:

La distance a la table

4.14.2.3 `int PlateformeMondeDesCubes::distanceFinaleATable (const EcoAgentID & c) const`

Determination du nombre d'EcoAgent separant de la table en position finale.

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

4.14.2.4 `int PlateformeMondeDesCubes::getNombreDeCubes () const`

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

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

Returns:

le nombre de cubes

4.14.2.5 `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*`

4.14.2.6 `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](#).

4.14.2.7 `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

4.14.2.8 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

4.14.2.9 void PlateformeMondeDesCubes::resoudre () [virtual]

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

Methode qui permet de realiser une etape et une seule de la resolution du probleme du monde des cubes par eco-resolution

Implements [PlateformeEcoResolution](#).

4.14.2.10 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

4.14.2.11 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

4.14.2.12 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

4.14.2.13 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

4.14.2.14 string PlateformeMondeDesCubes::toString () const

Methode qui permet d'obtenir un affichage minimal de l'agencement des cubes.

Methode qui permet d'obtenir un affichage minimal de l'agencement des cubes

Returns:

- La chaine de caractere de type string a afficher

4.14.3 Friends And Related Function Documentation**4.14.3.1 ostream& operator<< (ostream & f, const PlateformeMondeDesCubes & p) [friend]**

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.

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

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

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

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

4.15.1 Detailed Description

Cette classe abstraite est une squelette pour une regle.

Cette classe abstraite est une squelette pour une regle qui sera ajoutée dans une [PlateformeEcoResolution](#). Une regle est une condition ou un ensemble de conditions qui doivent être vérifiées pour que la résolution puisse se faire.

4.15.2 Member Function Documentation

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

Vérification de la regle.

Returns:

true si la regle est vérifiée, false sinon (ou bien une exception en fonction des cas)

Implemented in [AucuneSurcharge](#).

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

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

4.16 reliesATable Class Reference

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

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

4.16.1 Detailed Description

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

Cette regle verifie que tous les cubes sont relies a la table a l'initialisation de la resolution ainsi que dans l'etat final de la resolution.

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

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

4.17 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.

4.17.1 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.

4.17.2 Constructor & Destructor Documentation

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

Constructeur.

Constructeur de la classe [Singleton](#).

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

Destructeur.

Destructeur de la classe [Singleton](#).

4.17.3 Member Function Documentation

4.17.3.1 `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

4.17.3.2 `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`

4.18 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 Cette procedure ne fait rien car la table ne fuit jamais.
- void [rechercherSatisfaction](#) ()
Suite d'operations realisees par la table lorsqu'elle cherche a se satisfaire Cette procedure ne fait rien car la table est toujours satisfaite.
- void [agresser](#) ([EcoAgent](#) &a)
Suite d'operations realisees lorsque la table agresse un autre eco-agent Cette procedure ne fait rien car la table n'agresse aucun autre [EcoAgent](#).
- void [estAgresse](#) ()
Suite d'operations realisees par la table lorsqu'elle est agressee Cette procedure ne fait rien car la table ne reagit pas lorsqu'elle est agresse.
- void [faireFuite](#) ()
Suite d'operations realisees par la fuite lorsqu'elle fuit Cette procedure ne fait rien car la table ne peut pas fuir.
- void [faireSatisfaction](#) ()
Suite d'operations realisees par la table lorsqu'elle se satisfait Cette procedure ne fait rien car la table est toujours satisfaite.
- void [agir](#) ()
Suite d'operations realisees par la table lorsqu'il agit Cette procedure ne fait rien car la table n'agit jamais. C'est juste un support.
- void [initialiser](#) ()
Initialisation de la table avant l'EcoResolution Cette procedure ne fait rien, la table etant toujours satisfaite.

4.18.1 Detailed Description

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

Classe derivee de la classe [EcoAgent](#) designant le support sur lequel vont etre poses les cubes. La table est suppose de capacite infinie. C'est a dire qu'il est possible de poser autant de cubes qu'on veut dessus. La [Table](#) est toujours satisfaite.

4.18.2 Constructor & Destructor Documentation

4.18.2.1 `Table::Table ()`

Constructeur.

Constructeur de la classe [Table](#) par default. Un [EcoAgentID](#) sera automatiquement genere pour la table. L'Etat de la table est fixe a SATISFAIT.

4.18.2.2 `Table::Table (EcoAgentID & id)`

Constructeur.

Constructeur de la classe [Table](#)

Parameters:

id : identifiant unique que la table se verra attribuer

4.18.2.3 `Table::~~Table ()`

Destructeur.

Destructeur de la classe [Table](#)

4.18.3 Member Function Documentation

4.18.3.1 `void Table::agresser (EcoAgent & a) [virtual]`

Suite d'operations realisees lorsque la table agresse un autre eco-agent Cette procedure ne fait rien car la table n'agresse aucun autre [EcoAgent](#).

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 "Exceptions.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 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 [EcoAgent](#)

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 [EcoAgent](#).

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 [EcoAgent](#)

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/Exceptions.hpp File Reference

Liste des Exceptions possibles.

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](#).
- class [ExceptionUnCubeEstSurcharge](#)
Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes sont surcharges".
- class [ExceptionUnCubeSeraSurcharge](#)
Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes seront surcharges".
- class [ExceptionIlexisteUneBoucle](#)
Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes sont relies a la table".
- class [ExceptionCubeNonRelie](#)
Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes sont relies a la table".
- class [ExceptionIlexisteraUneBoucle](#)
Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes seront relies a la table".
- class [ExceptionCubeSeraNonRelie](#)
Exception lancee dans le cadre de la verification de la regle "est-ce que les cubes seront relies a la table".

5.7.1 Detailed Description

Liste des Exceptions possibles.

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 "Exceptions.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 <string>
#include <algorithm>
```

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

Cette regle verifie que tous les cubes sont relies a la table a l'initialisation de la resolution ainsi que dans l'etat final de la resolution.

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

Include dependency graph for reliesATable.hpp:

5.11.1 Detailed Description

Cette regle verifie que tous les cubes sont relies a la table a l'initialisation de la resolution ainsi que dans l'etat final de la resolution.

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](#). La [Table](#) est le support sur lequel sont poses les cubes.

Index

- ~Cube
 - Cube, [11](#)
- ~EcoAgentID
 - EcoAgentID, [19](#)
- ~PlateformeEcoResolution
 - PlateformeEcoResolution, [28](#)
- ~Singleton
 - Singleton, [39](#)
- ~Table
 - Table, [42](#)
- addEcoAgent
 - PlateformeEcoResolution, [28](#)
 - PlateformeMondeDesCubes, [33](#)
- addRegle
 - PlateformeEcoResolution, [28](#)
- agresser
 - Cube, [11](#)
 - EcoAgent, [15](#)
 - Table, [42](#)
- AucuneSurcharge, [7](#)
 - pasSurcharges, [7](#)
 - serontPasSurcharges, [8](#)
 - verifier, [8](#)
- compareEcoAgentID, [9](#)
- Cube, [10](#)
 - ~Cube, [11](#)
 - agresser, [11](#)
 - Cube, [11](#)
 - initialiser, [12](#)
- distanceATable
 - PlateformeMondeDesCubes, [33](#)
- distanceFinaleATable
 - PlateformeMondeDesCubes, [33](#)
- EcoAgent, [13](#)
 - agresser, [15](#)
 - EcoAgent, [14](#)
 - getEtat, [15](#)
 - getId, [15](#)
 - getPositionCourante, [15](#)
 - getPositionFinale, [15](#)
 - getVisite, [15](#)
 - initialiser, [16](#)
 - setEtat, [16](#)
 - setId, [16](#)
 - setPositionCourante, [16](#)
 - setPositionFinale, [16](#)
 - setVisite, [17](#)
- EcoAgentID, [18](#)
 - ~EcoAgentID, [19](#)
 - EcoAgentID, [19](#)
 - getId, [19](#)
 - operator<, [19](#)
 - operator==, [19](#)
- ecoagents
 - PlateformeEcoResolution, [31](#)
- ExceptionCubeNonRelie, [20](#)
- ExceptionCubeSeraNonRelie, [21](#)
- ExceptionEcoAgentDejaEnregistre, [22](#)
- ExceptionIlexisteraUneBoucle, [23](#)
- ExceptionIlexisteUneBoucle, [24](#)
- ExceptionUnCubeEstSurcharge, [25](#)
- ExceptionUnCubeSeraSurcharge, [26](#)
- getEcoAgent
 - PlateformeEcoResolution, [29](#)
- getEcoAgentAuDessus
 - PlateformeEcoResolution, [29](#)
- getEcoAgents
 - PlateformeEcoResolution, [29](#)
- getEtat
 - EcoAgent, [15](#)
- getId
 - EcoAgent, [15](#)
 - EcoAgentID, [19](#)
- getInstance
 - Singleton, [40](#)
- getNombreDeCubes
 - PlateformeMondeDesCubes, [34](#)
- getPositionCourante
 - EcoAgent, [15](#)
- getPositionFinale
 - EcoAgent, [15](#)
- getRegles
 - PlateformeEcoResolution, [30](#)
- getTableID
 - PlateformeMondeDesCubes, [34](#)
- getVisite

- EcoAgent, 15
- initialiser
 - Cube, 12
 - EcoAgent, 16
 - PlateformeEcoResolution, 30
 - PlateformeMondeDesCubes, 34
- kill
 - Singleton, 40
- nombreEcoAgentAuDessus
 - PlateformeEcoResolution, 30
- obtenirCubePrioritaire
 - PlateformeMondeDesCubes, 34
- obtenirGeneur
 - PlateformeMondeDesCubes, 34
- operator<
 - EcoAgentID, 19
- operator<<
 - PlateformeMondeDesCubes, 36
- operator==
 - EcoAgentID, 19
- pasSurcharges
 - AucuneSurcharge, 7
- PlateformeEcoResolution, 27
 - ~PlateformeEcoResolution, 28
 - addEcoAgent, 28
 - addRegle, 28
 - ecoagents, 31
 - getEcoAgent, 29
 - getEcoAgentAuDessus, 29
 - getEcoAgents, 29
 - getRegles, 30
 - initialiser, 30
 - nombreEcoAgentAuDessus, 30
 - PlateformeEcoResolution, 28
 - regles, 31
 - resoudre, 30
 - sontSatisfait, 30
 - verifierCoherence, 31
- PlateformeMondeDesCubes, 32
 - addEcoAgent, 33
 - distanceATable, 33
 - distanceFinaleATable, 33
 - getNombreDeCubes, 34
 - getTableID, 34
 - initialiser, 34
 - obtenirCubePrioritaire, 34
 - obtenirGeneur, 34
 - operator<<, 36
 - resoudre, 35
 - setCubeID, 35
 - setPositionCourante, 35
 - setPositionFinale, 35
 - setTableIdentifiant, 36
 - toString, 36
- Regle, 37
 - verifier, 37
- regles
 - PlateformeEcoResolution, 31
- reliesATable, 38
- resoudre
 - PlateformeEcoResolution, 30
 - PlateformeMondeDesCubes, 35
- serontPasSurcharges
 - AucuneSurcharge, 8
- setCubeID
 - PlateformeMondeDesCubes, 35
- setEtat
 - EcoAgent, 16
- setId
 - EcoAgent, 16
- setPositionCourante
 - EcoAgent, 16
 - PlateformeMondeDesCubes, 35
- setPositionFinale
 - EcoAgent, 16
 - PlateformeMondeDesCubes, 35
- setTableIdentifiant
 - PlateformeMondeDesCubes, 36
- setVisite
 - EcoAgent, 17
- Singleton, 39
 - ~Singleton, 39
 - getInstance, 40
 - kill, 40
 - Singleton, 39
- sontSatisfait
 - PlateformeEcoResolution, 30
- Table, 41
 - ~Table, 42
 - agresser, 42
 - Table, 42
- toString
 - PlateformeMondeDesCubes, 36
- trunk/include/aucuneSurcharge.hpp, 43
- trunk/include/compareEcoAgentID.hpp, 44
- trunk/include/cube.hpp, 45
- trunk/include/ecoAgent.hpp, 46
- trunk/include/ecoAgentID.hpp, 47
- trunk/include/etat.hpp, 48
- trunk/include/Exceptions.hpp, 49
- trunk/include/plateformeEcoResolution.hpp, 50

trunk/include/plateformeMondeDesCubes.hpp, [51](#)

trunk/include/regle.hpp, [52](#)

trunk/include/reliesATable.hpp, [53](#)

trunk/include/singleton.hpp, [54](#)

trunk/include/table.hpp, [55](#)

verifier

 AucuneSurcharge, [8](#)

 Regle, [37](#)

verifierCoherence

 PlateformeEcoResolution, [31](#)