

Monde des cubes

0.1

Generated by Doxygen 1.5.7.1

Tue Apr 14 11:30:43 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	Cube Class Reference	7
4.1.1	Detailed Description	8
4.1.2	Constructor & Destructor Documentation	8
4.1.2.1	Cube	8
4.1.2.2	Cube	8
4.1.2.3	~Cube	8
4.1.3	Member Function Documentation	8
4.1.3.1	agresser	8
4.1.3.2	initialiserEtat	9
4.2	EcoAgent Class Reference	10
4.2.1	Detailed Description	11
4.2.2	Member Function Documentation	11
4.2.2.1	agresser	11
4.2.2.2	getEtat	11
4.2.2.3	getId	11
4.2.2.4	getPositionCourante	11
4.2.2.5	getPositionFinale	12
4.2.2.6	setEtat	12
4.2.2.7	setId	12

4.2.2.8	setPositionCourante	12
4.2.2.9	setPositionFinale	12
4.3	EcoAgentID Class Reference	13
4.3.1	Detailed Description	13
4.3.2	Constructor & Destructor Documentation	13
4.3.2.1	EcoAgentID	13
4.3.2.2	~EcoAgentID	13
4.3.3	Member Function Documentation	14
4.3.3.1	getId	14
4.3.3.2	operator<	14
4.3.3.3	operator==	14
4.4	PlateformeEcoResolution Class Reference	15
4.4.1	Detailed Description	15
4.4.2	Constructor & Destructor Documentation	15
4.4.2.1	PlateformeEcoResolution	15
4.4.2.2	~PlateformeEcoResolution	16
4.4.3	Member Function Documentation	16
4.4.3.1	addEcoAgent	16
4.4.3.2	addRegle	16
4.4.3.3	getEcoAgent	16
4.4.3.4	initialiser	16
4.4.3.5	resoudre	17
4.4.3.6	verifierCoherence	17
4.5	PlateformeMondeDesCubes Class Reference	18
4.5.1	Detailed Description	19
4.5.2	Member Function Documentation	19
4.5.2.1	addEcoAgent	19
4.5.2.2	getNombreDeCubes	19
4.5.2.3	getTableID	19
4.5.2.4	initialiser	19
4.5.2.5	obtenirCubePrioritaire	20
4.5.2.6	obtenirGeneur	20
4.5.2.7	resoudre	20
4.5.2.8	setCubeID	20
4.5.2.9	setNombreDeCubes	20
4.5.2.10	setPositionCourante	21

4.5.2.11	setPositionFinale	21
4.5.2.12	setTableIdentifiant	21
4.5.2.13	verifierNombreDeCubes	21
4.6	Regle Class Reference	22
4.6.1	Detailed Description	22
4.6.2	Member Function Documentation	22
4.6.2.1	verifier	22
4.7	Singleton< T > Class Template Reference	23
4.7.1	Detailed Description	23
4.7.2	Constructor & Destructor Documentation	23
4.7.2.1	Singleton	23
4.7.2.2	~Singleton	23
4.7.3	Member Function Documentation	24
4.7.3.1	getInstance	24
4.7.3.2	kill	24
4.8	Table Class Reference	25
4.8.1	Detailed Description	25
4.8.2	Constructor & Destructor Documentation	26
4.8.2.1	Table	26
4.8.2.2	Table	26
4.8.2.3	~Table	26
4.8.3	Member Function Documentation	26
4.8.3.1	agresser	26
5	File Documentation	27
5.1	trunk/include/cube.hpp File Reference	27
5.1.1	Detailed Description	27
5.2	trunk/include/ecoAgent.hpp File Reference	28
5.2.1	Detailed Description	28
5.3	trunk/include/ecoAgentID.hpp File Reference	29
5.3.1	Detailed Description	29
5.4	trunk/include/etat.hpp File Reference	30
5.4.1	Detailed Description	30
5.5	trunk/include/plateformeEcoResolution.hpp File Reference	31
5.5.1	Detailed Description	31
5.6	trunk/include/plateformeMondeDesCubes.hpp File Reference	32
5.6.1	Detailed Description	32

5.7	trunk/include/regle.hpp File Reference	33
5.7.1	Detailed Description	33
5.8	trunk/include/singleton.hpp File Reference	34
5.8.1	Detailed Description	34
5.9	trunk/include/table.hpp File Reference	35
5.9.1	Detailed Description	35

Chapter 1

Class Index

1.1 Class Hierarchy

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

EcoAgent	10
Cube	7
Table	25
EcoAgentID	13
PlateformeEcoResolution	15
PlateformeMondeDesCubes	18
Regle	22
Singleton< T >	23
Singleton< PlateformeMondeDesCubes >	23
PlateformeMondeDesCubes	18

Chapter 2

Class Index

2.1 Class List

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

Cube (Classe derivee de la classe EcoAgent designant un Cube)	7
EcoAgent (Classe abstraite qui permet de creer un eco-agent)	10
EcoAgentID (Identifiant unique d'un eco-agent)	13
PlateformeEcoResolution (Classe representant une plateforme d'eco-resolution abstraite)	15
PlateformeMondeDesCubes (Classe representant une plateforme d'eco-resolution appliquee au monde des cubes)	18
Regle (Cette classe abstraite est une squelette pour une regle)	22
Singleton< T > (Template de classe permettant de rendre une classe instanciable une seule fois)	23
Table (Classe derivee de la classe EcoAgent designant le Support sur lequel vont etre poses les cubes)	25

Chapter 3

File Index

3.1 File List

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

trunk/include/ cube.hpp (Implementation du module cube qui est un derive d'un EcoAgent) . . .	27
trunk/include/ ecoAgent.hpp (Mise en place de la classe abstraite EcoAgent)	28
trunk/include/ ecoAgentID.hpp (Implementation de la classe EcoAgentID)	29
trunk/include/ etat.hpp (Enumeration des etats possibles des eco-agents)	30
trunk/include/ plateformeEcoResolution.hpp (Plateforme abstraite d'eco-resolution)	31
trunk/include/ plateformeMondeDesCubes.hpp (Plateforme d'eco-resolution appliquee au monde des cubes)	32
trunk/include/ regle.hpp (Squelette d'une regle pour une plateforme d'eco-resolution)	33
trunk/include/ singleton.hpp (Implementation du design pattern singleton)	34
trunk/include/ table.hpp (Implementation du module table qui est un derive d'un EcoAgent) . . .	35

Chapter 4

Class Documentation

4.1 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](#) (const [Cube](#) &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 [initialiserEtat](#) ()
Initialisation de l'etat du cube.
- void [agir](#) ()
Suite d'operations realisees par le cube lorsqu'il agit.

4.1.1 Detailed Description

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

4.1.2 Constructor & Destructor Documentation

4.1.2.1 [Cube::Cube](#) ()

Constructeur.

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

4.1.2.2 [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

4.1.2.3 [Cube::~~Cube](#) ()

Destructeur.

Destructeur de la classe [Cube](#)

4.1.3 Member Function Documentation

4.1.3.1 void [Cube::agresser \(const Cube & a\)](#)

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

Parameters:

a : [EcoAgent](#) a agresser

4.1.3.2 void Cube::initialiserEtat ()

Initialisation de l'état du cube.

Cette methode permet d'initialiser l'état 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"

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

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

4.2 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

- [Etat](#) [getEtat](#) () const
Obtention de l'Etat.
- [EcoAgentID](#) [getId](#) () const
Obtention de l'EcoAgentID.
- void [setEtat](#) (const [Etat](#) e)
Changement de l'Etat.
- void [setId](#) (const [EcoAgentID](#) &id)
Changement de l'EcoAgentID.
- 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 [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](#) (const [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.

4.2.1 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.

4.2.2 Member Function Documentation

4.2.2.1 virtual void EcoAgent::agresser (const EcoAgent & *a*) [pure virtual]

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

Parameters:

a : [EcoAgent](#) a agresser

Implemented in [Table](#).

4.2.2.2 Etat EcoAgent::getEtat () const

Obtention de l'Etat.

Methode qui permet d'obtenir l'attribut prive Etat

Returns:

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

4.2.2.3 EcoAgentID EcoAgent::getId () const

Obtention de l'EcoAgentID.

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

Returns:

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

4.2.2.4 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

4.2.2.5 `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

4.2.2.6 `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

4.2.2.7 `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

4.2.2.8 `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

4.2.2.9 `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

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

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

4.3 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](#).

Static Public Member Functions

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

4.3.1 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

4.3.2 Constructor & Destructor Documentation

4.3.2.1 [EcoAgentID::EcoAgentID](#) ()

Constructeur.

Constructeur de la classe [EcoAgentID](#)

4.3.2.2 [EcoAgentID::~~EcoAgentID](#) ()

Destructeur.

Destructeur de la classe [EcoAgentID](#)

4.3.3 Member Function Documentation

4.3.3.1 `int EcoAgentID::getId () const`

Obtention de l'identifiant.

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

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

Comparaison de [EcoAgentID](#).

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

4.3.3.3 `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/ecoAgent.cpp](#)

4.4 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](#) (const [EcoAgent](#) &ea)
Ajout d'un eco-agent.
- void [addRegle](#) (const [Regle](#) &r)
Ajout d'une nouvelle regle.
- bool [verifierCoherence](#) ()
Verification du respect des regles apres l'initialisation de la plateforme.
- virtual void [initialiser](#) ()=0
Initialisation de la resolution.
- virtual void [resoudre](#) ()=0
Resolution du probleme par eco-resolution.

4.4.1 Detailed Description

classe representant une plateforme d'eco-resolution abstraite

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

4.4.2 Constructor & Destructor Documentation

4.4.2.1 PlateformeEcoResolution::PlateformeEcoResolution ()

Constructeur.

Constructeur de la classe abstraite [PlateformeEcoResolution](#)

4.4.2.2 PlateformeEcoResolution::~~PlateformeEcoResolution ()

Destructeur.

Destructeur de la classe abstraite [PlateformeEcoResolution](#)

4.4.3 Member Function Documentation

4.4.3.1 void PlateformeEcoResolution::addEcoAgent (const EcoAgent & *ea*)

Ajout d'un eco-agent.

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

Parameters:

ea : l'eco-agent a ajouter

4.4.3.2 void PlateformeEcoResolution::addRegle (const Regle & *r*)

Ajout d'une nouvelle regle.

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

Parameters:

r : la regle a ajouter

4.4.3.3 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

4.4.3.4 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](#).

4.4.3.5 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](#).

4.4.3.6 bool PlateformeEcoResolution::verifierCoherence ()

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

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

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

4.5 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.
- [Cube](#) * [obtenirCubePrioritaire](#) ()
Obtention du cube qui a la priorite pour agir dans la plateforme d'eco-resolution.
- 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](#) ()
Obtention de l'identifiant de la table.
- bool [verifierNombreDeCubes](#) ()
Verification de la coherence du nombre de cubes (s'il est bien positif).
- [Cube](#) * [obtenirGeneur](#) (const [Cube](#) ¤tCube)
Obtention du geneur du cube courant.
- void [addEcoAgent](#) (const [Cube](#) &cube)
Ajout d'un cube dans la plateforme.
- void [setPositionFinale](#) (const [EcoAgentID](#) &c, const [EcoAgentID](#) &pos)
Determination de la position finale d'un cube de la plateforme.
- void [setPositionCourante](#) (const [EcoAgentID](#) &c, const [EcoAgentID](#) &pos)
Determination de la position courante d'un cube de la plateforme.
- void [setCubeID](#) (const [Cube](#) ¤tCube, const [EcoAgentID](#) &id)
Determination de l'identifiant d'un cube.

Friends

- class [Singleton](#)< [PlateformeMondeDesCubes](#) >

4.5.1 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.

4.5.2 Member Function Documentation

4.5.2.1 void PlateformeMondeDesCubes::addEcoAgent (const Cube & *cube*)

Ajout d'un cube dans la plateforme.

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

Parameters:

cube : le cube a ajouter

4.5.2.2 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

4.5.2.3 EcoAgentID PlateformeMondeDesCubes::getTableID ()

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](#))

4.5.2.4 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](#).

4.5.2.5 Cube* PlateformeMondeDesCubes::obtenirCubePrioritaire ()

Obtention du cube qui a la priorite pour agir dans la plateforme d'eco-resolution.

Methode qui permet d'obtenir le cube qui doit agir en priorite dans la plateforme d'eco-resolution

Returns:

un pointeur sur le [Cube](#) prioritaire s'il existe, NULL sinon

4.5.2.6 Cube* PlateformeMondeDesCubes::obtenirGeneur (const Cube & *currentCube*)

Obtention du geneur du cube courant.

Methode qui permet d'obtenir le geneur du cube courant

Parameters:

currentCube : le cube courant

Returns:

un pointeur sur le cube geneur s'il existe, NULL sinon

4.5.2.7 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](#).

4.5.2.8 void PlateformeMondeDesCubes::setCubeID (const Cube & *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

4.5.2.9 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

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

Determination de la position courante d'un cube de la plateforme.

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

Parameters:

c : identifiant du cube a positionner

pos : position courante du cube, ie identifiant du cube destinataire

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

Determination de la position finale d'un cube de la plateforme.

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

Parameters:

c : identifiant du cube a positionner

pos : position finale du cube, ie identifiant du cube destinataire

4.5.2.12 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

4.5.2.13 bool PlateformeMondeDesCubes::verifierNombreDeCubes ()

Verification de la coherence du nombre de cubes (s'il est bien positif).

Methode qui permet de verifier si le nombre des cubes est bien strictement positif

Returns:

true si le nombre des cubes est strictement positif, false sinon

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

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

4.6 Regle Class Reference

Cette classe abstraite est une squelette pour une regle.

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

Public Member Functions

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

4.6.1 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](#)

4.6.2 Member Function Documentation

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

Verification de la regle.

Returns:

true si la regle est verifiee, false sinon

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

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

4.7 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.

4.7.1 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.

4.7.2 Constructor & Destructor Documentation

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

Constructeur.

Constructeur de la classe [Singleton](#).

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

Destructeur.

Destructeur de la classe [Singleton](#).

4.7.3 Member Function Documentation

4.7.3.1 `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

4.7.3.2 `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`

4.8 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](#) (const [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](#) (const [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.

4.8.1 Detailed Description

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

4.8.2 Constructor & Destructor Documentation

4.8.2.1 Table::Table ()

Constructeur.

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

4.8.2.2 Table::Table (const EcoAgentID & id)

Constructeur.

Constructeur de la classe [Table](#)

Parameters:

id : identifiant unique que l'agent se verra attribuer

4.8.2.3 Table::~~Table ()

Destructeur.

Destructeur de la classe [Table](#)

4.8.3 Member Function Documentation

4.8.3.1 void Table::agresser (const 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 file:

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

Chapter 5

File Documentation

5.1 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 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](#).

5.1.1 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](#).

5.2 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.

5.2.1 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

5.3 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.

5.3.1 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

5.4 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.

5.4.1 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

5.5 trunk/include/plateformeEcoResolution.hpp File Reference

Plateforme abstraite d'eco-resolution.

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

5.5.1 Detailed Description

Plateforme abstraite d'eco-resolution.

Author:

GDD
Pauline Requena
Arnaud Faure

Version:

0.1

Date:

29 mars 2009

5.6 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:

Classes

- class [PlateformeMondeDesCubes](#)

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

5.6.1 Detailed Description

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

Author:

GDD
Pauline Requena
Arnaud Faure

Version:

0.1

Date:

02 avril 2009

5.7 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.

5.7.1 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

5.8 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.

5.8.1 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.

5.9 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.

5.9.1 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, [8](#)
- ~EcoAgentID
 - EcoAgentID, [13](#)
- ~PlateformeEcoResolution
 - PlateformeEcoResolution, [15](#)
- ~Singleton
 - Singleton, [23](#)
- ~Table
 - Table, [26](#)
- addEcoAgent
 - PlateformeEcoResolution, [16](#)
 - PlateformeMondeDesCubes, [19](#)
- addRegle
 - PlateformeEcoResolution, [16](#)
- agresser
 - Cube, [8](#)
 - EcoAgent, [11](#)
 - Table, [26](#)
- Cube, [7](#)
 - ~Cube, [8](#)
 - agresser, [8](#)
 - Cube, [8](#)
 - initialiserEtat, [8](#)
- EcoAgent, [10](#)
 - agresser, [11](#)
 - getEtat, [11](#)
 - getId, [11](#)
 - getPositionCourante, [11](#)
 - getPositionFinale, [11](#)
 - setEtat, [12](#)
 - setId, [12](#)
 - setPositionCourante, [12](#)
 - setPositionFinale, [12](#)
- EcoAgentID, [13](#)
 - ~EcoAgentID, [13](#)
 - EcoAgentID, [13](#)
 - getId, [14](#)
 - operator<, [14](#)
 - operator==, [14](#)
- getEcoAgent
 - PlateformeEcoResolution, [16](#)
- getEtat
 - EcoAgent, [11](#)
- getId
 - EcoAgent, [11](#)
 - EcoAgentID, [14](#)
- getInstance
 - Singleton, [24](#)
- getNombreDeCubes
 - PlateformeMondeDesCubes, [19](#)
- getPositionCourante
 - EcoAgent, [11](#)
- getPositionFinale
 - EcoAgent, [11](#)
- getTableID
 - PlateformeMondeDesCubes, [19](#)
- initialiser
 - PlateformeEcoResolution, [16](#)
 - PlateformeMondeDesCubes, [19](#)
- initialiserEtat
 - Cube, [8](#)
- kill
 - Singleton, [24](#)
- obtenirCubePrioritaire
 - PlateformeMondeDesCubes, [19](#)
- obtenirGeneur
 - PlateformeMondeDesCubes, [20](#)
- operator<
 - EcoAgentID, [14](#)
- operator==
 - EcoAgentID, [14](#)
- PlateformeEcoResolution, [15](#)
 - ~PlateformeEcoResolution, [15](#)
 - addEcoAgent, [16](#)
 - addRegle, [16](#)
 - getEcoAgent, [16](#)
 - initialiser, [16](#)
 - PlateformeEcoResolution, [15](#)
 - resoudre, [16](#)
 - verifierCoherence, [17](#)
- PlateformeMondeDesCubes, [18](#)
 - addEcoAgent, [19](#)
 - getNombreDeCubes, [19](#)

- getTableID, [19](#)
- initialiser, [19](#)
- obtenirCubePrioritaire, [19](#)
- obtenirGeneur, [20](#)
- resoudre, [20](#)
- setCubeID, [20](#)
- setNombreDeCubes, [20](#)
- setPositionCourante, [20](#)
- setPositionFinale, [21](#)
- setTableIdentifiant, [21](#)
- verifierNombreDeCubes, [21](#)
- Regle, [22](#)
 - verifier, [22](#)
- resoudre
 - PlateformeEcoResolution, [16](#)
 - PlateformeMondeDesCubes, [20](#)
- setCubeID
 - PlateformeMondeDesCubes, [20](#)
- setEtat
 - EcoAgent, [12](#)
- setId
 - EcoAgent, [12](#)
- setNombreDeCubes
 - PlateformeMondeDesCubes, [20](#)
- setPositionCourante
 - EcoAgent, [12](#)
 - PlateformeMondeDesCubes, [20](#)
- setPositionFinale
 - EcoAgent, [12](#)
 - PlateformeMondeDesCubes, [21](#)
- setTableIdentifiant
 - PlateformeMondeDesCubes, [21](#)
- Singleton, [23](#)
 - ~Singleton, [23](#)
 - getInstance, [24](#)
 - kill, [24](#)
 - Singleton, [23](#)
- Table, [25](#)
 - ~Table, [26](#)
 - agresser, [26](#)
 - Table, [26](#)
- trunk/include/cube.hpp, [27](#)
- trunk/include/ecoAgent.hpp, [28](#)
- trunk/include/ecoAgentID.hpp, [29](#)
- trunk/include/etat.hpp, [30](#)
- trunk/include/plateformeEcoResolution.hpp, [31](#)
- trunk/include/plateformeMondeDesCubes.hpp, [32](#)
- trunk/include/regle.hpp, [33](#)
- trunk/include/singleton.hpp, [34](#)
- trunk/include/table.hpp, [35](#)
- verifier