



Institut Géographique National  
Laboratoire COGIT



## Développer avec le module 3D de GeOxygene Version 1.0

Date de la dernière modification	21 Septembre 2012
Rédacteur	Mickaël BRASEBIN – IGN / Laboratoire COGIT mickael.brasebin@ign.fr
Diffusion	Libre
Contenu document	du Documentation pour développer avec le module 3D de GeOxygene

# TABLE DES MATIERES

<b>1</b>	<b>INTRODUCTION .....</b>	<b>3</b>
1.1	INTRODUCTION AU MODULE 3D .....	3
1.2	EXECUTION DE L'APPLICATION .....	4
<b>2</b>	<b>CONCEPT GENERAUX.....</b>	<b>5</b>
2.1	LE SCHEMA GEOGRAPHIQUE .....	5
2.2	MODELE GEOMETRIQUE .....	5
2.3	MODELE DE REPRESENTATION.....	7
2.3.1	<i>Représentation des entités.....</i>	7
2.3.2	<i>Représentation d'une couche .....</i>	9
2.3.3	<i>Modélisation d'une carte.....</i>	10
2.4	L'INTERFACE GRAPHIQUE .....	11
<b>3</b>	<b>LES PACKAGES .....</b>	<b>12</b>
3.1	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.....	12
3.2	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.CALCULATION.....	12
3.3	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.CONVERSION .....	13
3.4	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.EQUATION.....	13
3.5	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.GEOMETRY .....	13
3.6	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.GEOMETRY.TOPOLOGY .....	13
3.7	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.GUI .....	13
3.8	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.GUI.ACTIONPANELMENU .....	13
3.9	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.GUI.FILTER.....	13
3.10	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.GUI.MENU .....	13
3.11	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.IHM.NAVIGATION3D.....	14
3.12	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.GUI.RENDERER .....	14
3.13	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.GUI.TABLE .....	14
3.14	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.GUI.WINDOW .....	14
3.15	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.IO .....	14
3.16	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.REPRESENTATION .....	14
3.17	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.REPRESENTATION.BASIC .....	14
3.18	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.REPRESENTATION.CONTROL .....	15
3.19	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.REPRESENTATION.MODELLINGFILE .....	15
3.20	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.REPRESENTATION.SYMBOL .....	15
3.21	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.REPRESENTATION.TEXTURE .....	15
3.22	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.REPRESENTATION.TOPONYM .....	15
3.23	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.SAMPLE.....	15
3.24	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.SEMANTIC. ....	15
3.25	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.SIMPLIFICATION.....	16
3.26	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.TETRAEDRISTION.....	16
3.27	PACKAGE FR.IGN.COGIT.GEOXYGENE.SIG3D.UTIL .....	16
<b>4</b>	<b>EXEMPLE DE CODE .....</b>	<b>16</b>
4.1	CLASSE ADDMENU .....	16
4.2	CLASSE DISPLAYDATA .....	16
4.3	CLASSE DTMDISPLAY .....	16
4.4	CLASSE RGE.....	17
4.5	CLASSE SYMBOLOGY .....	18
4.6	CLASSE TOPONYM .....	18
<b>5</b>	<b>BIBLIOGRAPHIE .....</b>	<b>18</b>

# 1 Introduction

## 1.1 Introduction au module 3D

Le module 3D a été conçu par le laboratoire COGIT afin de permettre le développement et l'intégration de ses recherches dans le domaine de la 3D. Ce module fait partie du projet GeOxygene dont il étend le modèle afin de permettre l'utilisation et la représentation d'objets 3D. Le projet GeOxygene et son module 3D sont disponibles sur le site internet du projet GeOxygene : <http://oxygene-project.sourceforge.net/>. La Javadoc est également accessible à partir de cette adresse.

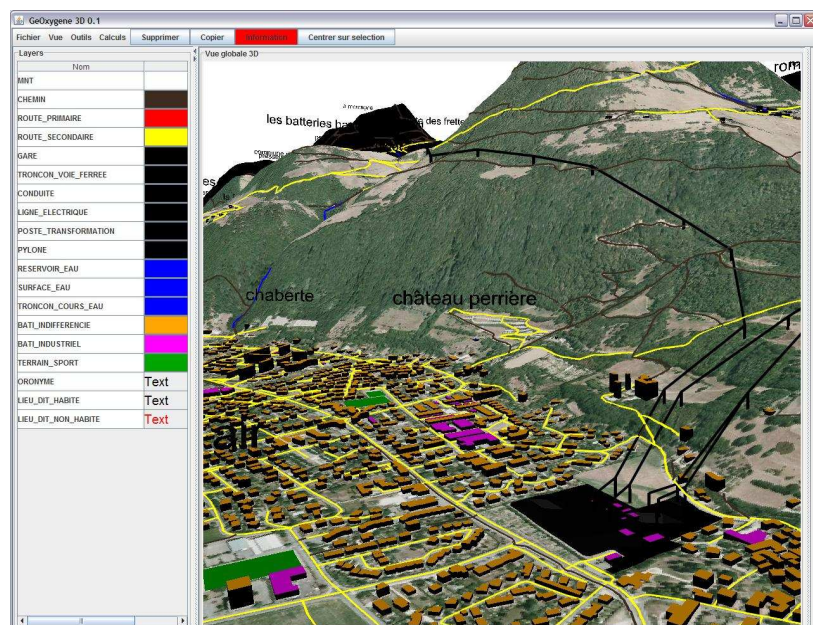
Deux documents sont fournis pour l'utilisation de ce module 3D :

- un document utilisateur pour expliquer le fonctionnement du viewer du module
- un document développeur pour expliquer comment utiliser le module 3D comme une bibliothèque de développement.

Le viewer du module 3D permet de :

- Charger et exporter des données 3D dans différents formats (Shapefile, CityGML, PostGIS),
- Naviguer en 3D (Déplacement, animation de la caméra),
- Représenter ces données (Gestion des couleurs, de la transparence, représentations d'informations textuelles, de sémiologie),
- Effectuer des opérations géométriques sur des solides (Opérations booléennes, centre de gravité etc.)

Le module fournit une base de développement. Il est possible d'exploiter les modèles et fonctionnalités du module afin d'accroître ses possibilités (Nouvelles fonctions, nouveaux chargeurs, nouvelles représentations etc.). Il est conseillé de lire le document utilisateur avant ce document afin de cerner les différentes capacités du module.



**Figure 1 : Exemple d'affichage dans GeOxygene 3D**

Pour bien comprendre le document (notamment le paragraphe sur la représentation), il est nécessaire d'avoir quelques notions de Java 3D (La documentation se trouve sur le site du projet : <https://java3d.dev.java.net/>).

## 1.2 Exécution de l'application

Avant d'exécuter l'application, veuillez vérifier que la machine virtuelle Java 1.6 ou supérieure est installée dans votre environnement.

Si ce n'est pas le cas, il est possible de la télécharger à l'adresse suivante : <http://java.sun.com/javase/downloads/index.jsp>

Pour pouvoir exécuter tout code en rapport avec le module, il est nécessaire de passer en argument de la machine virtuelle la ligne de paramètres:

```
-Xms1156m      -Xmx1156m      -XX:PermSize=256m      -XX:MaxPermSize=256m      -
Djava.library.path=./lib/native_libraries/windows-i586/
```

Elle permet d'augmenter la mémoire disponible par l'application et d'indiquer où se trouvent les librairies natives utilisées par l'application.

La classe « Launcher » à la base du module permet d'afficher le visualisateur afin d'obtenir un projet vide ou d'afficher l'une des démonstrations proposées. Le code de ces démonstrations sera présenté dans le chapitre « Exemple de code ». Il peut être nécessaire de changer certains chemins d'accès, si les données liées aux exemples n'ont pas été récupérées.

Pour ouvrir directement une fenêtre vide, il suffit exécuter la classe « fr.ign.cogit.geoxygene.sig3d.gui.MainWindow ».

## 2 Concept généraux

### 2.1 Le schéma géographique

Il s'agit du schéma géographique du module GeOxygene. Il est décrit dans le guide utilisateur [GeOxygene, 2005].

### 2.2 Modèle géométrique

Le modèle géométrique est une extension du modèle de GeOxygene (voir guide utilisateur de GeOxygene [GeOxygene, 2005]).

On peut réutiliser dans le module 3D toutes les primitives géométriques implémentées dans GeOxygene. Cependant, il est indispensable de définir les géométries en 3 dimensions (Utilisation du X, du Y et du Z lors de l'implémentation des géométries). Si ce n'est pas le cas, les géométries devraient être rejetées par l'application (Voir exemple : Classe DisplayData).

En plus des primitives déjà implémentés dans la version 2D, les primitives GM\_Solid et GM\_MultiSolid permettant de représenter les volumes sont désormais utilisables. Deux possibilités pour instancier un GM\_Solid :

- Soit par la définition de sa frontière (classe GM\_SolidBoundary),
- Soit en donnant une liste de GM\_Polygon définissant l'extérieur.  
Dans ce cas, il n'est possible de modéliser que des solides à trous n'ayant que des faces extérieures (type « La Défense ») mais pas de solide contenant des trous (type « gruyère »).

Certaines classes du module permettent de créer des solides de diverses manières :

- par opérations booléennes (module calculation, Cf. § 3.2 : Package fr.ign.cogit.geoxygene.sig3d.calculations)
- par extrusion d'un polygone (module conversion, Cf. § 3.3 : Package fr.ign.cogit.geoxygene.sig3d.conversion).



## 2.3 Modèle de représentation

Les classes gérant la représentation se trouvent dans le package : « fr.ign.cogit.geoxygene.sig3d.representation » (Cf. § 3.16 ).

Ces classes permettent de faire le lien entre le modèle géographique de GeOxygène et le modèle de représentation de Java3D. La gestion du graphe de scène est cachée et rendue automatique grâce aux classes implémentées. Cependant, il est nécessaire de la connaître si l'on souhaite étendre ces classes afin de proposer de nouvelles manières de représenter les données.

Pour en savoir plus sur Java3D et les graphes de scène, vous pouvez consulter ce lien : <http://f.armel.free.fr/tutj3d/index.php>

### 2.3.1 Représentation des entités

Figure 3 présente le modèle de représentation. Chaque entité est rattachée à une représentation 3D (Classe héritant de l'interface *I3DRepresentation*) grâce à la méthode *setRepresentation()* de la classe *FT\_Feature*. Cette représentation sert à faire le lien entre une entité de GeOxygene et une branche de scène de Java3D (classe *BranchGroup*).

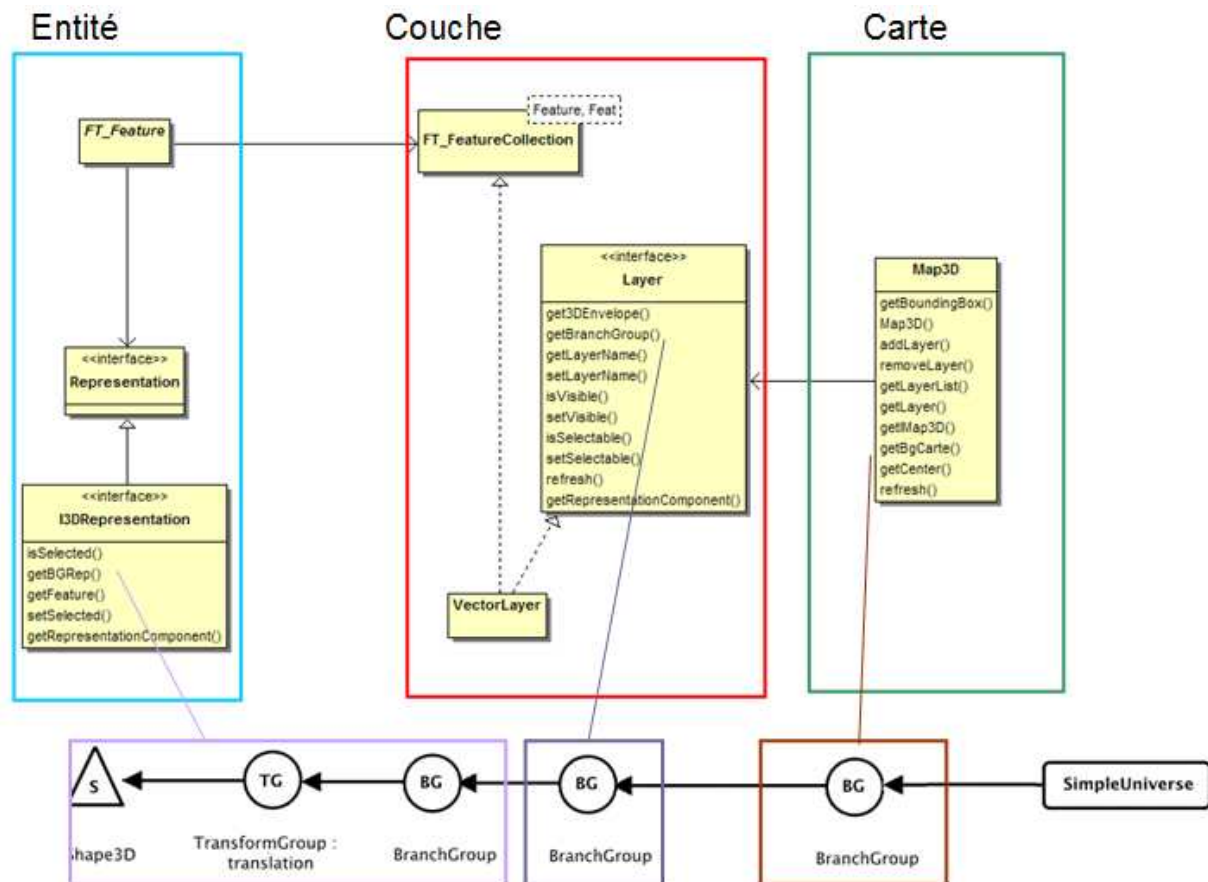
Il est possible de définir une représentation personnalisée en implémentant *I3DRepresentation*. Il est nécessaire de définir les méthodes suivantes :

- *isSelected* : indique si un objet est sélectionné ou non,
- *getBGRep* : indique quel objet de la classe *BranchGroup* (classe Java3D) sert à représenter l'objet,
- *getFeature* : indique à quelle entité est liée cette représentation,
- *setSelected* : permet d'indiquer quel événement appliquer lors de la sélection de l'objet,
- *getRepresentationComponent* : renvoie un objet *Component* Java. Il est utilisé pour représenter le style de l'objet dans la table de contenu de l'interface graphique.

La classe abstraite *Default3DRep* propose un début d'implémentation de l'interface *I3DRepresentation*.

- Un constructeur par défaut *Default3DRep()* permet de renseigner les autorisations nécessaires au *BranchGroup* qui servira à représenter dans Java3D l'entité,
- Les fonctions *setSelected* et *isSelected* sont gérées. La sélection aura pour conséquence d'entourer l'objet d'une sphère cyan translucide,
- Des accesseurs sont également implémentés, ils renvoient les informations stockées dans des attributs protégés (à compléter comme l'entité liée à la représentation et le *BranchGroup* qui est de base vide de toute information géométrique),

- La méthode *getRepresentationComponent* renvoie un Label contenant le nom de la classe de la représentation.



**Figure 3 : Schéma de représentation des entités géographiques et lien avec le graphe de scène.**

Plusieurs implémentations standards de « I3DRepresentation » sont disponibles dans les sous packages de « fr.ign.cogit.geoxygene.sig3d.representation ». Elles permettent de créer les branches de scène correspondant à une entité à l'aide de quelques paramètres. Il s'agit des classes :

- ObjectXD (X =0,1, 2,3) [sous-package basic] : permet de créer une représentation pour toutes les géométries GeoOxygene définies en 3D. Ces représentations prennent en compte : la couleur, la transparence et le type de représentation de l'objet (de manière solide ou filaire).
- RepresentationModel [sous-package modellingfile] : permet de créer une représentation pour les objets ponctuels en utilisant un fichier de modélisation 3D. Les fichiers .obj et .3DS sont gérés.
- Symbol2D ou Symbol3D [sous-package symbol] : permet de créer une représentation des objets ponctuels sous la forme d'un carré ou d'un cube sur lequel est plaquée une image. L'image et les dimensions sont paramétrables.
- TexturedSurface [sous-package texture] : permet d'utiliser la texturation générique d'objets (pour les objets possédant une surface).



- `ToponymeBasique3D` [sous-package `toponym`] : permet d'associer une représentation de type toponyme 3D à un objet ponctuel. Sont paramétrables : la police, l'orientation (possibilité d'avoir une orientation toujours face à la caméra), la couleur, la taille, la transparence ...

### 2.3.2 Représentation d'une couche

Pour afficher une liste d'objets dans une carte, il est nécessaire de les regrouper dans une couche (Objet dans une classe implémentant « Layer », package `fr.ign.cogit.geoxygene.sig3d.semantic`). Ces couches sont également associées à une branche de scène dans Java3D. Cela permet de bénéficier d'une gestion de l'affichage au niveau de la couche telle que l'on trouve dans les SIG « classiques » : rendre une couche sélectionnable, afficher/cacher une couche etc.

L'interface « Layer » oblige l'implémentation des fonctions suivantes :

- *getEnvelope3D* : renvoie un objet de la classe « `fr.ign.cogit.geoxygene.sig3d.geometry.Box3D` » qui indique les coordonnées maximales de la scène,
- *getBranchGroup* : renvoie la branche de la scène correspondant à la couche,
- *getLayerName* et *setLayerName* : gèrent le nom de la couche,
- *isVisible* et *setVisible* : gèrent les informations de visibilité d'une couche,
- *isSelectable* et *setSelectable* : influent sur la sélectabilité d'une couche,
- *refresh* : met à jour les informations de représentation d'une couche,
- *getRepresentationComponent* : renvoie un objet Component Java. Il est utilisé pour représenter le style de l'objet dans la table de contenu de l'interface graphique.

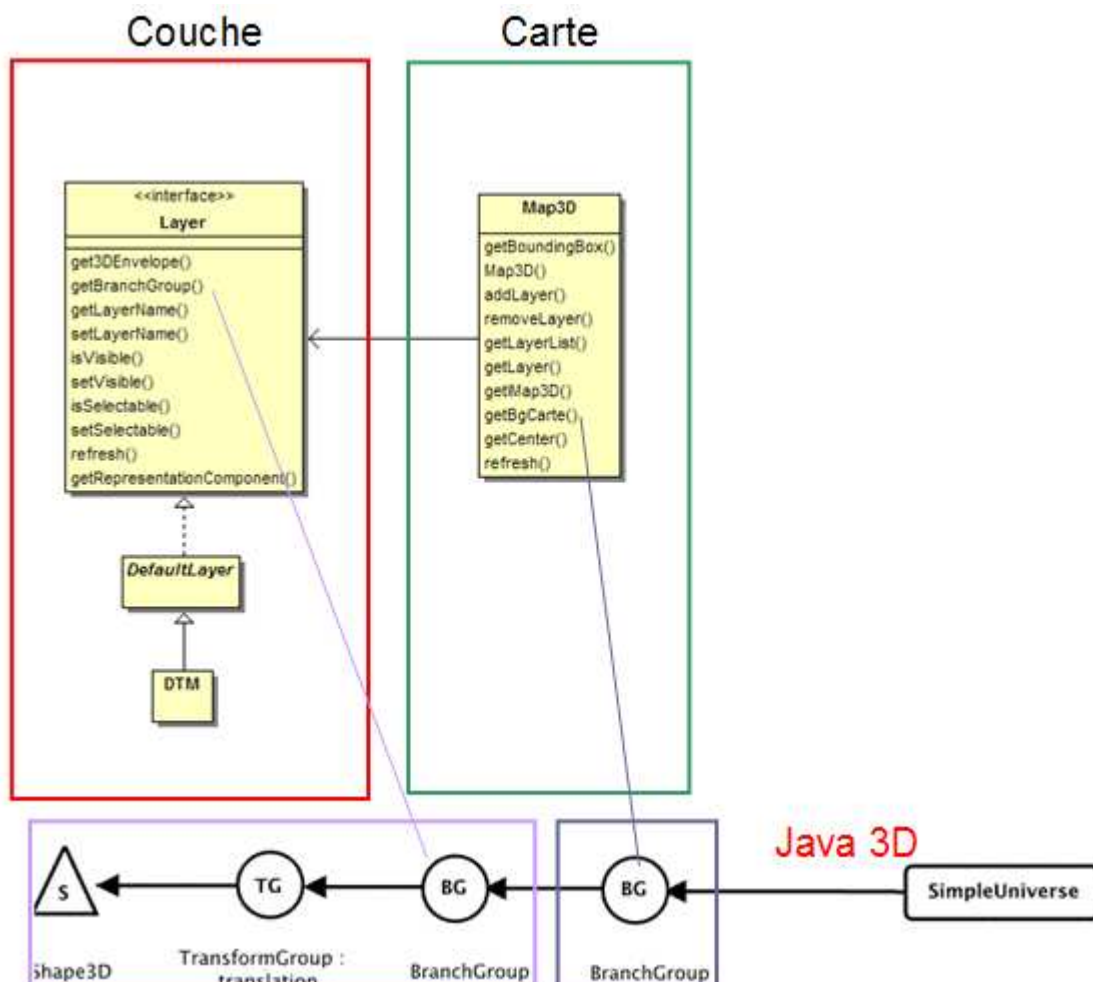
Dans le même package, la classe abstraite « `DefaultLayer` » implémente certaines de ces méthodes :

- *DefaultLayer* : propose un constructeur permettant de créer les autorisations nécessaires relatives au `BranchGroup` de la couche,
- *getLayerName*, *isVisible*, *setVisible*, *isSelectable* et *setSelectable* sont gérées,
- La méthode *getRepresentationComponent* renvoie un Label contenant le nom de la classe de la représentation.

Deux implémentations de la classe « Layer » sont disponibles :

-VectorLayer : cette classe étend FT\_FeatureCollection. Il s'agit d'une couche d'objets vectoriels. Elle permet de définir une liste d'objets et de les ranger dans une même couche comme dans les SIG classiques. Dans ce cas, afin répercuter la gestion de l'affichage de la couche aux objets la constituant, les branches de scène des objets appartenant à la couche sont rattachés à la branche de scène de la couche. L'implémentation proposée de la méthode *getRepresentationComponent* fait appel à la méthode *getRepresentationComponent* du premier objet de la couche,

- DTM : cette classe permet de gérer l'affichage de MNT. Figure 4 montre le mode de représentation pour un MNT : aucune entité n'est définie, on représente directement la branche de scène du MNT. Le MNT, chargeable à partir de fichier au format .asc, peut s'afficher à l'aide d'une orthophoto que l'on plaque ou d'un dégradé de couleur. La classe `fr.ign.cogit.geoxygene.sig3d.util.ColorShade` contient quelques propositions de dégradés accessibles depuis l'interface graphique.



**Figure 4 : Représentation pour un MNT**

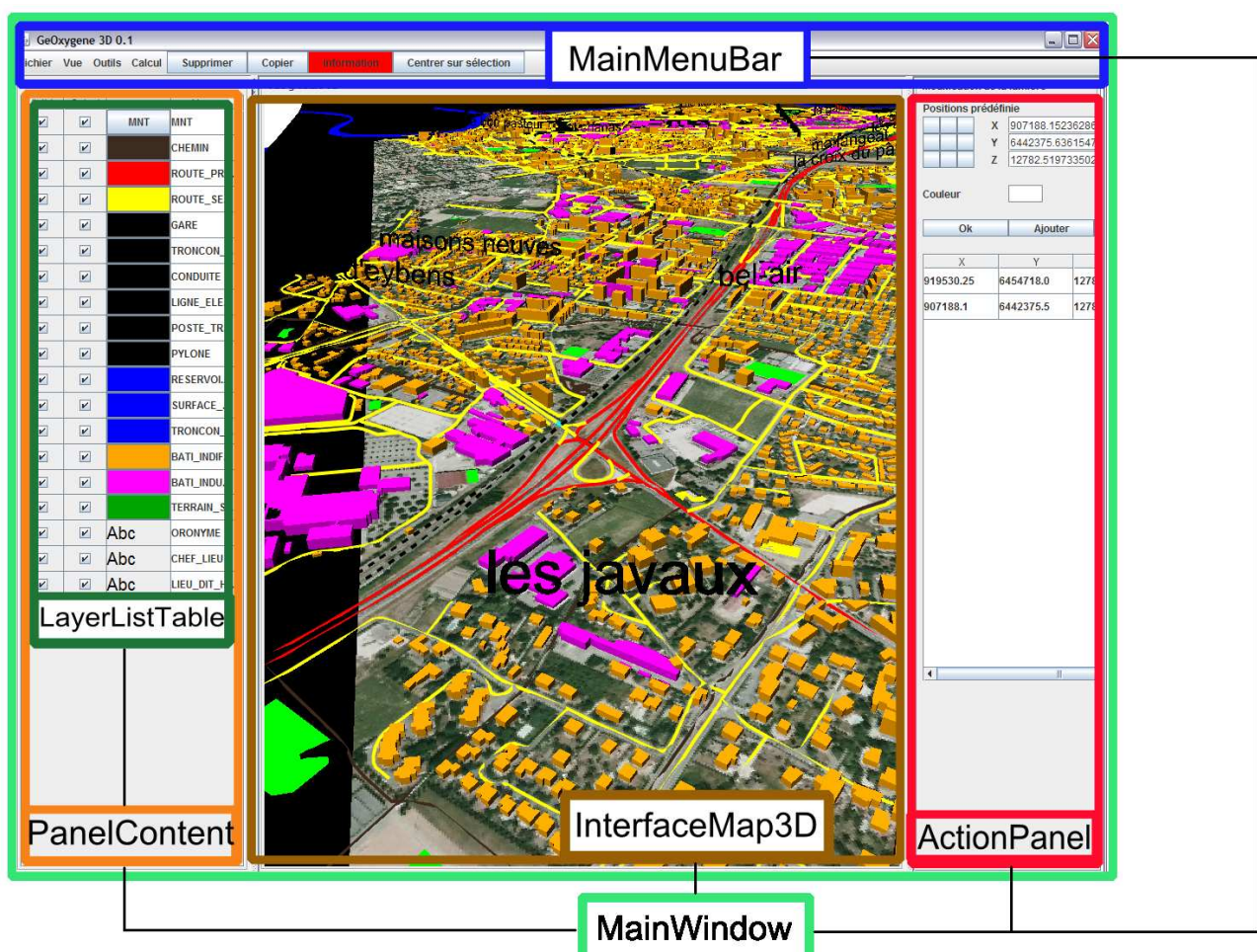
### 2.3.3 Modélisation d'une carte

Afin d'être affiché, l'ensemble des couches est regroupé dans une carte (classe « Map3D » package fr.ign.cogit.geoxygene.sig3d.semantic). Les objets de la classe « Map3D » permettent de gérer les couches. Les méthodes *addLayer* et *removeLayer* permettent respectivement d'ajouter et de supprimer une couche de la carte. La carte est associée à une branche de scène qui contient les branches de scènes de toutes les couches associées à la carte afin de synchroniser leurs affichages. La méthode *refresh()* permet de déclencher les méthodes *refresh()* de chacune des couches.

## 2.4 L'interface graphique

Les classes de l'interface graphique du viewer se trouvent dans le package fr.ign.cogit.geoxygene.sig3d.gui.

Il est possible de modifier l'interface graphique afin par exemple d'ajouter de nouvelles fonctionnalités ou de gérer de nouvelles implémentations de représentations à partir de celle-ci.



**Figure 5 : Interface du viewer, des classes correspondantes**

L'interface est définie par 4 classes principales, chacune de ces classes faisant référence de manière bijective à la classe *MainWindow* :

- **MainWindow** : il s'agit du point d'entrée de l'application. Cette classe permet de lancer l'interface de l'application 3D et fait référence aux différentes parties de l'interface,
- **MainMenuBar**: cette barre contient les différents menus et boutons. Il est possible d'y ajouter des boutons ou menus comme c'est le cas avec les containers Swing,
- **InterfaceMap3D** : il s'agit de la partie de l'interface gérant l'affichage 3D. Le contenu d'un objet de type Map3D y est représenté (voir chapitre Modèle de représentation),
- **PanelContent** : il s'agit de la partie gérant l'affichage des couches. La gestion des couches se fait à partir de la classe LayerListTable qui étend JTable. La classe définissant la table se trouve dans le package « fr.ign.cogit.geoxygene.sig3d.gui.table.layerstable».
- **ActionPanel** : il s'agit d'un container qui contient un panneau permettant de paramétrer les différentes fonctionnalités. La méthode *setPanel()* permet de remplacer le panneau actif par un autre panneau.

L'exemple codé dans la classe « AddMenu » illustre une manière de rajouter des outils (Cf. §4.1 : Classe AddMenu).

### 3 Les packages

Cette section vise à recenser les principaux packages et les classes les plus importantes du module afin d'en indiquer leur utilité. Les packages sont décrits par ordre alphabétique.

#### 3.1 Package fr.ign.cogit.geoxygene.sig3d

Il s'agit du package racine. Il contient une classe « Launcher » qui permet d'ouvrir le viewer et d'exécuter les exemples présentés.

#### 3.2 Package fr.ign.cogit.geoxygene.sig3d.calculation

Ce package contient les classes permettant d'effectuer des opérations géométriques sur les géométries 3D : calculs géométriques de bases en 3D (aire, volume, enveloppe convexe, centre de gravité, extrémités, translations, ...), d'indicateurs (convexité ...), de proximité, d'opération booléennes 3D entre deux objets (intersection, union, différence), de buffer 3D...

### 3.3 Package `fr.ign.cogit.geoxygene.sig3d.conversion`

Ce package permet d'effectuer des conversions entre les géométries GeOxygene et les géométries Java3D ou des transformations de géométrie (extrusion).

### 3.4 Package `fr.ign.cogit.geoxygene.sig3d.equation`

Ce package contient des classes permettant de définir des plans par équation.

### 3.5 Package `fr.ign.cogit.geoxygene.sig3d.geometry`

Ce package contient des géométries définies pour le besoin de la 3D.

### 3.6 Package `fr.ign.cogit.geoxygene.sig3d.geometry.topology`

Ce package contient un modèle topologique pour décrire sous forme de triangles des solides. La classe `fr.ign.cogit.geoxygene.sig3d.tetraedrisation.TetraedrisationTopo` permet d'effectuer la décomposition d'un solide dans ce modèle.

### 3.7 Package `fr.ign.cogit.geoxygene.sig3d.gui`

Ce package contient l'ensemble des éléments composant l'interface graphique.

### 3.8 Package `fr.ign.cogit.geoxygene.sig3d.gui.actionpanelmenu`

Ce package contient les différents éléments pouvant apparaître dans le panneau droit (`PanelContent`) de l'application.

### 3.9 Package `fr.ign.cogit.geoxygene.sig3d.gui.filter`

Ce package contient l'ensemble des filtres utilisés dans les recherches de fichier.

### 3.10 Package `fr.ign.cogit.geoxygene.sig3d.gui.menu`

Ce package contient les différents menu déroulants de l'application.

### 3.11 Package `fr.ign.cogit.geoxygene.sig3d.ihm.navigation3D`

Ce package contient les éléments relatifs à la navigation 3D.

### 3.12 Package `fr.ign.cogit.geoxygene.sig3d.gui.render`

Ce package contient les différents `render`, notamment pour le menu déroulant de dégradés de couleurs.

### 3.13 Package `fr.ign.cogit.geoxygene.sig3d.gui.table`

Ce package contient les différentes implémentations des `JTable` à savoir :

- `FeaturesListTable` : table listant les entités d'une couche vectorielle,
- `LayersListTable` : table listant les couches d'une carte,
- `LightsListTable` : table listant les lumières d'un univers,
- `GenericTexturesList` : liste contenant les textures chargées en mémoire par la classe `TextureManager`.

### 3.14 Package `fr.ign.cogit.geoxygene.sig3d.gui.window`

Ce package contient l'ensemble des fenêtres de menus apparaissant dans l'application.

### 3.15 Package `fr.ign.cogit.geoxygene.sig3d.io`

Ce package contient les différentes classes de chargement de données.

### 3.16 Package `fr.ign.cogit.geoxygene.sig3d.representation`

Ce package contient les classes mères et les packages relatifs aux représentations (Voir chapitre 2.3).

La classe « `ConstantRepresentation` » permet de gérer les paramètres globaux d'affichage comme le facteur d'échelle à appliquer à la carte, la couleur du fond de l'univers ou la couleur de sélection.

### 3.17 Package `fr.ign.cogit.geoxygene.sig3d.representation.basic`

Ce package contient les représentations de bases (Classes `ObjetXD` Cf. § 2.3) pour représenter de couleur unie les objets ponctuels, linéaires, surfaciques ou volumiques.



### 3.18 Package

`fr.ign.cogit.geoxygene.sig3d.representation.control`

Ce package contient les classes d'objets apparaissant dans l'univers 3D et aidant à la navigation (Par exemple, classe gérant le niveau constant)

### 3.19 Package

`fr.ign.cogit.geoxygene.sig3d.representation.modellingfile`

Ce package permet de gérer les classes relatives à la représentation avec un fichier de modélisation (Cf. § 2.3).

### 3.20 Package

`fr.ign.cogit.geoxygene.sig3d.representation.symbol`

Ce package permet de gérer les classes relatives à la représentation d'objets ponctuels sous forme de symboles (Cf. § 2.3). Un symbole est un rectangle ou un cube pour lequel une image est plaquée sur chacune de ses faces.

### 3.21 Package

`fr.ign.cogit.geoxygene.sig3d.representation.texture`

Ce package permet de gérer les classes relatives à la représentation de manière texturée d'objets (Cf. § 2.3).

À noter, la classe TextureManager permet de charger des objets de texture Java de manière à éviter de charger plusieurs fois en mémoire la même texture.

### 3.22 Package

`fr.ign.cogit.geoxygene.sig3d.representation.toponym`

Ce package permet de gérer les classes relatives à la représentation d'objets ponctuels sous forme de texte (Cf. § 2.3).

### 3.23 Package `fr.ign.cogit.geoxygene.sig3d.sample`

Ce package contient les exemples de code (Cf. § 4 : Exemple de code).

### 3.24 Package `fr.ign.cogit.geoxygene.sig3d.semantic`.

Ce package contient les différents éléments relatifs à la gestion de couche (Classes Carte3D, Couches etc.).

### 3.25 Package fr.ign.cogit.geoxygene.sig3d.simplification

Ce package est relatif aux algorithmes de simplifications 3D.

### 3.26 Package fr.ign.cogit.geoxygene.sig3d.tetraedristion

Ce package est relatif aux décompositions d'objets.

### 3.27 Package fr.ign.cogit.geoxygene.sig3d.util

Ce package contient des classes utilitaires. Notamment la classe ColorShade qui stocke les dégradés accessibles depuis l'interface.

## 4 Exemple de code

Le package fr.ign.cogit.geoxygene.sig3d.sample contient des classes mettant en œuvre les opérations les plus courantes du module 3D. Dans cette section nous décrirons ce que fait chacune de ces classes. Vous êtes invités à consulter le code très documenté pour comprendre le fonctionnement. Il peut être nécessaire de changer le chemin de certaines ressources qui ne seraient pas présentes lors de l'exécution du code.

### 4.1 Classe AddMenu

Cette classe montre comment modifier l'interface graphique de l'application afin d'ajouter un bouton permettant d'afficher un nouveau menu dans le panneau droit.

Ce menu permet d'appliquer automatiquement des textures de tuile sur le toit et de briques sur les faces des bâtiments.

### 4.2 Classe DisplayData

Cette classe propose deux méthodes :

- *createCube ()* : qui montre comment implémenter une géométrie de type GM\_Solid, ici un cube,
- *main()* : cette méthode fait appel à la méthode fabriqueCube afin de montrer comment afficher cette géométrie dans l'univers 3D en lui appliquant un style.

### 4.3 Classe DTMDisplay



Cette classe montre comment afficher un MNT dans l'univers 3D soit en appliquant un dégradé de couleurs soit en plaquant une orthophoto.

Pour exécuter ce code, il est nécessaire de télécharger un MNT au format .asc. Il est possible d'en obtenir un sur le site internet de l'IGN. Pour y accéder : <http://professionnels.ign.fr/>, rubrique Echantillons gratuits> Produits gratuits.

Pour télécharger les MNT au format .asc, il faut sélectionner le menu « BD Alti ® », le format « ARC VIEW ® » et accepter les conditions d'utilisation.

Le code est exécutable en plaçant l'un des fichiers dans le dossier du module : « Ressource/demo3D/mnt » et en le renommant « ISERE\_25\_asc.asc ».

Attention, il peut être nécessaire de recompiler avant d'exécuter le code.

#### 4.4 Classe RGE

Cette classe montre comment afficher des données vectorielles de la BD Topo ® en les extrudant et/ou les plaquant sur un MNT. Une représentation prédéfinie sera appliquée aux géométries et un filtre est effectué, toutes les couches de la BD Topo ® ne sont donc pas chargées. Pour retrouver la liste des couches chargées et des couleurs appliquées, vous pouvez consulter la classe MNTRGE.

Il est également possible de proposer un suréchantillonnage des données (en changeant le booléen surechantillonnage). Attention, cela peut augmenter considérablement le temps de calcul puisque cette opération consiste à redécouper les données en fonction des mailles du MNT. (Il peut être conseillé de ne pas afficher les couches les plus volumineuses comme « Zone\_végétation » ou « Commune »).

Pour exécuter ce code, il est nécessaire de télécharger un MNT au format .asc et la BD Topo ® sur la même zone. On propose de le faire à partir du site <http://professionnels.ign.fr/>, rubrique Echantillons gratuits> Produits gratuits.

Il faut sélectionner le menu « BD Topo® », le format « ARCVIEW ® » et accepter les conditions d'utilisation pour lancer le téléchargement.

Dans le fichier .zip récupéré, 2 dossiers devraient être présents :

- MNT : il faudra placer son contenu dans le répertoire du module « Ressource/demo3D/mnt » et en le renommant « ISERE\_25\_asc.asc » si nécessaire,
- BDTOPO\_X356\_SHP\_L93 : il faudra placer les dossiers qu'il contient dans le répertoire du module « bdtopo\_lam93 ». L'application scannerait ses dossiers pour charger les shapefiles qu'il contient.

Attention, il peut être nécessaire de recompiler avant d'exécuter le code.

Utilisation du module 3D de GeOxygene

## 4.5 Classe Symbology

Cette classe montre comment appliquer un symbole à un objet 3D. Une centaine de points générés aléatoirement sont représentés grâce à un fichier .obj.

## 4.6 Classe Toponym

Cette classe permet de montrer comment paramétrer des représentations textuelles pour des objets ponctuels à partir d'une couche de données représentant des noms de ville.

# 5 Bibliographie

Brasebin M, 2009 GeOxygene: An Open 3D Framework for the Development of Geographic Applications. In: 12th International Conference on Geographic Information Science (AGILE'09), 2-5 June, Hanovre (Germany)

GeOxygene, 2005, Guide utilisateur, [http://oxygene-project.sourceforge.net/geoxygene/docs/geoxygene-guide\\_utilisateur-1.0.pdf](http://oxygene-project.sourceforge.net/geoxygene/docs/geoxygene-guide_utilisateur-1.0.pdf)

ISO, 2003, Geographic information — Spatial Schema ISO Draft International Standard 19107:2003.

Kada M., 2007, 3D Building Generalisation by Roof Simplification and Typification. in proceedings of ICC2007, the International Cartography Association Conference, Moscow (Russia), 2007.

Oasada, R., Funkhouser T., Chazelle B., and Dobkin, D. 2002. Shape distributions. ACM Trans. Graph. 21, 4, 807–832.