
OPENSOURCE3D

VERSION 0.1 ALPHA

Cette version de la documentation n'est plus maintenue. Vous pouvez désormais accéder à la version HTML en cours de refonte à cette adresse :

<http://www.openspace3d.com/documentation/fr>



Guide de prise en mains

I-MAGINER

8, rue Monteil

44000 NANTES

France

Tél. : + 33 (0)2 40 12 16 44

E-Mail : contact@openspace3d.com

Website: <http://www.openspace3d.com>



Sommaire

Avant-propos	1
Présentation de la technologie SCOL	2
L'origine de SCOL.....	2
Derrière la simplicité, la puissance.....	2
Langage Multiutilisateurs.....	2
Nombreuses possibilités d'enrichissement.....	2
Le SO3Engine un moteur 3D basé sur Ogre	3
Caractéristiques Techniques.....	3
Présentation de la plateforme OpenSpace3D	4
La plateforme de développement d'application 3D.....	4
Développez de manière rapide et intuitive.....	4
Un concept simple, un outil accessible à tous.....	4
Démarrer avec OpenSpace3D	5
Pré-requis.....	5
Organisation des fichiers dans la partition SCOL.....	5
Installation du scol Voy@ger.....	6
Installation de OpenSpace3D Editor.....	10
Lancement.....	13
Configurateur 3D.....	14
Ergonomie Générale de la Plateforme	17
Menu Général	18
Arbre de scène et gestions des Ressources.....	18
Ouvrir une scène OpenSpace3D (.xos).....	20
Importer une scène Ogre (.scene).....	22
Enregistrer une scène OpenSpace3D (.xos).....	25
Export d'applications développées sous OpenSpace3D	25
As a Scol File.....	25
As A Stand Alone Application.....	26
As a Web Page.....	29
Gestion de l'arbre de Scène et d'informations sur les Ressources	31
Scene Tree.....	31
Ajout de mesh : Wizard.....	54
Color Map.....	57
Edition 3D	58
Interface.....	58
Déplacement dans la 3D.....	60
Sélection d'un objet.....	61
Déplacement d'un objet.....	62



La duplication d'Objets	63
Le mode Play/Pause	65
Edition des Fonctions (PlugIT)	66
Interface.....	66
Les liens.....	67
Documentation des plugIT	72
Les plugIT « navigation ».....	77
PlugIT Walkthrough.....	77
PlugIT Examine View.....	78
Les plugIT « Network »	79
PlugIT chat.....	79
PlugIT connect	80
PlugIT message	81
PlugIT room item	82
Les plugIT « Input »	83
PlugIT Keyboard.....	83
PlugIT Joypad	85
PlugIT Wiimote.....	86
Wii Tools	86
PlugIT Mouse.....	86
PlugIT NeuroSky	89
PlugIT Serial	90
Les plugIT « Interface »	91
PlugIT flash Interface.....	91
Les plugIT « Material »	93
PlugIT Material color.....	93
PlugIT change Material.....	94
PlugIT change texture	95
PlugIT texture Text	96
Les plugIT « Maths »	97
PlugIT Get vector.....	97
PlugIT Set Vector.....	98
PlugIT vector average	99
PlugIT vector multiply	100
PlugIT vector permutation	101
PlugIT operator	102
Les plugIT « Media »	103
PlugIT Sound	103
PlugIT Youtube.....	104
PlugIT Speech	106
PlugIT Speech Recognition.....	109

Les plugIT « Misc»	112
PlugIT Open Url.....	112
PlugIT On init	113
PlugIT Close	114
PlugIT Timer.....	115
PlugIT Séquence	116
PlugIT Var.....	117
PlugIT Switch	118
PlugIT ActiveXmessage.....	119
PlugIT Switchcase.....	122
PlugIT If.....	124
PlugIT Dialog box.....	125
PlugIT Counter.....	126
PlugIT Call Url.....	127
plugIT input Dispatcher	128
PlugIT output Dispatcher	129
PlugIT Random Output :	130
Les plugIT « Object»	131
PlugIT getObject.....	131
PlugIT Set Active camera	132
PlugIT Animation	133
PlugIT : Anmation Transition	135
PlugIT Distance.....	136
PlugIT Rotate	137
PlugIT Light.....	138
PlugIT Hide.....	139
PlugIT Object link.....	140
PlugIT Get Object	141
PlugIT Get Camera	142
PlugIT objectFollow.....	143
PlugIT objectGoto.....	144
PlugIT Object Position.....	145
PlugIT target	146
Les plugIT « Physics» (cf Utilisation avancées : Moteur Physique)	147
plugIT PhysicContact.....	147
plugIT PhysicImpulse	148
plugIT physicTools	149
Les plugIT « Rendering».....	150
plugIT Stereo.....	150
PlugIT fullscreen	151
PlugIT Viewport.....	152
PlugIT Compositor	153
Debug et Log	155
Utilisation avancée et Définitions.....	157
Ressources Graphiques (3D)	157
Group Resources	161
Group Meshes.....	162

Resources Directories	163
Importer une ressources Ogre (shaders, texture, material, mesh)	164
Exporter un groupe	168
Définitions et utilisation avancée des paramètres	169
Les SkyBoxes et les environnements de scène	179
Le moteur Physique	180
Contact et Support.....	184



Avant-propos

Le projet OpenSpace3D a pris naissance dans les locaux de la société I-maginer. En effet, celle-ci souhaite démocratiser la création de contenu 3D temps réel à un large public.

Ce projet est Open Source, ce qui favorise sa diffusion et son utilisation au sein de studios de production 3d (infographiste/intégrateur) mais également auprès des écoles de création multimédia.

Cependant, OpenSpace3D peut également être utilisé au sein de grands compte pour répondre à des projets clients, dans les studios de jeux vidéo ainsi que par les amateurs constituant sa communauté.

L'équipe de développement est heureuse de vous présenter cette nouvelle version bénéficiant de fonctionnalités nouvelles.

Le présent document constitue une manière de prendre en main de manière autonome le logiciel.

Ainsi, il est conçu pour permettre aux utilisateurs néophytes d'apprendre d'une manière simple et progressive à utiliser la plateforme OpenSpace3D.

Si vous êtes une entreprise : pour toute question relative à un besoin de formation sur la plateforme ou un besoin de support contactez : contact@openspace3d.com

Le forum Scroling : <http://www.scolring.org/> soutenu par une communauté (dont les développeurs de la plateforme OpenSpace3D) est également présent afin de vous renseigner.

Bonne lecture et Bonne créa !

L'équipe OpenSpace3D



Présentation de la technologie SCOL

L'origine de SCOL

SCOL est un langage de programmation directement pensé pour l'utilisation en réseau, qui ouvre aux sites internet l'interactivité entre les utilisateurs, les fonctions multimédia et la navigation en 3 dimensions.

Derrière la simplicité, la puissance

C'est un langage « interprété », ce qui laisse au concepteur d'un site SCOL la possibilité de le faire évoluer selon ses désirs et ses besoins. Cependant, l'apprentissage du langage n'est nécessaire qu'aux *SCOLMaster* désirant développer des applications très spécifiques. En effet, s'il est vrai que **SCOL** est très flexible, il est aussi vrai qu'il reste relativement complexe.

Langage Multiutilisateurs

Dans le modèle traditionnel de l'Internet, les utilisateurs ne sont pas en contact direct. Le serveur se scinde en autant de parties qu'il y a de clients en ligne.

Scol permet au contraire à plusieurs connectés de communiquer et d'interagir ; Le serveur Scol ne se scinde pas, il relaie les communications. La navigation n'est plus individuelle mais collective.

Cette caractéristique est idéale dans le domaine de la 3D, du jeu en ligne, de la communauté virtuelle, du commerce électronique...

Nombreuses possibilités d'enrichissement

Scol inclut des bibliothèques de développement très complètes (réseau, moteur 3d, interface 2d, audio, vidéo, SQL...).



Le SO3Engine un moteur 3D basé sur Ogre

Depuis Novembre 2008, I-maginer travaille activement sur un nouveau Moteur 3D : le SO3Engine.

C'est ce moteur qui sera abordé dans ce document.

Ainsi, plutôt que de réinventer la roue, nous avons souhaité intégrer un moteur 3D très puissant « Ogre 3D » dans la technologie SCOL.

<http://www.ogre3d.org/>

En effet, celui-ci permet le développement d'application 3D avec un rendu photo réaliste et très performant.

Les fonctionnalités du SO3Engine dérivent ainsi directement des fonctionnalités du moteur Ogre.

Caractéristiques Techniques

- Plateforme & 3D API support : Compatibilité DirectX/OpenGL, support Hardware et Information Système
- Management des scènes : Management des Objets 3D (mesh, lights, cameras, SkyBoxes...) pour avoir le contrôle, du graphe de scène, Eclairages Dynamiques, Ombres temps réels.
- Material/Shaders Supports : Effets sur les matériaux, Supports vertex and fragment shaders Texture compression
- Animations Support des animations par squelettes, animation des nœuds de scène, morphing.
- Moteur Physique par le biais de l'intégration de Newton Physics Engine
- Autres caractéristiques: Ressources Dynamiques, Moteur Physique basé sur Newton, Exports pour 3DS Max, Blender, Sketch up
- Effets de rendu (Stéréoscopie)
- Gestion des contenus Flash (2D et 3D)



Présentation de la plateforme OpenSpace3D

La plateforme de développement d'application 3D

Avec **OpenSpace3D**, plateforme directement développée avec la technologie SCOL, tous les utilisateurs peuvent enfin construire intégralement des scènes 3D interactives de grande richesse graphique sans entrer une seule ligne de code. Cette plateforme dispose de nombreuses fonctions prêtes à l'emploi, d'une grande souplesse et faciles d'utilisation.

La réalisation d'une application **OpenSpace3D** consiste donc à intégrer différentes fonctions et à définir leurs interactions réciproques.

Développez de manière rapide et intuitive

OpenSpace3D permet de créer intégralement son application en assemblant des fonctions sans programmation.

Chacune des fonctions disponibles gère une fonctionnalité de l'application, par exemple une vidéo, un lien web, une animation... A partir de cette bibliothèque, choisissez vos fonctions et combinez-les librement pour bâtir votre application en définissant les liens entre les fonctions.

Un concept simple, un outil accessible à tous

OpenSpace3D est une plateforme multi-usages comportant plusieurs niveaux en fonctions des compétences de vos équipes.



Infographiste 3D :

Intégrez facilement et rapidement vos propres productions 3D en les intégrant en temps réel dans notre plateforme et en leur appliquant des fonctions et des interactions.



Intégrateur :

Ce mode déjà très complet permet de construire intégralement une application en utilisant les éléments fournis de base dans notre bibliothèque.



Développeur :

Ce niveau s'adresse aux développeurs qui souhaitent aller plus loin en programmant de nouvelles fonctionnalités grâce au langage de programmation SCOL car la plateforme est ouverte.



Démarrer avec OpenSpace3D

Pré-requis

Organisation des fichiers dans la partition SCOL

Rappel : sauf mention contraire, le chemin des différents répertoires sera donné relativement au sous répertoire *Partition_LocalUsr* du répertoire mes documents.

Exemple : *C:\Documents and Settings\User\Mes Documents\scol voyager\Partition_LocalUsr*

Les fichiers que vous utiliserez pour développer une application doivent obligatoirement être placés dans le répertoire *Partition_LocalUsr* -> exemple : *C:\Documents and Settings\User\Mes Documents\scol voyager\Partition_LocalUsr\ma_scene\monobjet.mesh*

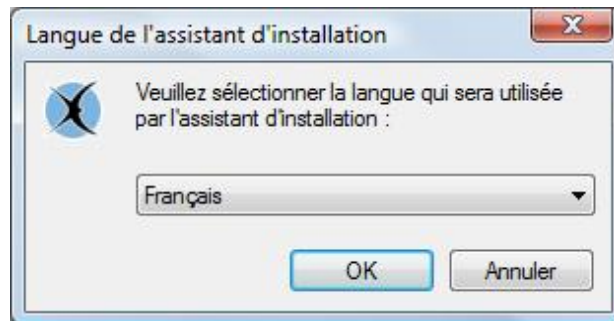
OpenSpace3D ne lira que les éléments **placés dans ce répertoire uniquement !**

Il importe donc que **l'infographiste produise ses ressources graphiques directement à partir de ce répertoire** sinon il lui sera impossible d'importer ses propres objets 3D.

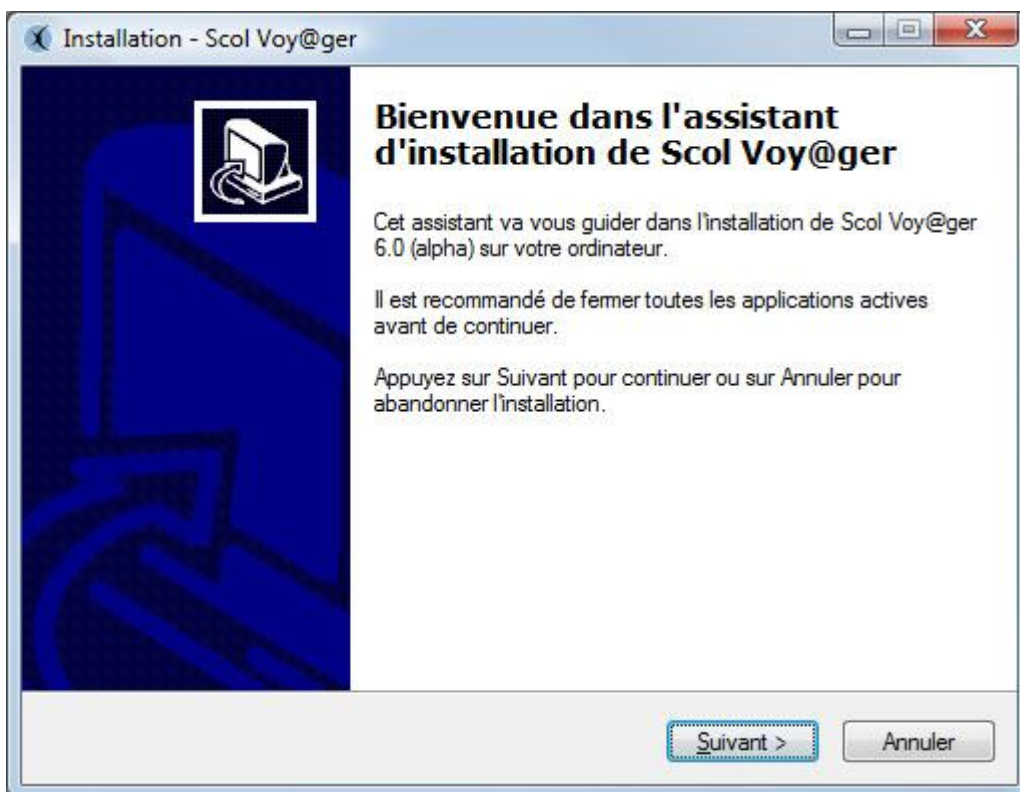
Idéalement, l'infographiste exporte ses ressources graphiques dans un dossier correspondant à sa scène (cf. : "Documentation export Ogre/SCOL").

Installation du scol Voy@ger

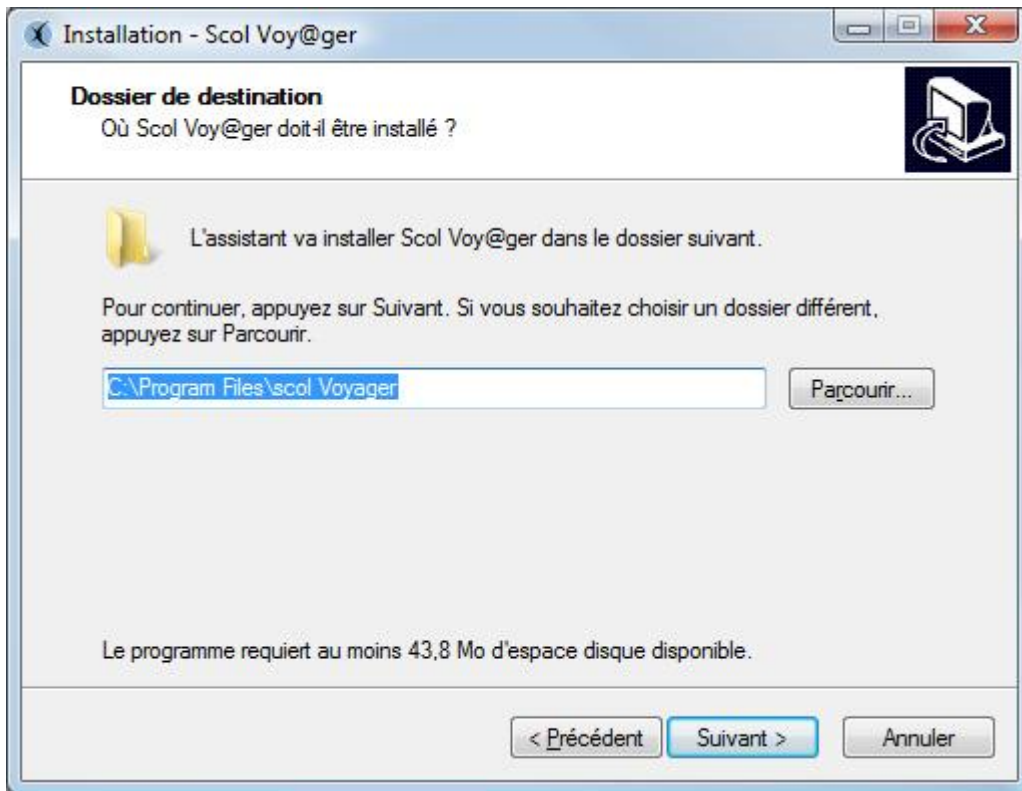
Exécutez le fichier "scol_plugin.exe".



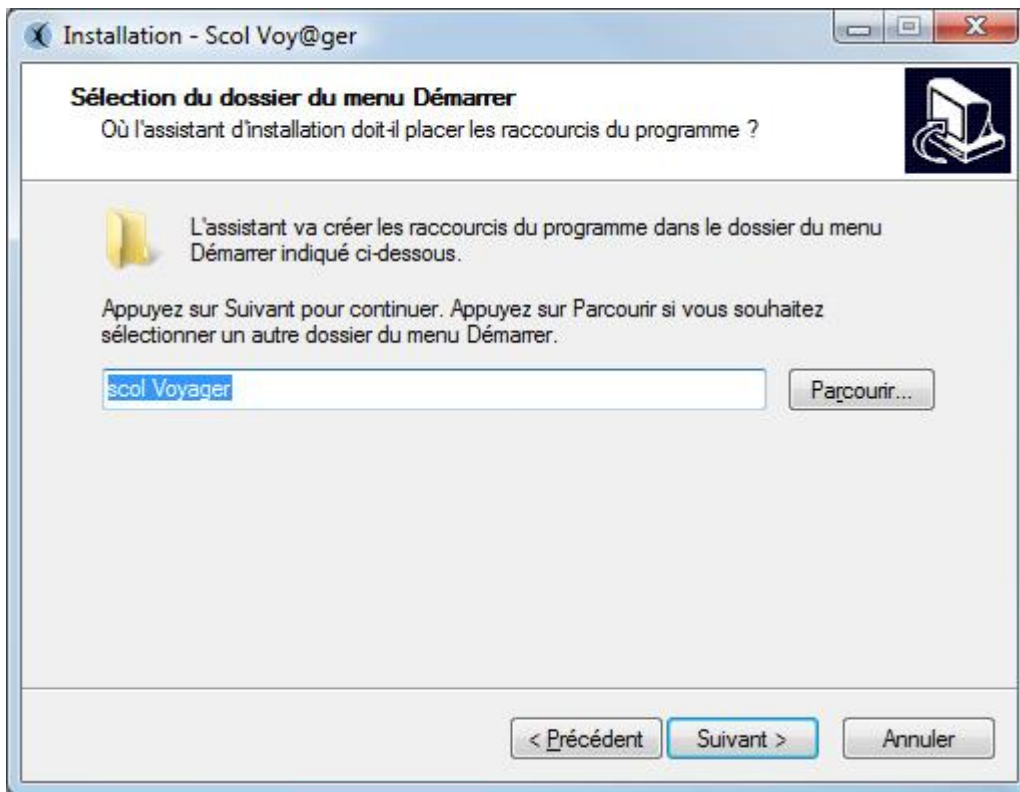
Sélectionnez votre langue puis cliquez sur "Ok".



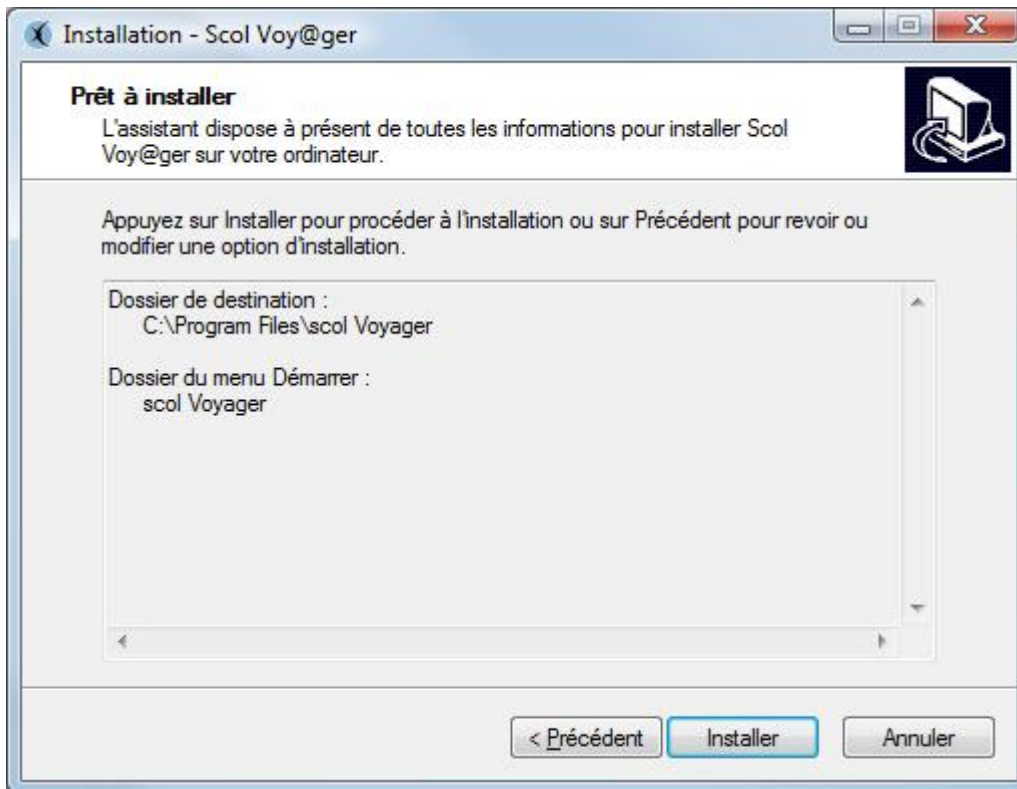
Sur l'écran de bienvenue cliquez "suivant".



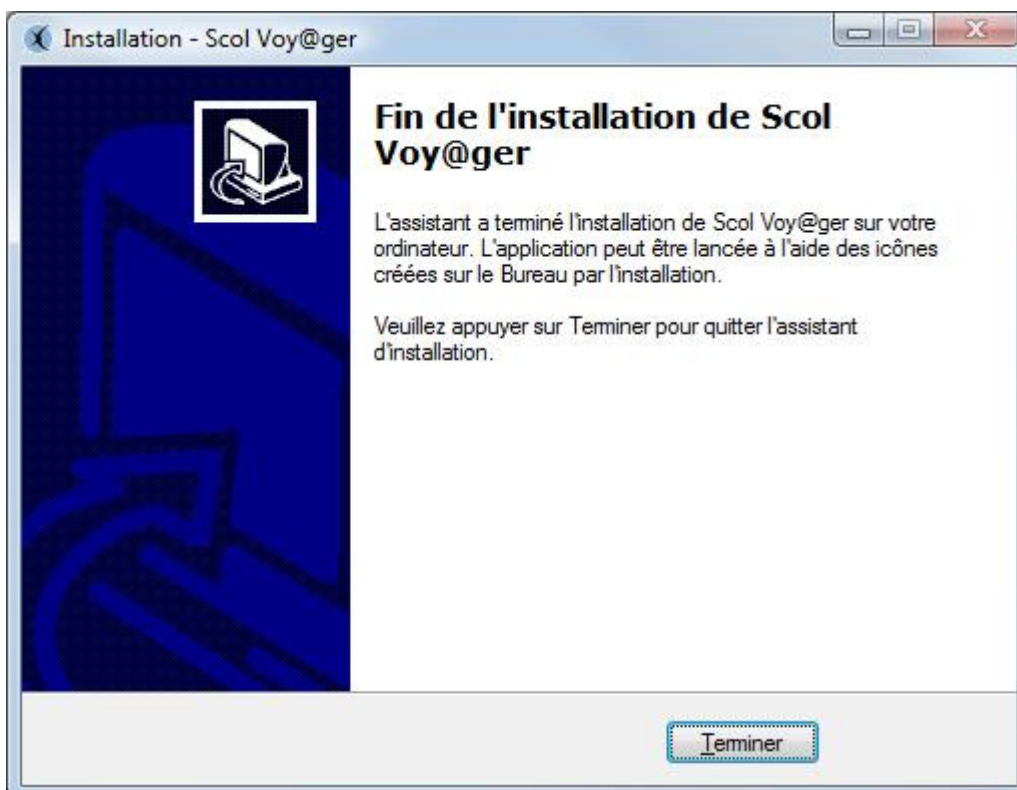
Sur l'écran "Dossier de destination", vous pouvez préciser le chemin d'installation, par défaut "C:\Program Files\scol Voyager", puis cliquez sur "Suivant".



Sur l'écran "Sélection du dossier du menu Démarrer", vous pouvez préciser le nom du groupe de programme qui contiendra les raccourcis de l'installation, par défaut "scol Voyager", puis cliquez sur "Suivant".



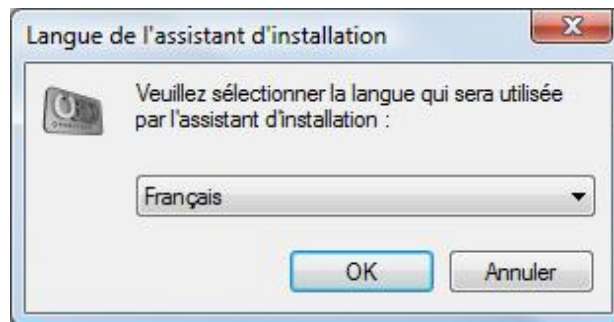
Sur l'écran "Prêt à installer", cliquez sur "Installer" pour installer l'application.



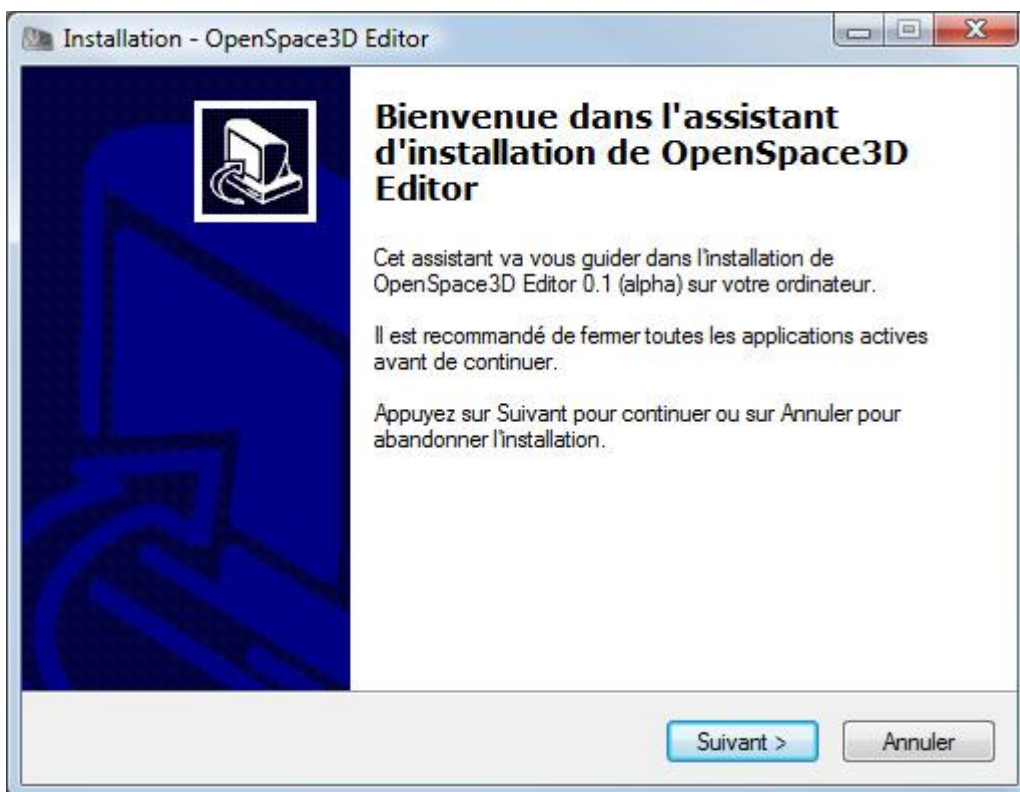
Le Scol Voy@ger est maintenant installé sur votre ordinateur, cliquez "Terminer" pour fermer.

Installation de OpenSpace3D Editor

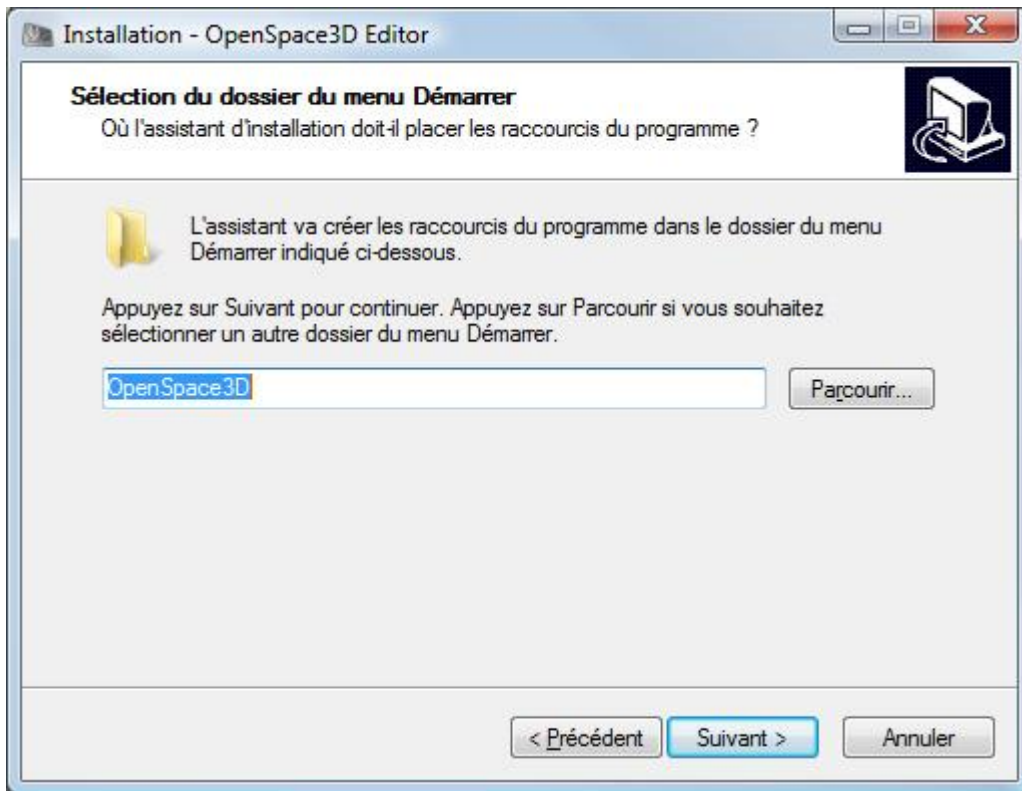
Exécutez le fichier "OpenSpace3D_editor_setup.exe".



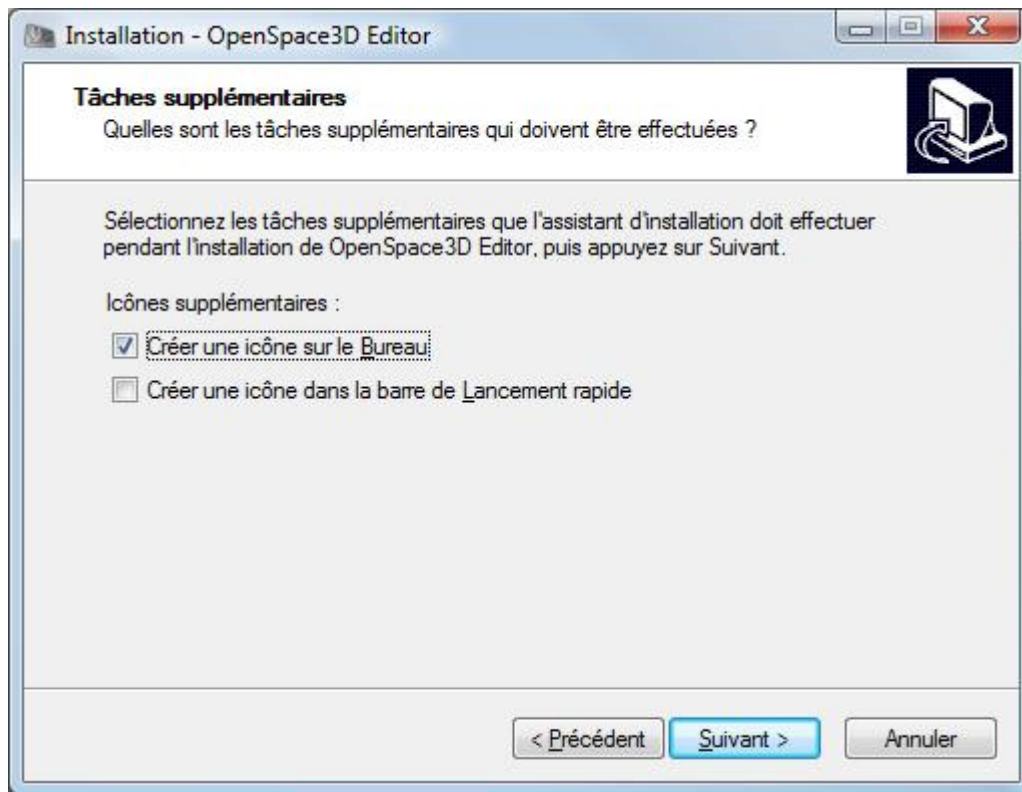
Sélectionnez votre langue puis cliquez sur "Ok".



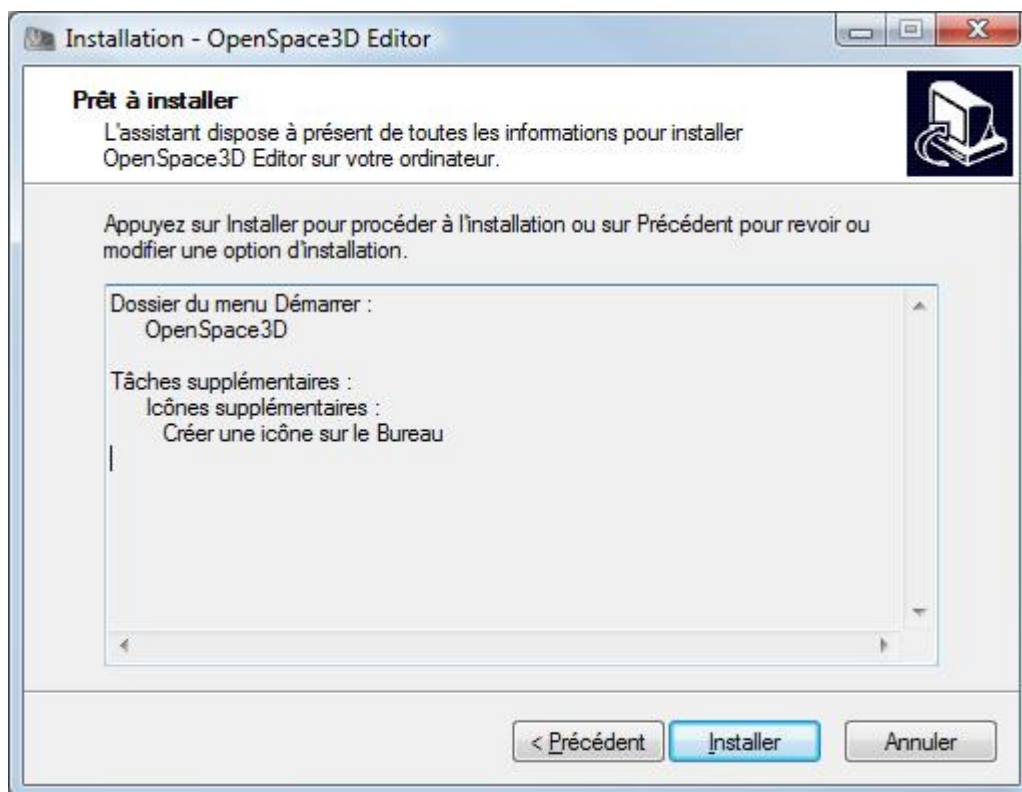
Sur l'écran de bienvenue cliquez "suivant".



Sur l'écran "Sélection du dossier du menu Démarrer", vous pouvez préciser le nom du groupe de programme qui contiendra les raccourcis de l'installation, par défaut "OpenSpace3D", puis cliquez sur "Suivant".

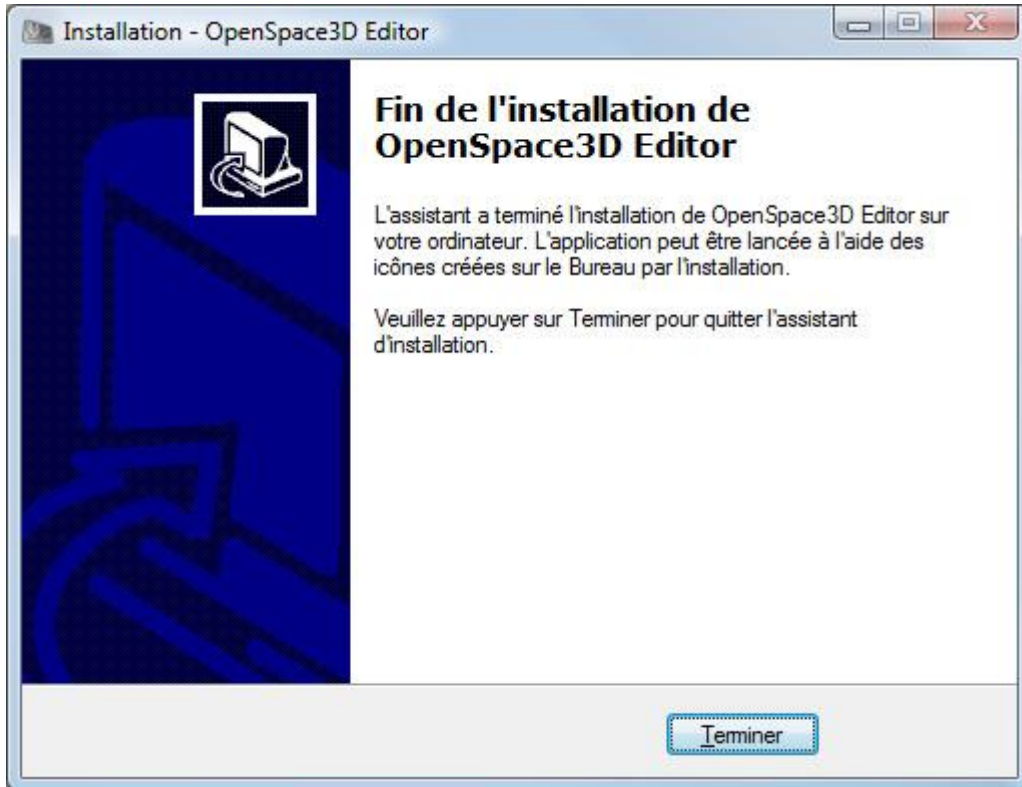


Sur l'écran "Tâches supplémentaires", vous pouvez activer ou désactiver la création des raccourcis sur le bureau Windows et la barre de lancement rapide, puis cliquez sur "Suivant".





Sur l'écran "Prêt à installer", cliquez sur "Installer" pour installer l'application.



OpenSpace3D Editor est maintenant installé sur votre ordinateur, cliquez "Terminer" pour fermer.

Lancement

Pour lancer OpenSpace3D Editor cliquez sur le raccourci :

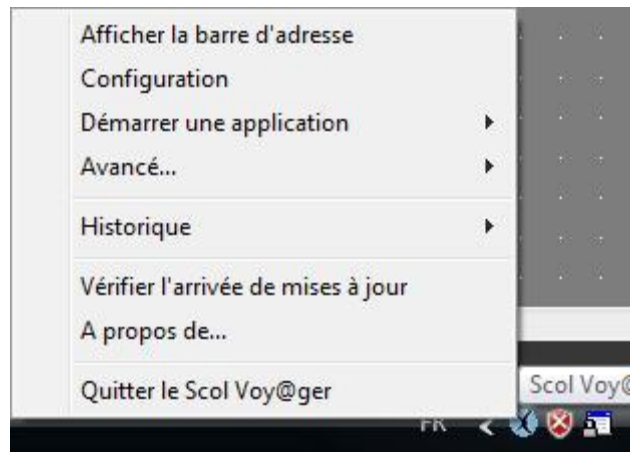


Configurateur 3D

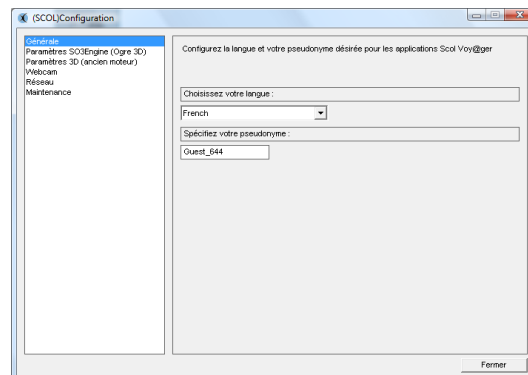
Une fois le Scol Voy@ger installé, Il est possible de configurer les paramètres du moteur 3d « SO3Engine » via la configuration du Scol Voy@ger



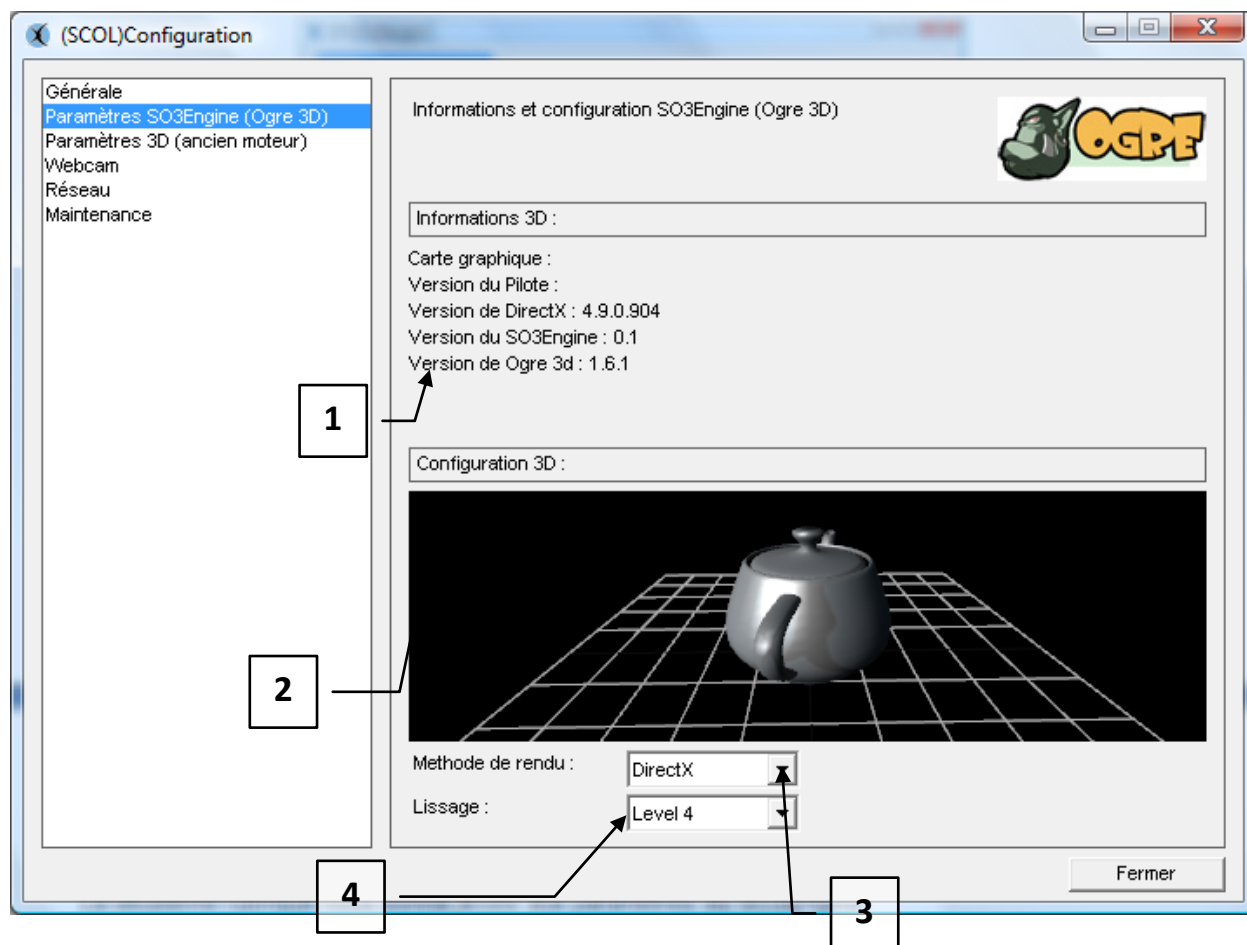
Par un clic droit sur l'icône du scol Voy@ger :



On peut avoir accès au menu configuration :



La deuxième rubrique nous donne accès aux paramètres du SO3Engine.



1/ C'est la rubrique d'information 3D sur le matériel de l'utilisateur ainsi que la version de DirectX installée et les versions du SO3Engine et de Ogre 3D utilisées.

2/ C'est la zone 3D de Test qui permet de valider le bon fonctionnement de la 3D pour la suite

3/ Méthode de rendu : 2 méthodes de rendu dites middleware peuvent être utilisées :

- DirectX
- OpenGL

N.B 1 : Si l'on modifie ces paramètres, le scol Voy@ger devra être relancé pour que les modifications prennent effets.



N.B 2 : DirectX/OpenGL

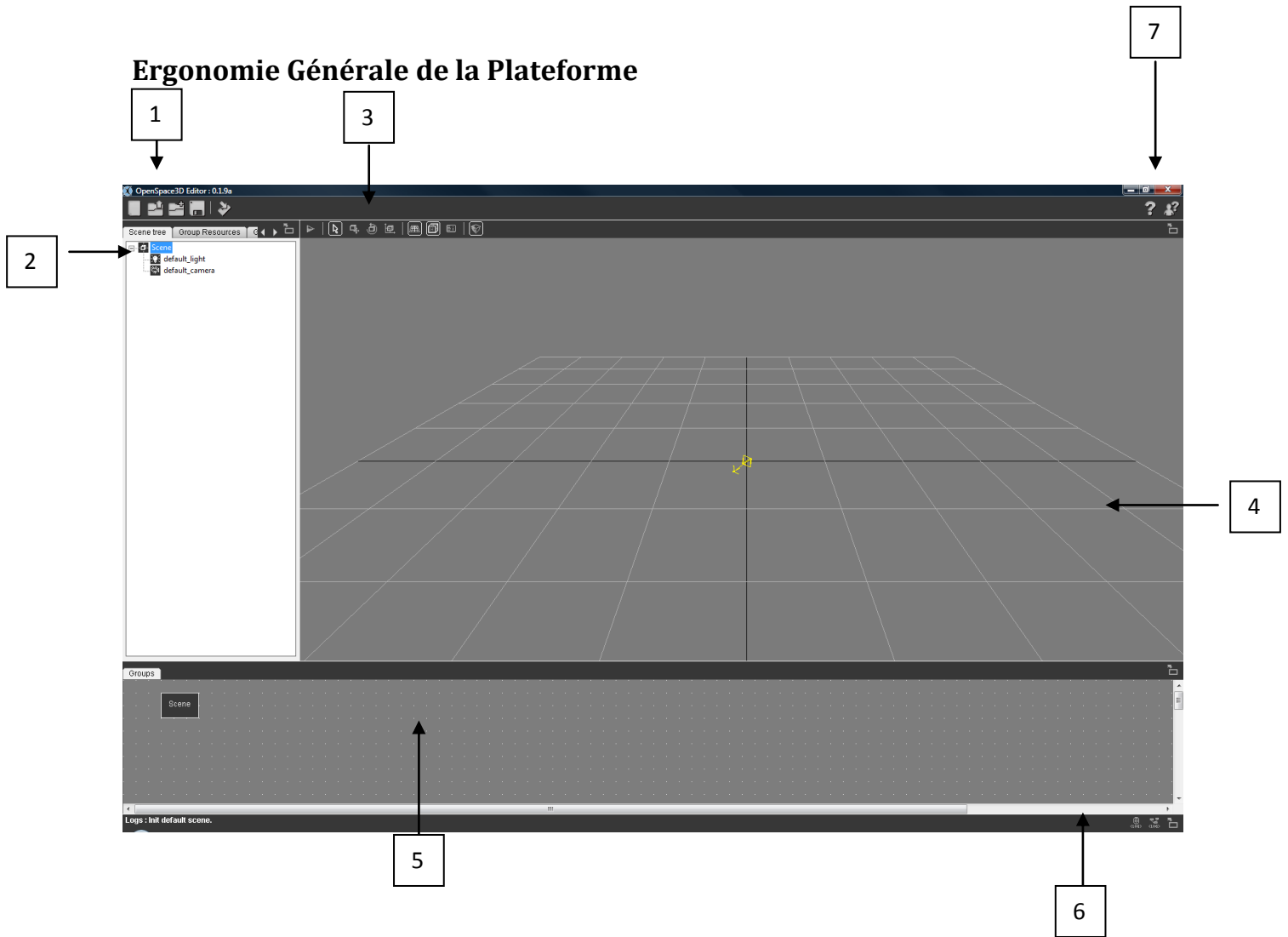
Sous Windows, il est fortement recommandé de travailler en mode DirectX, pour s'assurer de la bonne stabilité de la 3D.

4/ Lissage : Ce sont les paramètres d'Anti-Aliasing qui assurent une bonne qualité 3D afin d'éviter les effets d'escalier sur le rendu.

Il est à noter que plus le nombre de Level (DirectX) ou de pass (OpenGL) est important plus la qualité graphique sera améliorée.

Cependant si l'on constate une lourdeur dans l'affichage des scènes 3D, il peut être utile de diminuer le lissage afin d'optimiser les scènes 3D.







Ergonomie Générale de la Plateforme



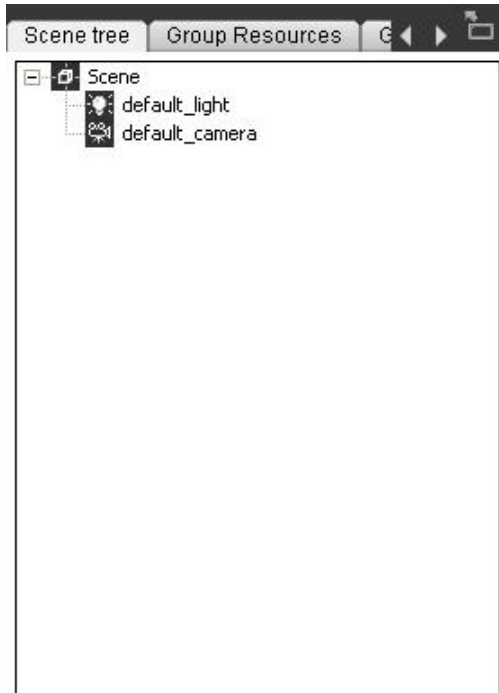
1	Menu Général
2	Arbre de scène et Gestion des Ressources
3	Menu d'Edition 3D
4	Vue 3D
5	Zone d'Edition des Fonctions
6	Zone de Logs
7	Aides et à Propos

Menu Général



	Ctrl+N	New : Nouvelle scène, vous permet de télécharger la scène courante dans OpenSpace3D et d'en créer une nouvelle.
	Ctrl+O	Open Scene : Ouvrir, vous permet d'aller parcourir vos fichiers pour charger un fichier de scène OpenSpace3D (.xos) dans OpenSpace3D pour l'éditer.
	Ctrl+I	Import Scene : Importer une scène Ogre Max, vous permet de parcourir vos fichiers pour charger un fichier de scène (.scene) dans la scène OpenSpace3D courante
	Ctrl+S	Save : Enregistrer, vous permet d'enregistrer la scène courante en .xos
	Ctrl+Shift+S	Save as OpenSpace3D scene : vous permet d'enregistrer la scène courante sous un autre nom au format .xos
	Ctrl+E	Export to OpenSpace3D Player : Créer un fichier de lancement de l'application ou une intégration dans un navigateur internet par l'upload sur un server html

Arbre de scène et gestions des Ressources

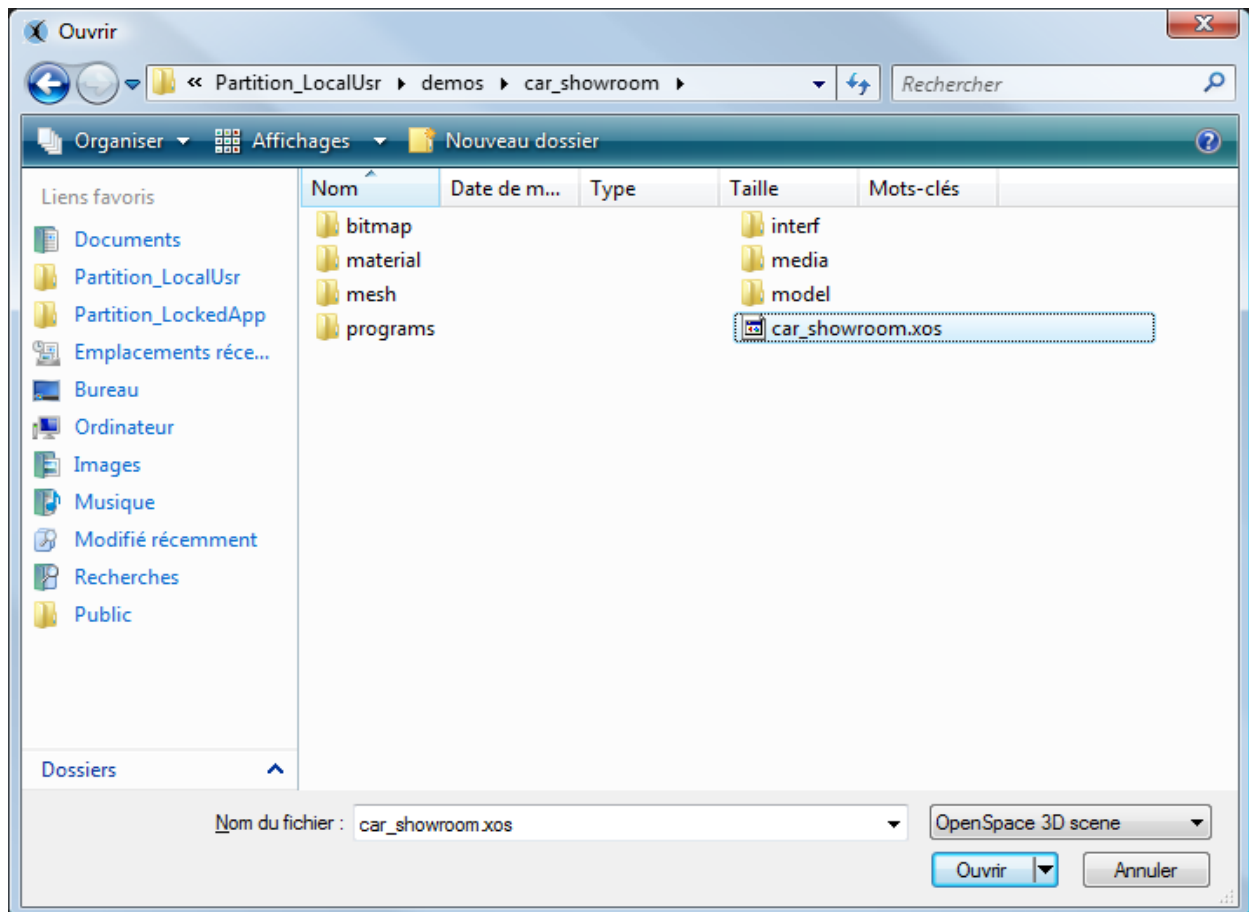


Ouvrir une scène OpenSpace3D (.xos)



Dans la zone Menu, cliquez sur l'icône « Open Scene »

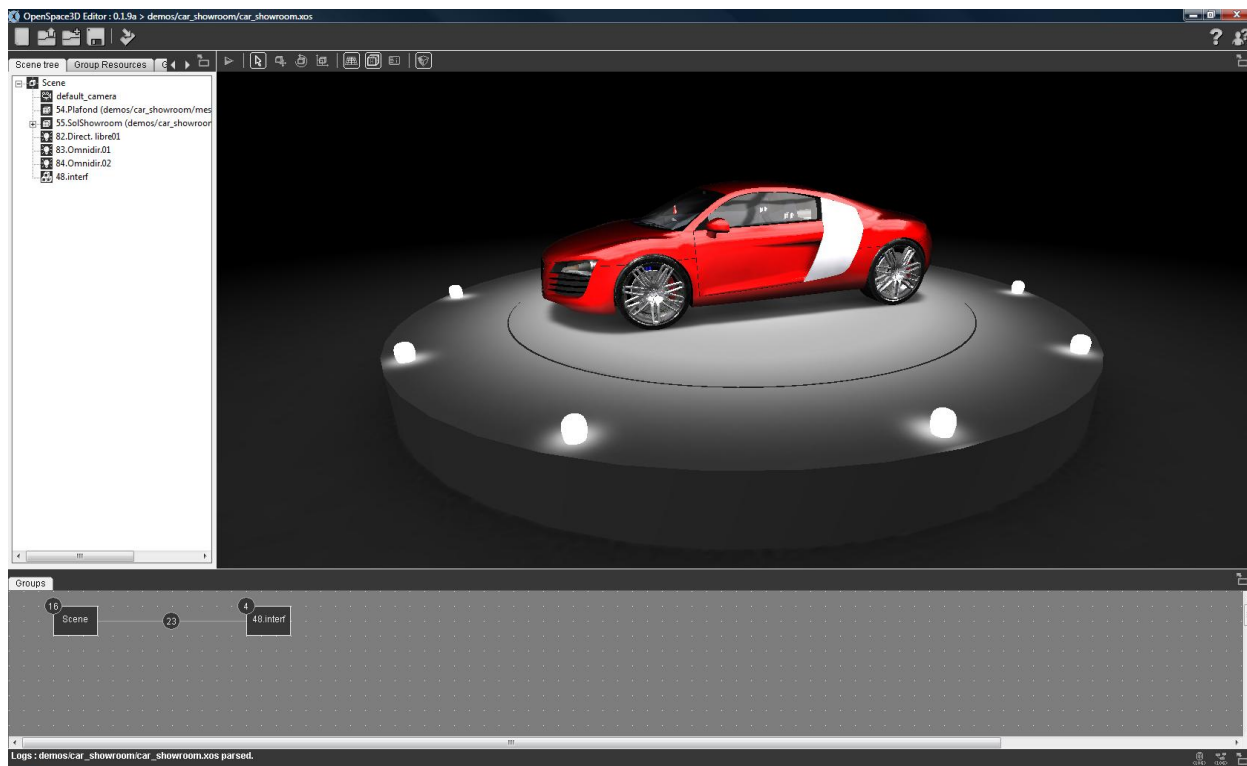
Parcourez votre dossier *Partition_LocalUsr* pour aller sélectionner votre fichier de scène à ouvrir :



Lorsque vous importez des ressources graphiques Ogre, il est possible que les chemins des ressources dans les fichiers *.material et *.program (shaders) ne correspondent pas aux chemin Scol.

Acceptez la proposition qui peut vous être proposée en cliquant sur « Ok » pour corriger les chemins dans les fichiers de ressources.

Patientez quelques instants pendant qu'OpenSpace3D charge votre scène :



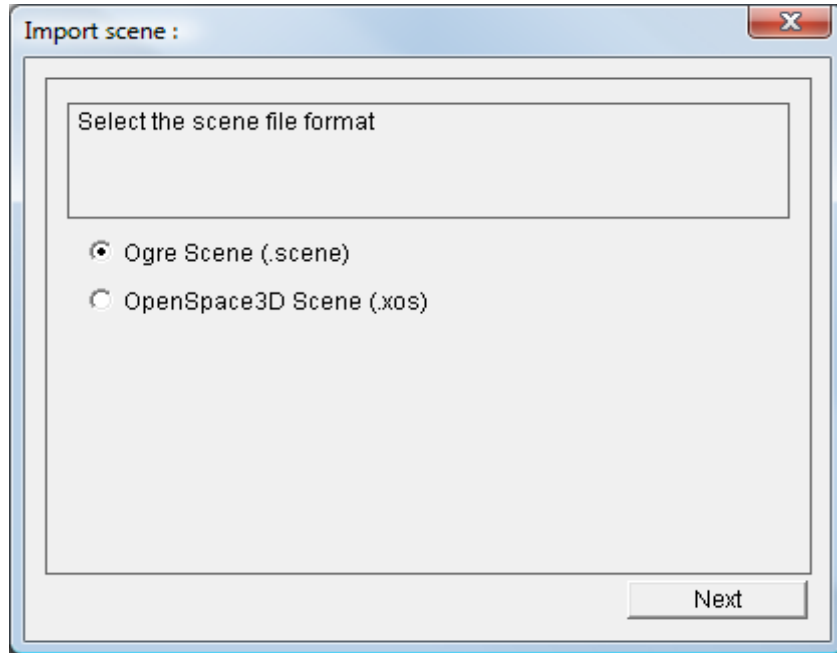
Votre scène est maintenant chargée dans OpenSpace3D, la zone d'Édition 3D affiche les ressources de la scène et la zone d'informations et de gestion des ressources s'est enrichie.

Importer une scène Ogre (.scene)

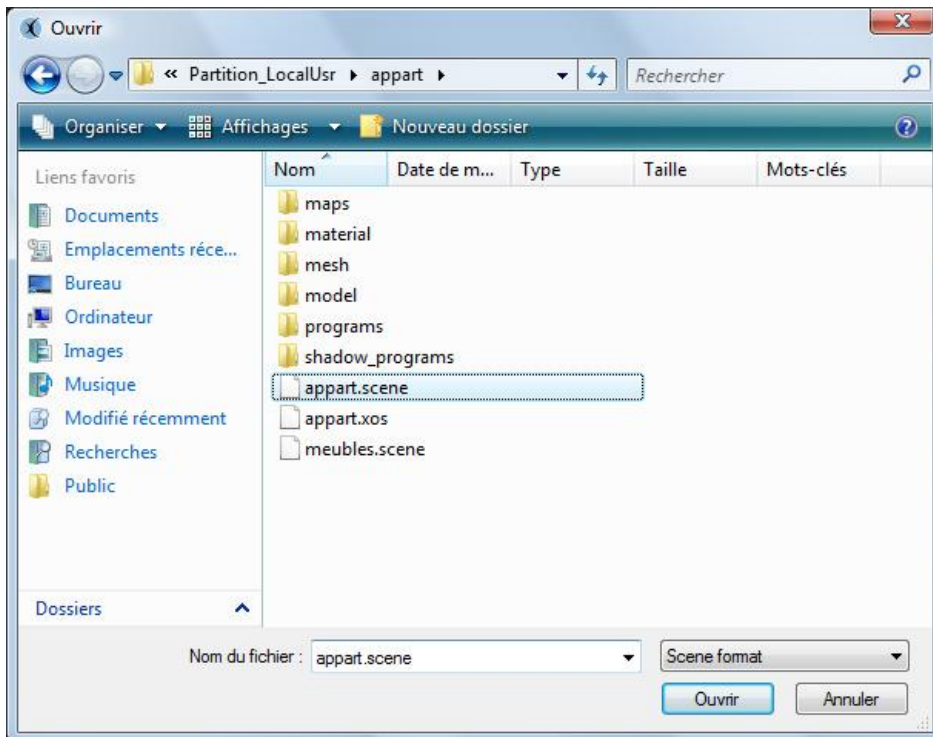
Cliquez sur l'icône « Import Scene »



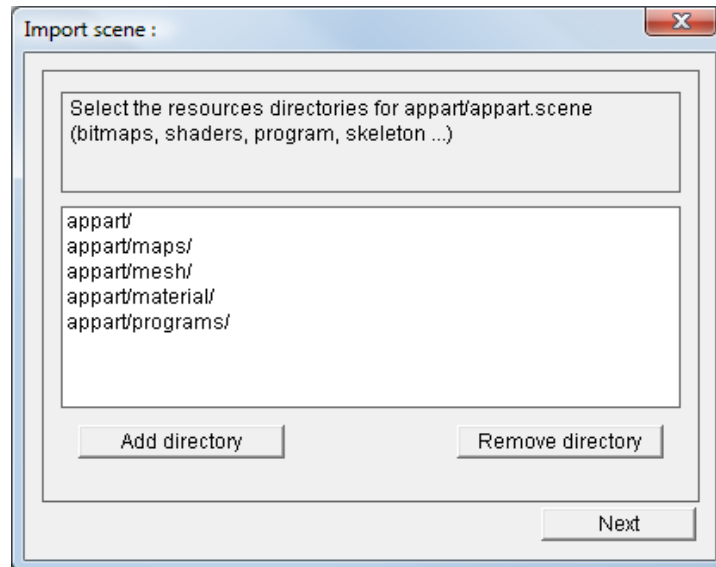
Un assistant apparaît:



Cliquez sur “Next” puis parcourez votre dossier `\Partition_LocalUsr` pour aller sélectionner votre fichier de scène Ogre Scene à ouvrir :



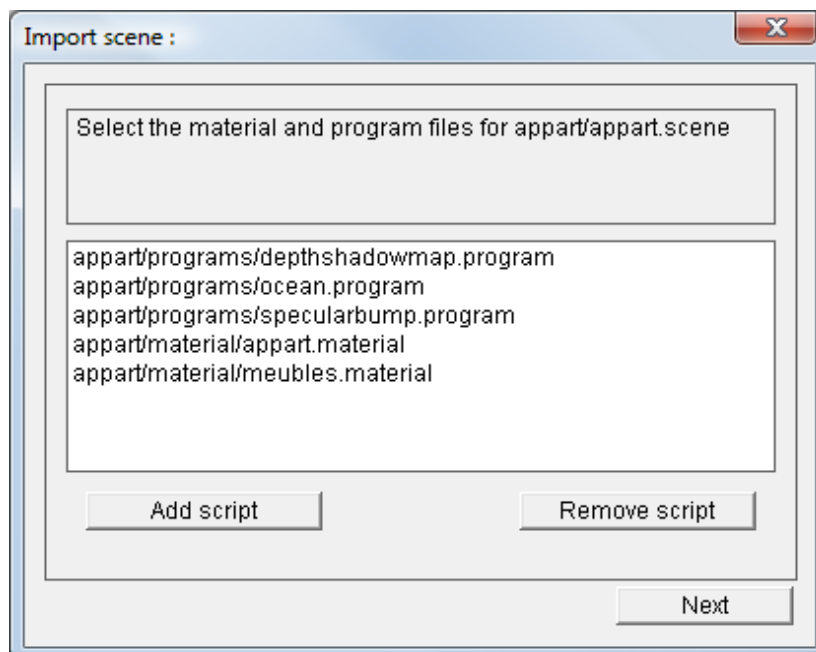
Si les informations sont présentes dans le fichier les dossiers de ressources seront automatiquement ajoutés.



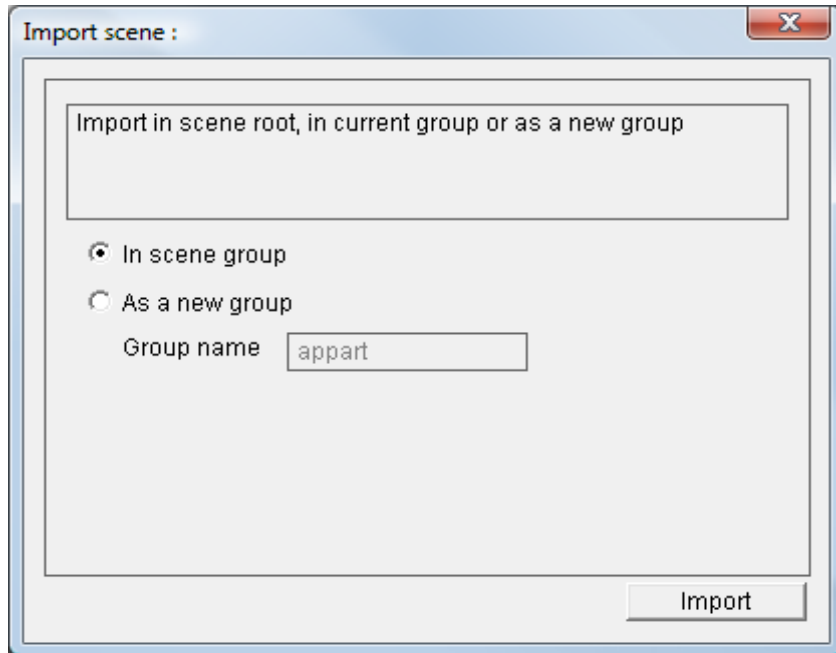
Sinon il vous faudra préciser les dossiers où se trouvent les ressources (textures, mesh, material, shaders ...)

NB : vous pouvez également importer vos ressources dans OpenSpace3D : .xos, .scene ou mesh en utilisant la fonction de Drag and Drop (Glisser/Déposer) depuis votre explorateur Windows vers la zone d'Édition 3D d'OpenSpace3D.

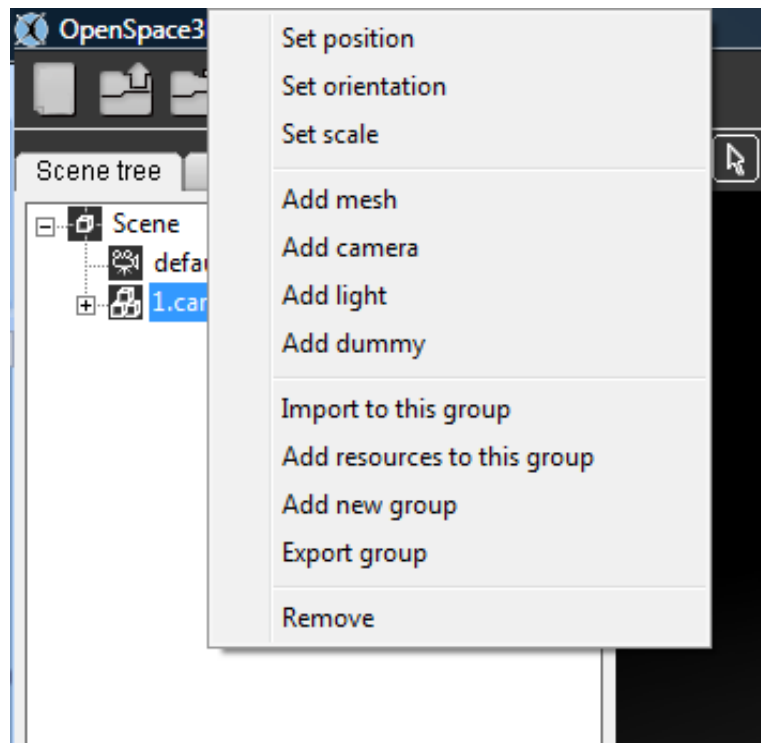
À l'étape suivante vous devrez renseigner les fichiers de script material et program nécessaires à votre scène, les scripts trouvés dans les dossiers seront automatiquement ajoutés.



Vous pouvez ensuite choisir de quelle façon votre scène sera importée dans votre scène courante. Soit dans le groupe par défaut "Scene", soit dans un nouveau groupe.



Si vous avez accédé à l'assistant d'importation depuis le menu d'un groupe sa sélection sera aussi possible.



Enregistrer une scène OpenSpace3D (.xos)

Cliquez sur le bouton « Save »



Raccourci clavier : CTRL+S

Export d'applications développées sous OpenSpace3D

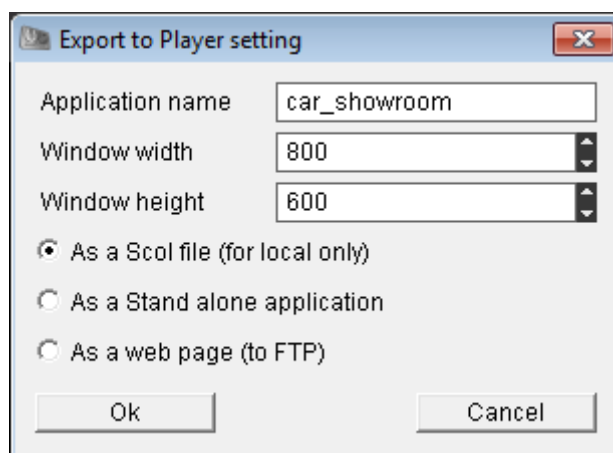
Cette partie concerne les possibilités d'export d'une application développée sous OpenSpace3D.

Sous OpenSpace3D, l'interface d'export est accessible via l'icône



Info Bulle: « Export to OpenSpace3D Player »

Cette Interface s'affiche:



Le paramètre « Application Name » permet de définir le nom de l'application qui sera affiché dans la fenêtre d'OpenSpace3D Player.

Le paramètre « Window width » permet de définir la largeur de la fenêtre d'OpenSpace3D Player.

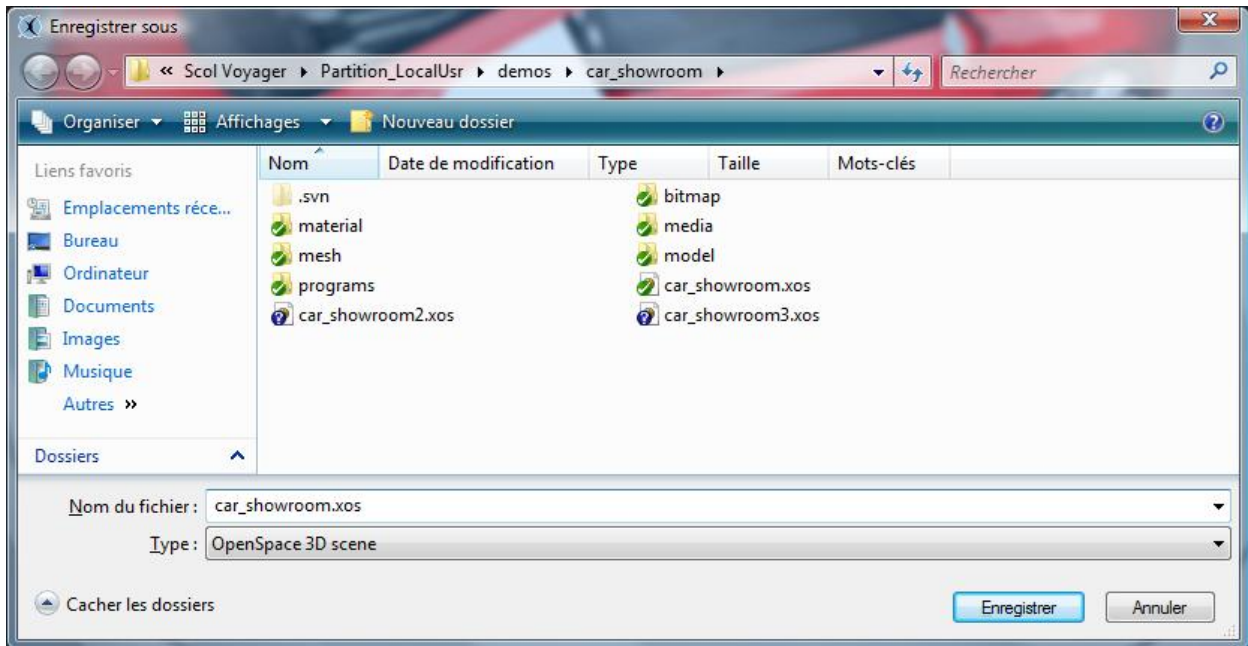
Le paramètre « Window height » permet de définir la hauteur de la fenêtre d'OpenSpace3D Player.

Vous avez ensuite le choix de créer un export parmi trois possibilités :

As a Scol File

Un fichier exécutable .scol qui vous permettra de lancer l'application directement dans le Player (sur la machine de développement seulement).

Après validation de ce type d'export, lorsque vous cliquez sur le bouton « Ok » une fenêtre de sélection de fichier apparaît.



Lorsque vous cliquez « Enregistrer » un fichier permettant de lancer votre scène directement dans OpenSpace3D Player est créé. (Attention ce fichier ne contient aucune ressource de votre scène).

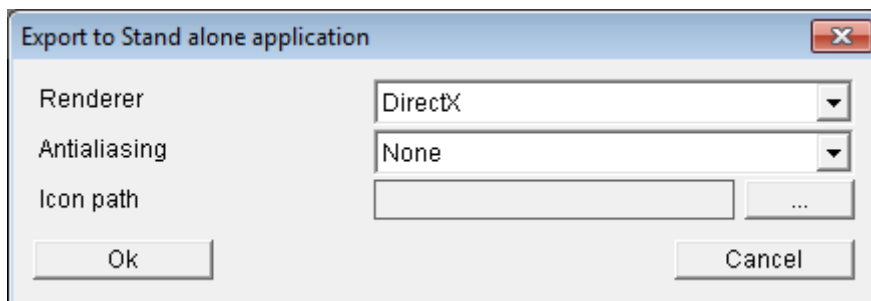
Il vous suffit ensuite de double cliquer sur le fichier pour lancer votre scène dans OpenSpace3D Player.

As A Stand Alone Application

Dans ce cas de figure, OpenSpace3D vous assiste dans le but d'exporter votre application en tant que stand alone exécutable. Celui-ci contiendra toutes les ressources ainsi que le voyageur nécessaire au lancement de votre projet.

Ce mode est très utile dans le cas où l'on souhaite distribuer une application sur un CD, une clé USB ou encore si l'on souhaite compiler un setup d'installation.

Après avoir choisit cette option, une nouvelle fenêtre apparait :



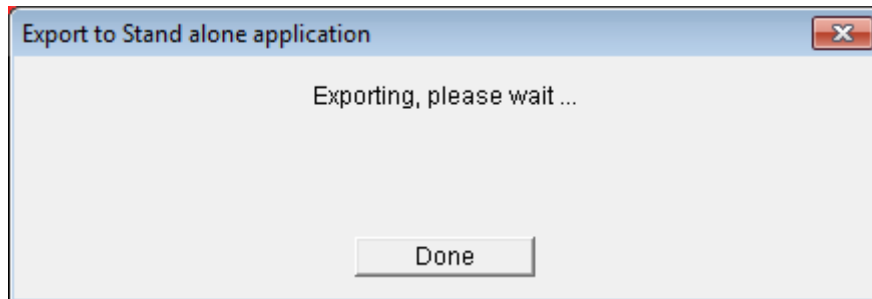
Elle permet de paramétrer certaines options de votre application :

« Renderer » : Permet de choisir le mode de rendu middleware (OpenGL/DirectX)

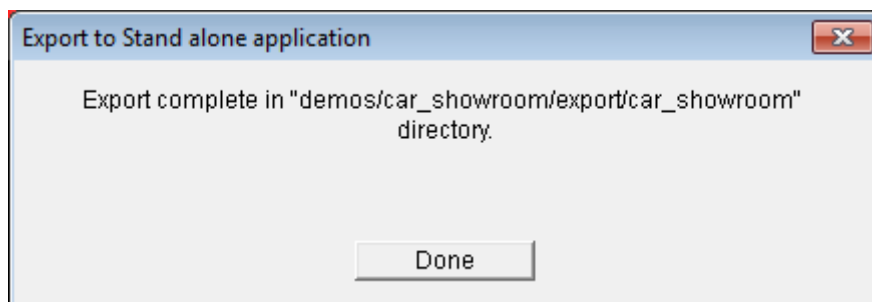
« Anti aliasing » : Permet de choisir le niveau d'Anti-aliasing de l'application.

« Icon Path » : Permet de choisir directement l'icône de votre exécutable.

Après avoir validé cette interface, OpenSpace3D vous indique qu'il est en train d'exporter votre application (il copie tous les fichiers nécessaires)



Une fois ceci terminé, OpenSpace3D indiquera :

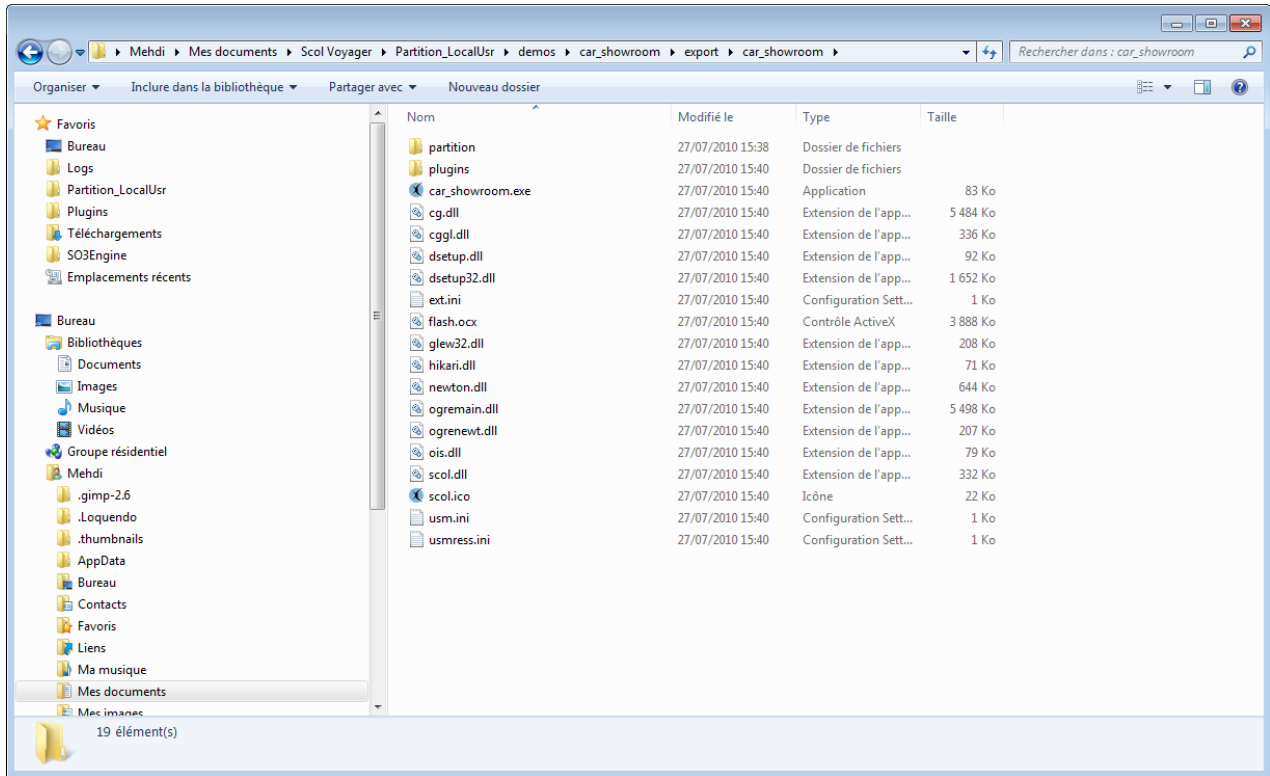


Votre application développée a été exportée dans le dossier stipulé par cette dernière interface, dans notre cas :

« demos/car_showroom/export/car_showroom »

N.B : Ce chemin est relatif à votre partition scol de travail

Si l'on se rend à la racine de ce dossier, on constate l'arborescence suivante :



Il vous suffit de lancer l'exécutable :

Ici : car_showromm.exe

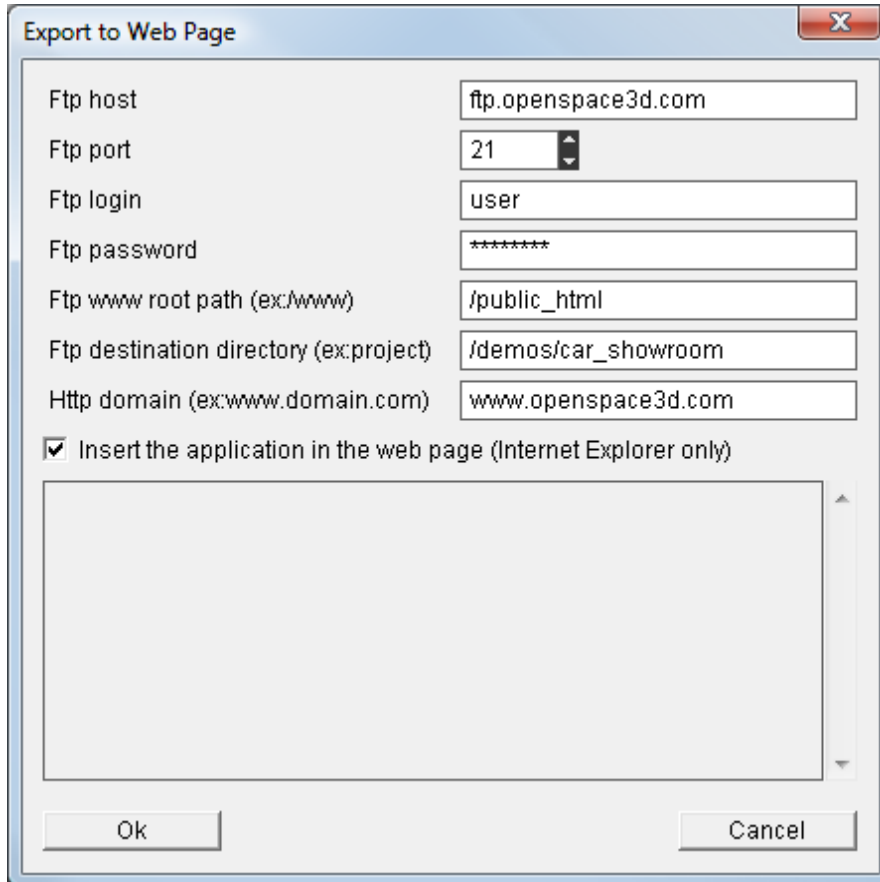
N.B : Toutes les ressources OpenSpace3D nécessaires au fonctionnement de votre application se trouvent dans le dossier partition/nom_du_projet

N.B 2 : On remarque en fait qu'un voyageur complet a été copié ! Ce qui rend l'application portable !

As a Web Page

Offre la possibilité d'exporter l'application sur un serveur Web / Ftp en tant que page web.

Lorsque vous cliquez sur le bouton « Ok » une fenêtre de paramétrage apparaît.



Le paramètre « Ftp host » est l'adresse du serveur ftp.

Le paramètre « Ftp port » est le port du serveur ftp.

Le paramètre « Ftp login » est le login de connexion au serveur ftp.

Le paramètre « Ftp password » est le mot de passe de connexion au serveur ftp.

Le paramètre « Ftp www root path » est le chemin depuis le root ftp vers le dossier contenant le site internet. Par exemple « /public_html/monsite/ ».

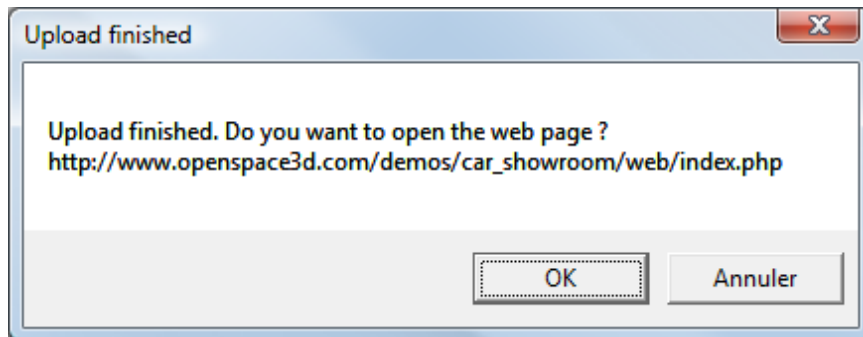
Le paramètre « Ftp destination directory » est le chemin dans lequel l'application sera téléchargée. Par exemple « /demos/mon_application/ ».

Le paramètre « Http domain » est l'adresse http de votre site.

Cochez la case « Insert the application in the web page » si vous souhaitez que l'application 3D soit intégrée dans la page d'un navigateur (Internet Explorer, chrome, firefox..).

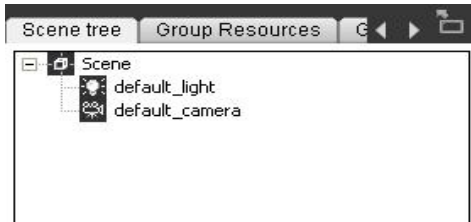
Lorsque vous cliquez le bouton « Ok », la procédure de connexion et de téléchargement commence.

A la fin de la procédure une pop-up d'alerte vous proposera de tester l'application.



Gestion de l'arbre de Scène et d'informations sur les Ressources

L'interface générale de cet outil est la suivante :

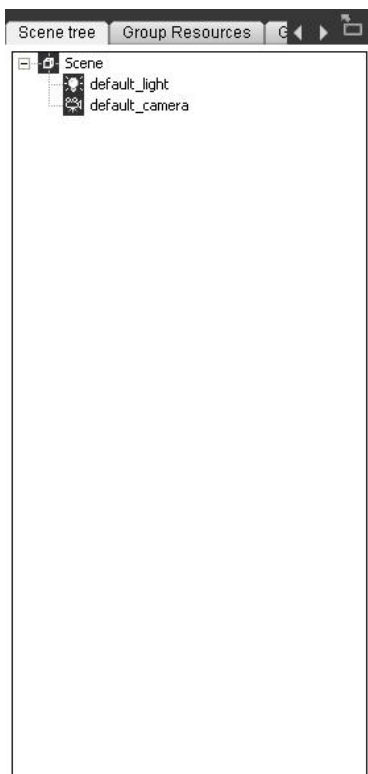


Cette interface est composée de plusieurs onglets.

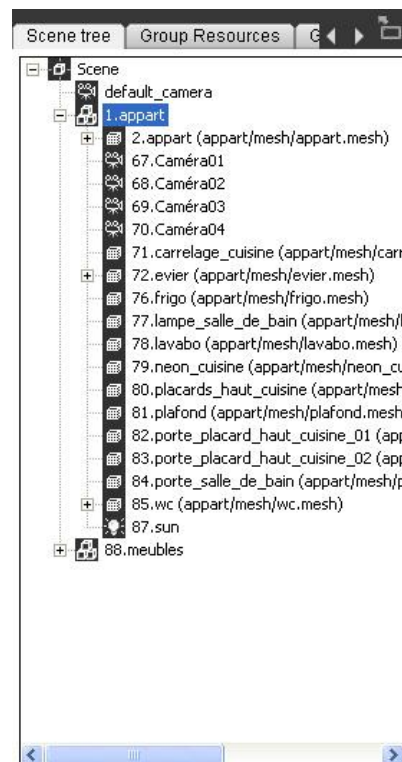
- Scene Tree
- Group Resources
- Group Meshes
- Resources Directories

Scene Tree

Exemple d'arbre de scène vierge




Exemple d'arbre de scène ouvert





Chacun des éléments de l'arbre d'une scène chargée dans OS3DEditor est représentés par une icône qui le caractérise :

N.B : C'est par un clic droit sur les différents éléments de l'arbre de scène que l'on accède aux différents paramètres relatifs à chaque objet de l'arbre



Scène : « C'est la représentation du nœud principal de l'application 3D »

- Set background color
- Set ambient color
- Set shadow methode
- Set fog setting
- Set material scheme
- Set physic setting

- Show scene grid
- Show scene helpers
- Show scene infos
- Auto fit on select

- Add sky box
- Add sky dome
- Add sky plan
- Add compositor
- Add mesh
- Add camera
- Add light
- Add dummy
- Add particle system

- Import to scene
- Add resources to scene
- Add new group

Set Background color : Fixe la couleur de fond de la fenêtre 3D (Cf Color Map)

Set Ambient color : Fixe la couleur de la lumière ambiante de la scène.(Cf Color Map)

Set Shadow Method : Options sur les ombres temps réelles dans la scène (Cf. 24 Utilisation Avancée et définitions)

Set Fog setting : Paramètres du brouillard de la scène 3D.

Set physic setting : Paramètres physiques de la scène 3D (cf : 24 utilisation avancée : Moteur Physique)



Show scene grid : Permet d'afficher ou de cacher la grille dans la scène 3D.

Show scene helpers : Permet d'afficher ou de cacher les icônes pour les caméras, dummies et lumières.

Show scene infos : Permet d'afficher ou de cacher les informations de rendu.

Auto fit on select : Permet d'activer ou de désactiver le centrage automatique sur un objet lorsqu'il est sélectionné.

Add sky dome : Permet l'ajout d'un ciel de type dome.

Add sky box : Permet l'ajout d'un ciel de type boîte.

Add sky plan : Permet l'ajout d'un ciel de type plan.

Add mesh : Permet l'ajout d'un mesh dans la scène (Cf. 24 Utilisation Avancée et définition)

Add compositor : permet l'ajout de compositor dans la scene

Add Camera : Permet l'ajout d'une caméra

Add Light : Permet l'ajout d'une lumière

Add Dummy : Permet l'ajout d'un nœud de scène dans la scène (Cf 24 Utilisation Avancée et définition)

Add particle sytem : permet l'ajout d'un système de particules dans la scene

Import to scene (Cf. 24 Utilisation Avancée et définition)

Add ressources to scene : Permet l'ajout d'une ressource graphique dans la scène (Cf. 24 Utilisation Avancée et définition)

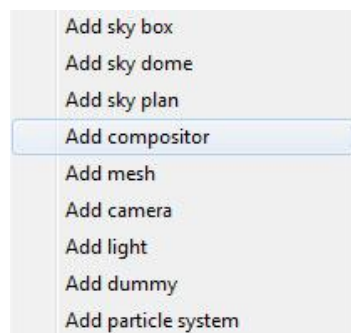
Add new Group (Cf. 24 Utilisation Avancée et définition)

Compositors : "Représentation d'un compositors"

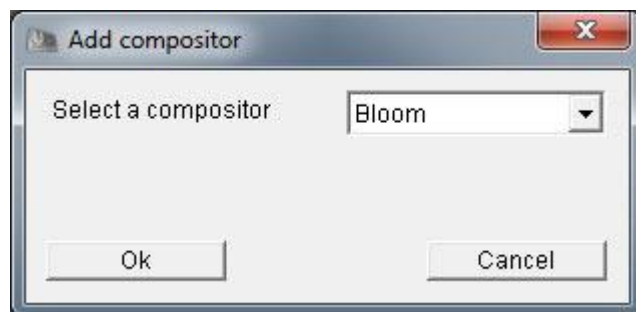
Pour charger un système de particule dans votre scène vous devez préalablement avoir mis vos fichiers ".material, .compositors et les ressources associées (shaders) " dans votre dossier Partition_LocalUsr sous le répertoire Scol Voyager de vos documents

Une fois cette opération effectuée charger vos ressources dans la scène : clic droit "Scene" puis "Add ressources to scene"

Vous pouvez désormais ajouter un compositors grâce au menu "Add Compositor" depuis le menu clic droit sur le noeud de Scene



Sélectionnez le compositors que vous souhaitez ajouter depuis la liste déroulante :

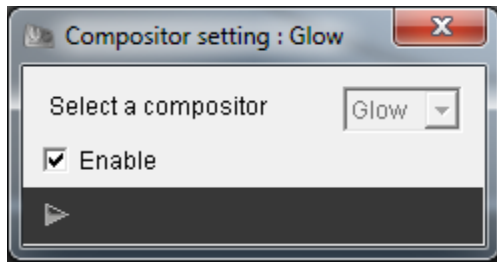


Renommez-le pour mieux repérer vos différents compositors dans la scène puis cliquez sur "Ok"

Pensez à activer le compositors en effectuant un clic droit sur le compositors dans l'arbre de scène



Utilisez le PlugIt Compositor pour faire intégrer le compositors avec d'autres PlugITs



Explications des compositors

Les Compositors permettent d'effectuer des rendus en ajoutant des filtres de post-processing sur l'image finale.

Ceci permet notamment d'obtenir des effets tels que le rendu noir et Blanc, sépia ou encore un rendu caméra nocturne.

Une bibliothèque de compositors est fournit avec OpenSpace3d et vous permettra de tester, modifier ou recréer certains effets dans votre applications.

Quelques notions

```
// Black and white effect
compositor B&W
{
  technique
  {
    // Temporary textures
    texture rt0 target_width target_height PF_A8R8G8B8

    target rt0
    {
      // Render output from previous compositor (or original scene)
      input previous
    }

    target_output
    {
      // Start with clear output
      input none
      // Draw a fullscreen quad with the black and white image
      pass render_quad
      {
        // Renders a fullscreen quad with a material
        material Ogre/Compositor/BlackAndWhite
      }
    }
  }
}
```

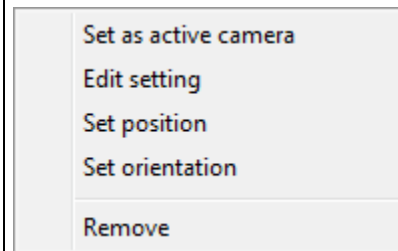
```

        input 0 rt0
    }
}
}
}

```

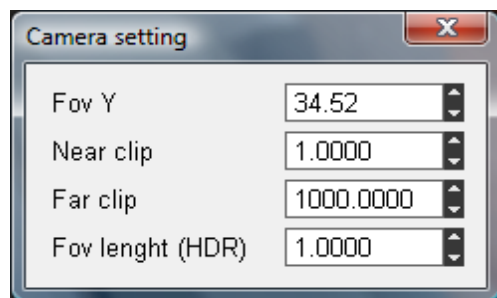
Tout comme les material, les compositors autorisent l'utilisation de plusieurs techniques concernant le rendu final.

Le moteur 3D calcul dans une texture l'image rendu de l'application et applique cette texture sur un material qui permet d'obtenir l'effet souhaité.

**Camera : « Représentation d'une caméra 3D »**

Set as Active Camera : Permet de rendre la camera sélectionnée active dans la zone 3D

Edit Setting :



En entrant une valeur dans la zone de texte ou par clic Gauche + déplacement sur les flèche haut/bas

On modifie immédiatement la valeur en dynamique

Pour plus d'informations concernant ces paramètres se reporter à la partie 24 Utilisation avancée et Définitions.

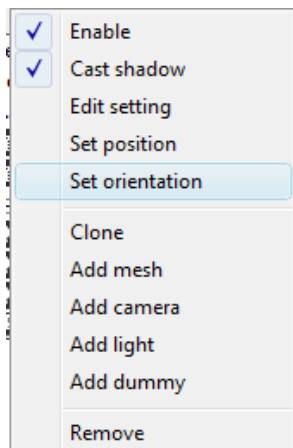
Set position : permet de définir la position de la caméra

Set orientation : permet de définir l'orientation de la caméra

Remove : Supprime la camera de la scène 3D



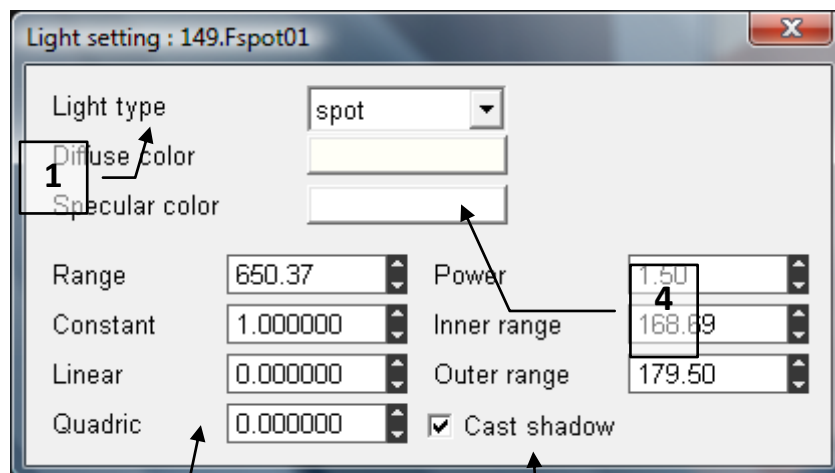
Light : « Représentation d'une lumière 3D »



Enable : Permet d'activer ou de désactiver la lumière.

Cast Shadow : Permet d'activer ou de désactiver la projection des ombres par cette lumière.

Edit Setting :



2

3

1. Light Type :

Le menu déroulant permet de choisir le type de lumière voulue dans la scène

Trois types : Spot, Directionnel, Point. (Cf. 24 Utilisation Avancée et définition)

2. Les Paramètres de lumière

En entrant une valeur dans la zone de texte ou par clic Gauche + déplacement sur les flèche haut/bas, on modifie immédiatement la valeur en dynamique.



(Cf. 24 Utilisation avancée et Définition).

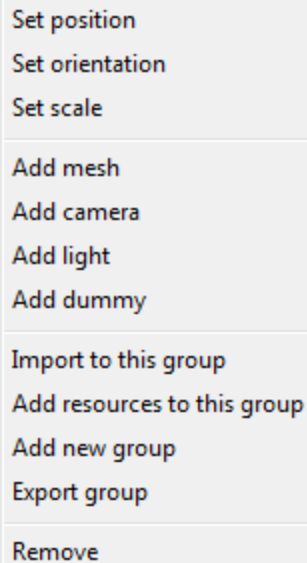
N.B : Les derniers paramètres (Inner Range et Outer Range) sont accessibles que pour le cas d'une lumière spot car ils définissent l'ouverture de cette lumière.

3. Cast Shadows

Lorsque cette option est cochée la lumière projettera les ombres dans la scène

4. Diffuse et Specular color

Ces deux options permettent de définir la couleur de la lumière

**Group : « C'est la représentation d'un groupe »** (Cf. 24 Utilisation Avancée et définition)

- Set position
- Set orientation
- Set scale
- Add mesh
- Add camera
- Add light
- Add dummy
- Import to this group
- Add resources to this group
- Add new group
- Export group
- Remove

Set Position : Permet de modifier la position d'un groupe d'Objets

Set Orientation : Permet de modifier l'orientation d'un groupe d'Objets

Set Scale : Permet de modifier l'échelle d'un groupe d'Objets

Add mesh : Permet l'ajout d'un mesh dans la scène (Cf. Wizard mesh chapitre suivant)

Add Camera : Permet l'ajout d'une caméra

Add Light : Permet l'ajout d'une lumière

Add Dummy : Permet l'ajout d'un nœud de scène dans le groupe (Cf. 24 Utilisation Avancée et définition)

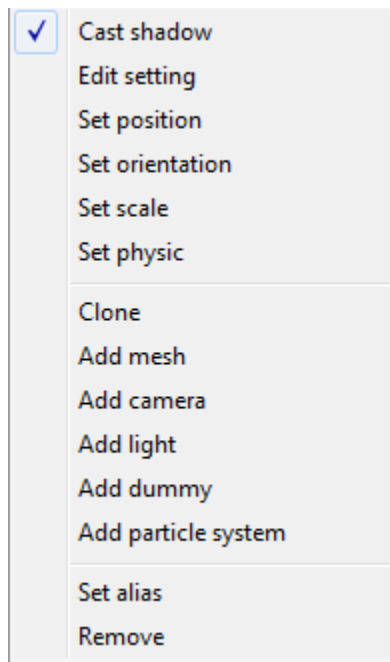
Import to scene (Cf. 24 Utilisation Avancée et définition)

Add resources to scene : Permet l'ajout d'une ressource graphique dans la scène (Cf. 24 Utilisation Avancée et définition)

Add new Group (Cf. 24 Utilisation Avancée et définition)

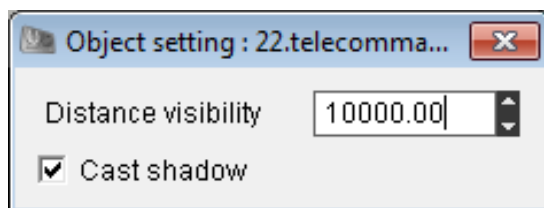
Export Group (Cf. 24 Utilisation Avancée et définition)

Remove : Suppression du groupe et de son contenu

**Mesh : « c'est la représentation d'un Objet 3D »**

Cast Shadow : Option pour spécifier si le mesh va projeter des ombres

Edit setting : permet de configurer des options spécifiques sur l'objet



Distance Visibility : Distance à partir de laquelle l'objet ne sera plus visible (LOD tout ou rien)

Cast Shadow : Autorise la projection d'ombres par l'objet

Set Position : Permet de modifier la position de l'objet

Set Orientation : Permet de modifier l'orientation de l'objet

Set Scale : Permet de modifier l'échelle de l'objet

Clone : Permet la duplication de l'objet dans l'arbre de scène (cf : 10. Edition 3D : duplication d'Objet)

Add Mesh : Ajoute un mesh à l'objet (Cf. 24 Utilisation avancée et Définitions)

Add Camera : Ajoute une camera fille de l'objet

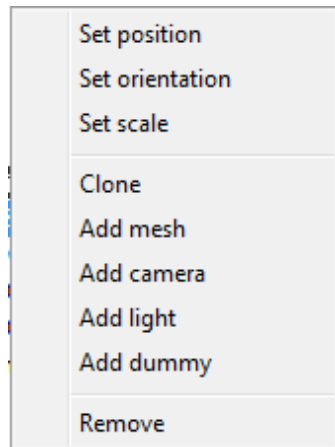
Add Light : Ajoute une lumière fille de l'objet

Add Dummy : Ajoute un nœud de scène sur l'objet

Remove: Suppression de l'objet et de ses fils



Nœud : « représentation d'un nœud de scène » (Cf. 24 Utilisation Avancée et définition)



Set Position : Permet de modifier la position du nœud

Set Orientation : Permet de modifier l'orientation du nœud

Set Scale : Permet de modifier l'échelle du nœud

Clone : duplique un nœud et sa hiérarchie

Add Mesh : Ajoute un mesh au nœud (Cf. 24 Utilisation avancée et Définitions)

Add Camera : Ajoute une camera fille du nœud

Add Light : Ajoute une lumière fille du nœud

Add Dummy : Ajoute un nœud de scène sur le nœud

Remove: Suppression du nœud et de ses fils

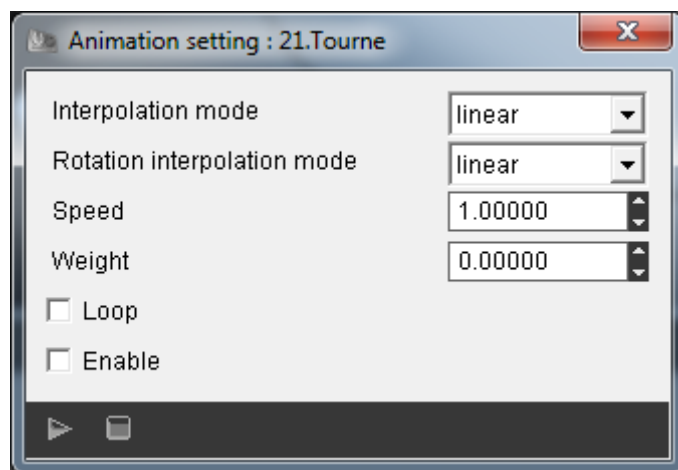
**Animation : « Représentation d'une animation »**

<input checked="" type="checkbox"/>	Enable
<input checked="" type="checkbox"/>	Loop
	Edit setting

Enable : Spécifie que l'animation sera active au lancement de la scène

Loop : Spécifie que l'animation sera jouée en boucle

Edit Setting : Configure les options de lecture de l'animation



Interpolation Mode : Permet de définir le type de mode d'interpolation entre les clés d'animations

Rotation interpolation Mode : Permet de définir les types d'interpolations entre les animations Concernant les rotations

Speed : Permet de définir la vitesse de lecture d'animation c'est-à-dire le temps qui sera passé entre chaque clés d'animation

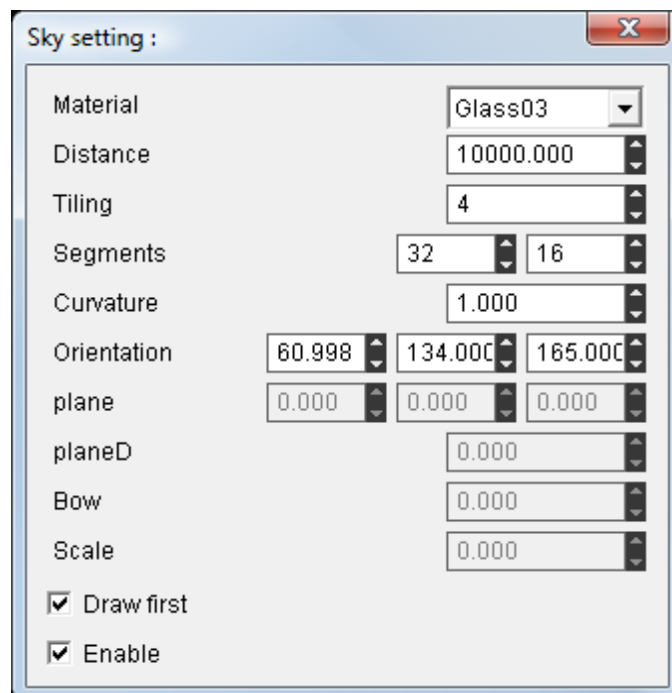
 Sky : « Représentation d'un Ciel »

Edit setting
Remove

Edit Setting : Configure les options du ciel

Remove: Suppression du ciel

Edition des paramètres d'un ciel (cf 24. Utilisation avancée : skies)



Material : Permet de sélectionner le matériau pour le ciel (pour ajouter des matériaux vous devez ajouter les ressources manuellement dans la scène)

Distance : distance du ciel

Tiling : Nombre de répétition de la texture

Segments : Nombre de segments X Y sur l'objet créé

Curvature : Quantité de courbure du ciel

Orientation : Orientation du ciel

Plane : vecteur pour le type plan

PlaneD : Distance du ciel pour le type plan



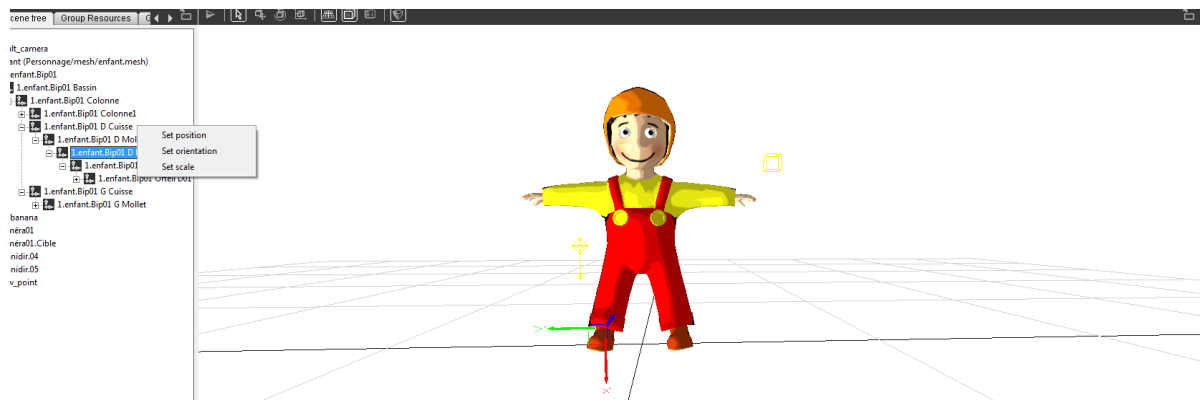
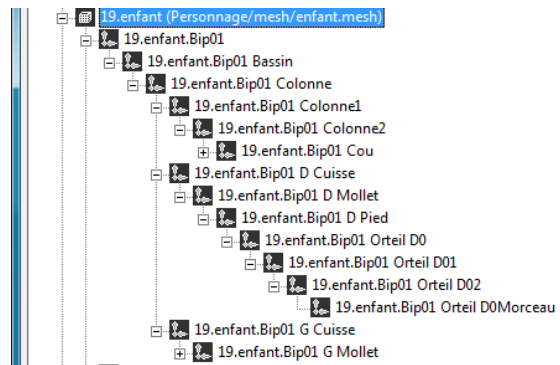
Bow : Quantité de courbure pour le type plan

Scale : Echelle du ciel pour le type plan

Draw first : Défini si le ciel est affiché avant les objets de la scène

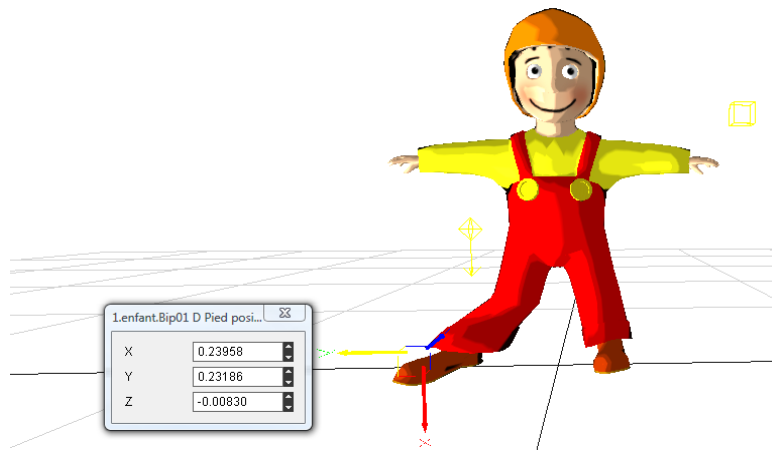
Enable : Active ou désactive le ciel

Bones : "Représentation des bones d'un objet"

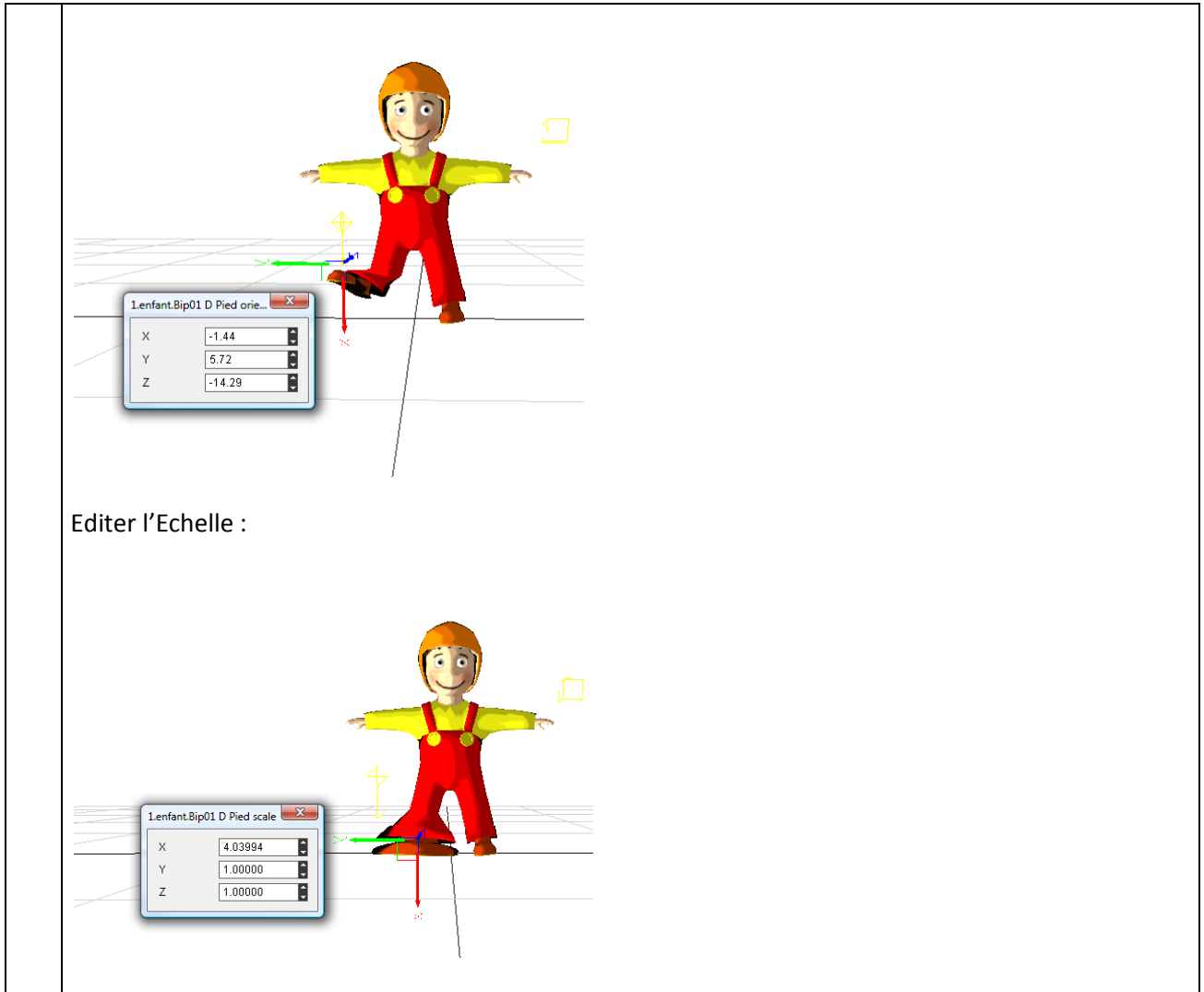


Sur chacun des bones vous pouvez :

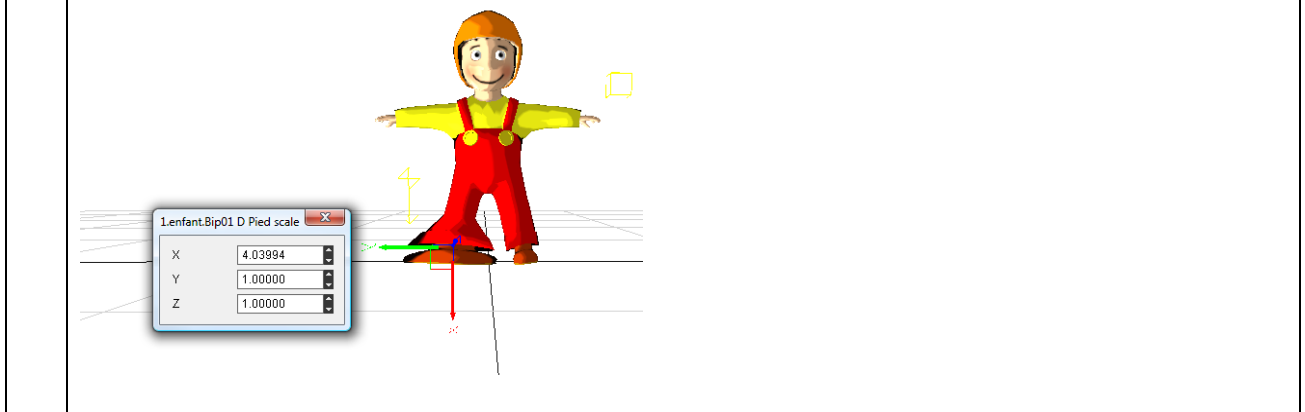
Editer la position :



Editer l'orientation :



Editer l'Echelle :

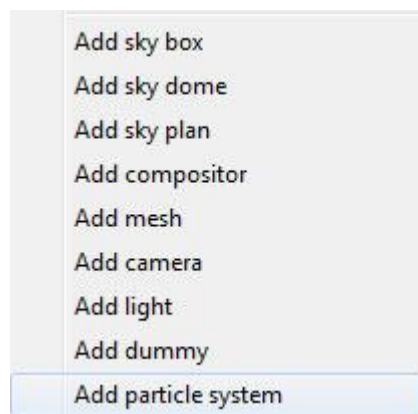


Particle system : Système de Particules :

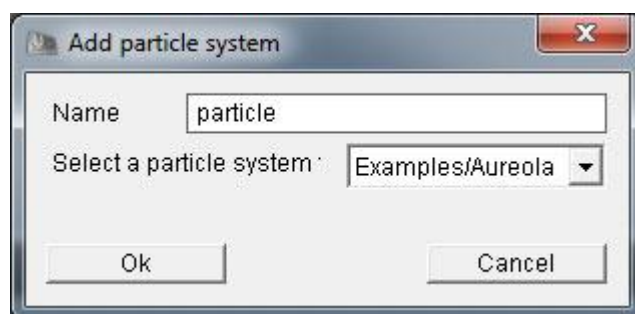
Pour charger un système de particule dans votre scène vous devez préalablement avoir mis vos fichiers ".material, .png et .particle" dans votre dossier Partition_LocalUsr sous le répertoire Scol Voyager de vos documents

Une fois cette opération effectuée charger vos ressources dans la scène : clic droit "Scene" puis "Add ressources to scene"

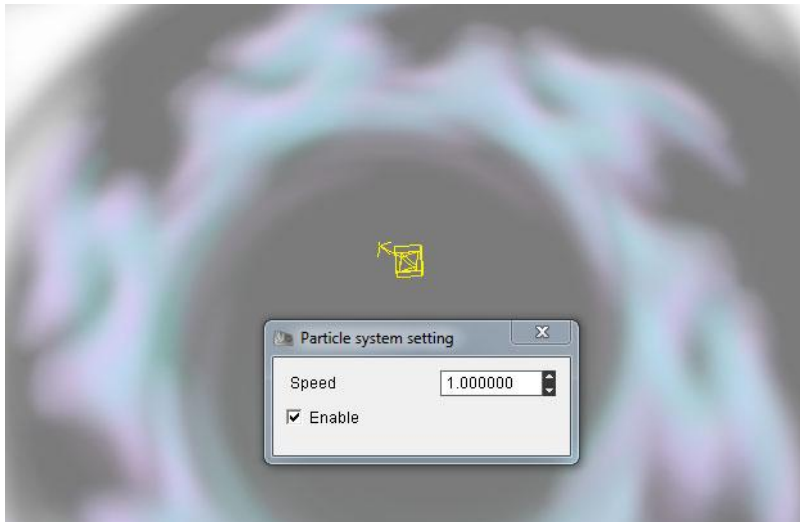
Vous pouvez désormais ajouter un système de particules grâce au menu "Add particle system" depuis le menu clic droit sur le noeud de Scene



Sélectionnez le système de particules que vous souhaitez ajouter depuis la liste déroulante :



Renommez-le pour mieux repérer vos différents systèmes de particules dans la scène puis cliquez sur "Ok"



Choisissez la vitesse et cochez la case "Enable" pour l'activer.

Les scripts de particules

Les particules dans OpenSpace3D reposent sur le système de particules d'Ogre3D. Ainsi, Ogre3D gère les données relatives aux particules par l'utilisation de script de particules. Dans les ressources disponibles avec OpenSpace3D, certains de ces scripts sont fournis. Ce qui vous permettra de modifier ou de créer vous-même d'autres scripts pour des effets de particules différents. Ceci à partir des templates Ogre3D prédéfinis ou via la création de nouveaux scripts

Les particules scripts sont contenus dans les fichiers possédant l'extension « .particle ». Ci-dessous, un exemple de particule script. Nous détaillons ensuite son fonctionnement et le vocabulaire qui lui est associé.

```
// Exudes aureola particles which around the model float upwards
particle_system Examples/Aureola
{
    material    Examples/Aureola
    particle_width 200
    particle_height 200
    cull_each    false
    quota        100
    billboard_type perpendicular_common
    common_direction 0 1 0
    common_up_vector 0 0 1

    // Area emitter
    emitter Box
    {
        angle        30
```



```
emission_rate 4
time_to_live 5
position 0 -100 0
direction 0 1 0
velocity_min 0
velocity_max 30
colour_range_start 0.3 0.3 0.3 0.0
colour_range_end 0.8 0.8 0.8 0.0
width 10
height 10
depth 30
}

// Make em float upwards
affector LinearForce
{
    force_vector 0 70 0
    force_application add
}

// Fader
affector ColourFader2
{
    red1 +0.4
    green1 +0.4
    blue1 +0.4
    alpha1 +0.7

    red2 -0.25
    green2 -0.25
    blue2 -0.25
    alpha2 -0.3333

    state_change 3.5
}

// Rotater
affector Rotator
{
    rotation_range_start 0
    rotation_range_end 360
    rotation_speed_range_start 0
    rotation_speed_range_end 180
```


}

}

Explications des particules scripts

Particle system :

La résultante visuelle d'un script de particules dans OpenSpace3d repose sur la notion de particule système. Un système de particules contient plusieurs types d'objets, nous décrivons plus après certaines des notions importantes à connaître pour une bonne manipulation des systèmes de particules :

- **Material** : Définit le nom du material que toutes les particules dans ce système utiliseront. Toutes les particules dans un système utilisent le même material, bien que chaque particule puisse coloriser ce material grâce à l'utilisation de sa propriété de couleur.
- **particle_width** : Définit la largeur des particules dans le monde. Notez que cette propriété est absolue.
- **particle_height**: Définit la hauteur des particules dans le monde. Notez que cette propriété est absolue.
- **quota** : Définit le nombre maximum de particules que ce système peut contenir à la fois. Lorsque cette limite est épuisée, les émetteurs ne seront pas autorisés à émettre des particules en plus.
- **Cull each**: Tous les systèmes de particules sont englobés par la boîte. Elle contient toutes les particules du système. Ceci est normalement suffisant pour les systèmes de particules assez limités localement où la plupart des particules sont visibles ou non. Cependant, pour les cas de particules réparties sur une zone plus large (par exemple un système de pluie), vous voudrez peut-être effectivement qu'au niveau du temps, chaque particule se manage individuellement. Vous faites cela en définissant le paramètre `cull_each` à `true`.
- **Sorted** : Par défaut, les particules ne sont pas triées. En fixant cet attribut à `'true'`, les particules seront triées par rapport à la caméra. Cela peut rendre certains effets à un coût de tri de plus petit.
- **Local space** : Par défaut, les particules sont émises dans l'espace du monde, de sorte que si vous transformez le nœud sur lequel le système est appliqué, cela n'affectera pas les particules (uniquement les émetteurs). Toutefois, pour créer certains effets vous souhaiterez que les particules restent attachées à l'espace local de l'émetteur. Cette option vous permet de le faire.
- **Billboard_type** : Il ya plus d'une méthode pour orienter un billboard .L'approche classique est qu'il soit directement orienté face à la caméra: c'est le comportement par défaut. Toutefois



cette disposition n'est bonne que pour les particules qui représentent quelque chose de vaguement sphérique comme une fusée lumineuse. Pour plus d'effets non linéaires comme le feu au laser, vous voudrez vraiment que la particule possède une orientation propre.

Les options pour ce paramètre sont:

point : La disposition par défaut, cela équivaut à des particules sphériques ou les billboard sont toujours face à la caméra.

oriented_common : Les particules sont orientées autour d'un vecteur directeur général fixe (voir **common_direction**), qui agit comme leur axe Y local. Le billboard tourne seulement autour de cet axe.

oriented_self

Les particules sont orientées autour de leur propre vecteur direction.

perpendicular_common

Les particules sont perpendiculaires à un vecteur directeur général fixe (voir **common_direction**), qui leur sert d'axe local Z.

perpendicular_self

Le vecteur des particules est perpendiculaire à leur propre direction, le système ne tourne pas face à la caméra.

- **Common-direction**: Requis seulement si le type d'affichage est réglé sur **oriented_common** ou **perpendicular_common**, ce vecteur est le vecteur direction commun utilisé pour orienter toutes les particules dans le système. **system.Common_up_vector** : Requis seulement si le type d'affichage est réglé sur **perpendicular_self** ou **perpendicular_common**, ce vecteur est le vecteur commun utilisé pour orienter toutes les particules dans le système.

Particle emitters :

Les émetteurs de particules sont classés par «type». Par exemple : 'Point' émet à partir d'un seul point. Alors que "boîte » émet au hasard dans une région. Vous pouvez ajouter un émetteur à un système, par emboîtement dans un autre, il est reconnu par le mot clé «émetteur» suivie du nom du type (sensible à la casse). Ogre supporte actuellement «Point», «Box», «cylindre», «ellipsoïde», «émetteurs HollowEllipsoid» et «Ring».

- **Angle** : Définit l'angle maximal (en degrés) d'émission des particules. Mettre ce paramètre à 10 permet de dévier les particules jusqu'à 10 degrés dans n'importe quelle direction opposée à la direction de l'émetteur. Une valeur de 180 signifie émettre dans n'importe quelle direction, tandis que 0 signifie émettre toujours exactement dans la direction de l'émetteur
- **Colour** : Définit une couleur statique pour toutes les particules émises.
- **Colour_range_start** and **colour_range_end** : Indique la gamme de couleurs disponibles pour les particules émises. La couleur réelle sera choisie au hasard entre ces 2 valeurs.
- **Direction** : Définit la direction de l'émetteur.
- **Emission_rate** : Définit par seconde l'émission des particules. Le taux d'émission sera également limité le quota.
- **Position** : Définit la position de l'émetteur
- **Velocity** : Définit une vitesse constante pour toutes les particules au moment de l'émission. Voir aussi les attributs **velocity_min** **velocity_max** qui vous permettent de définir une gamme de vitesses au lieu d'une valeur fixe.



- Time_to_live : Définit le nombre de secondes de vie pour chaque particule.
- Duration : Définit le nombre de secondes pendant lesquelles l'émetteur est actif. L'émetteur peut être démarré à nouveau,
- Repeat_delay : Définit le nombre de secondes à attendre avant que l'émission d'un émetteur à durée limitée recommence à émettre.

Particle Affectors :

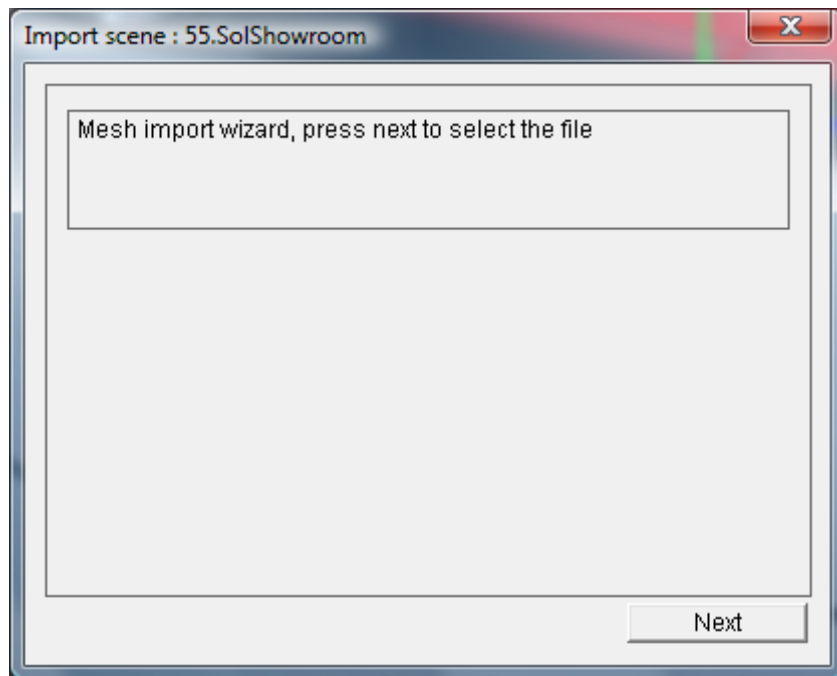
Ils permettent de modifier les particules au cours de leur vie. Ils sont classés par «type». Par exemple : Affectors 'LinearForce' applique une force sur toutes les particules, tandis que Affectors 'ColourFader' altère la couleur des particules en vol.

Pour d'avantage d'explications : voir <http://www.ogre3d.org/>

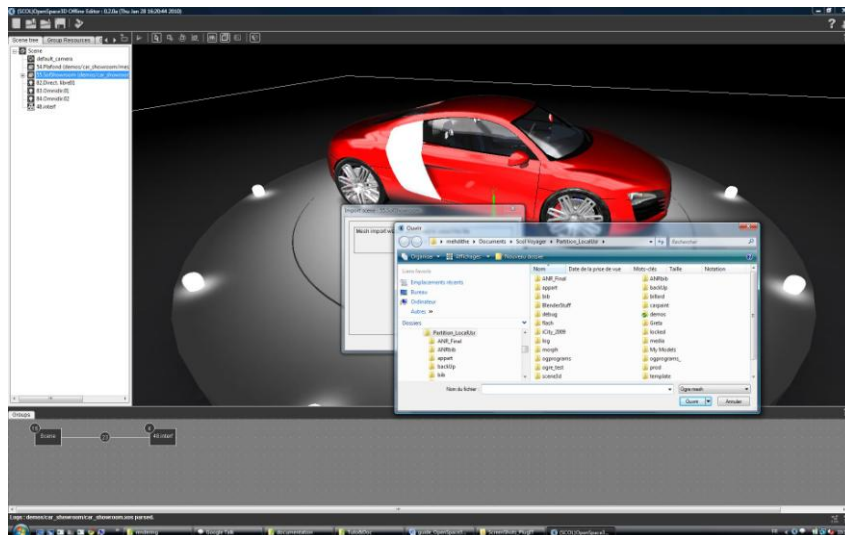
Ajout de mesh : Wizard

Une fonctionnalité très utile sous OpenSpace est la possibilité via un wizard d'ajouter un mesh de manière très simple :

On accède à ce wizard via un clic droit sur l'arbre de scène sur l'un des groupes de la scène ou la scène elle-même :

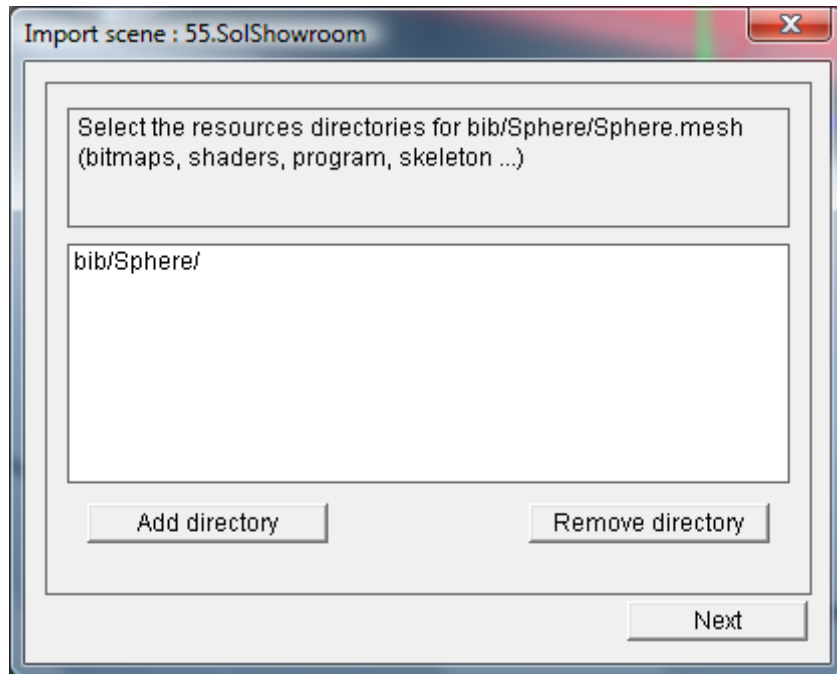


Next :



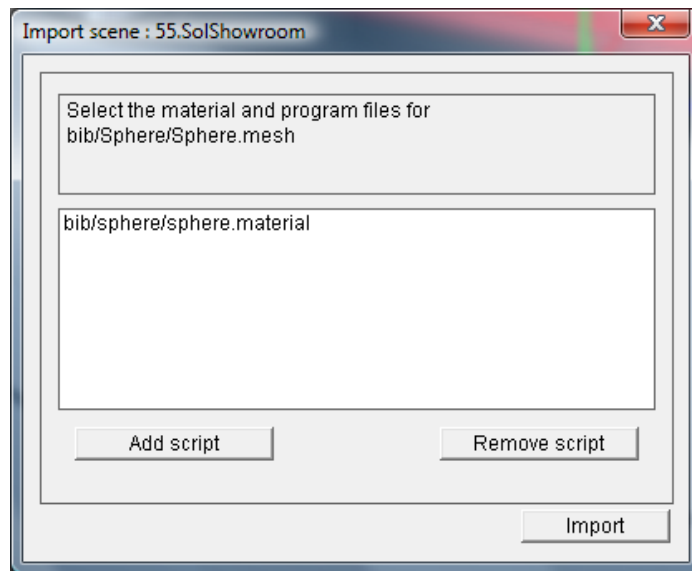
Un explorateur s'ouvre, permettant d'aller chercher le mesh que nous souhaitons importer dans la scène.

Puis valider :

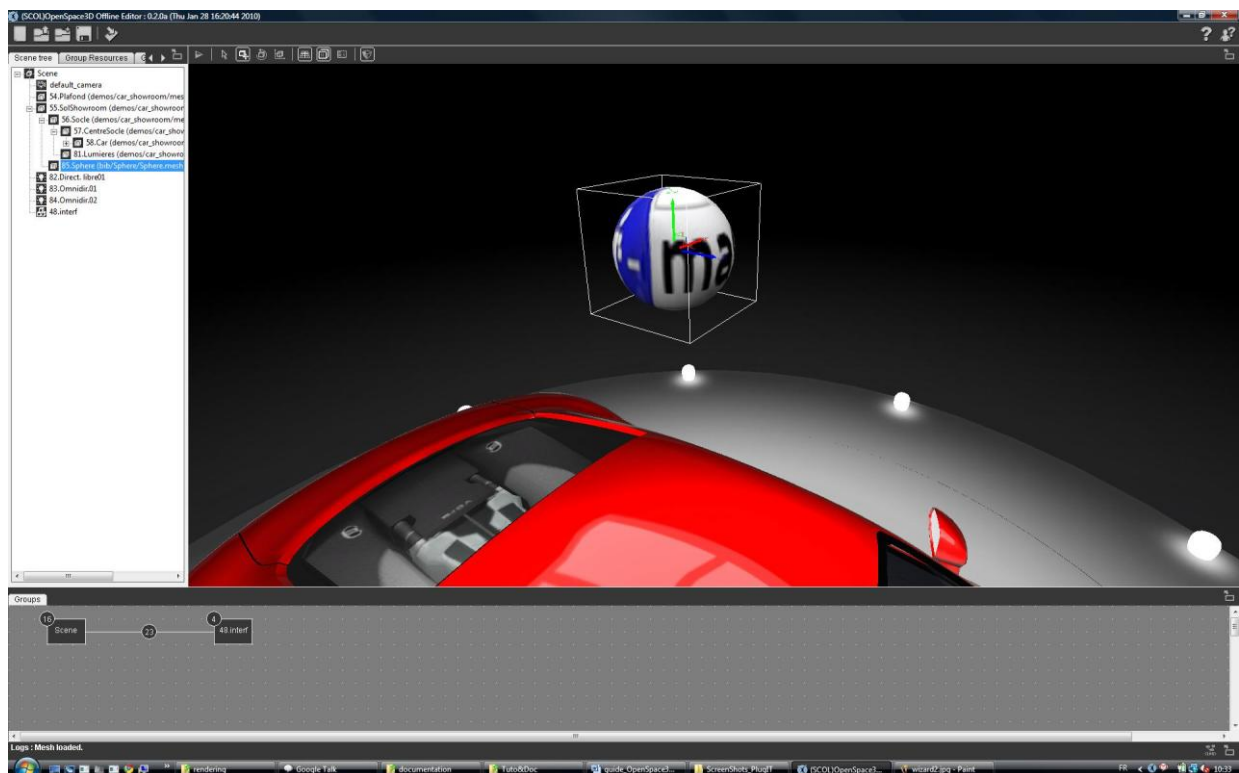


Ici, par exemple, on ajoute le mesh : sphere.mesh ce qui va automatiquement ajouter le répertoire de ressources dans les ressources locations (cf : 24.annexes utilisations avancées et ressources)

A présent, on doit aller chercher via « Add directory » les autres sous-dossiers contenant les différentes ressources relatives à notre mesh (material, textures, programme shaders..)



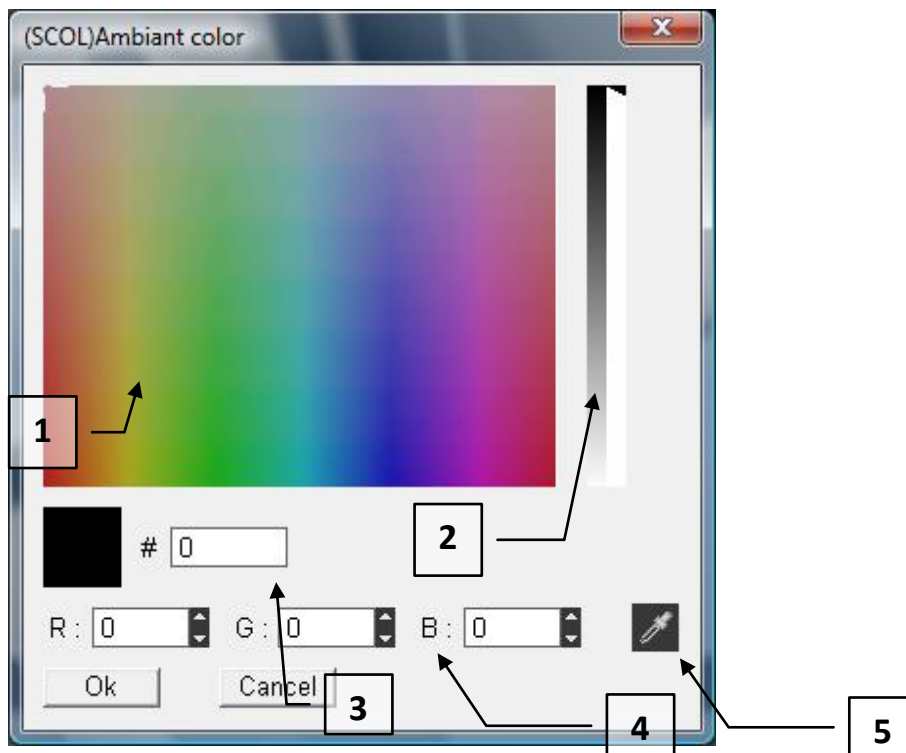
Cette dernière étape, nous permet d'aller ajouter via « Add script » le fichier material du mesh (celui-ci peut-être détecté automatiquement)



Ici, l'objet sphère a été ajouté avec succès.

Color Map

Le clic Gauche sur la zone de couleur permet d'ouvrir une fenêtre de colormap :



1 Sélection de la couleur directement dans la colorMap

2 Sélection de la saturation de la couleur

3 Entrer la couleur directement en Hexadécimal

4 Enter la couleur en RGB

5 Sélection d'une couleur n'importe où sur l'écran

« OK » pour validation

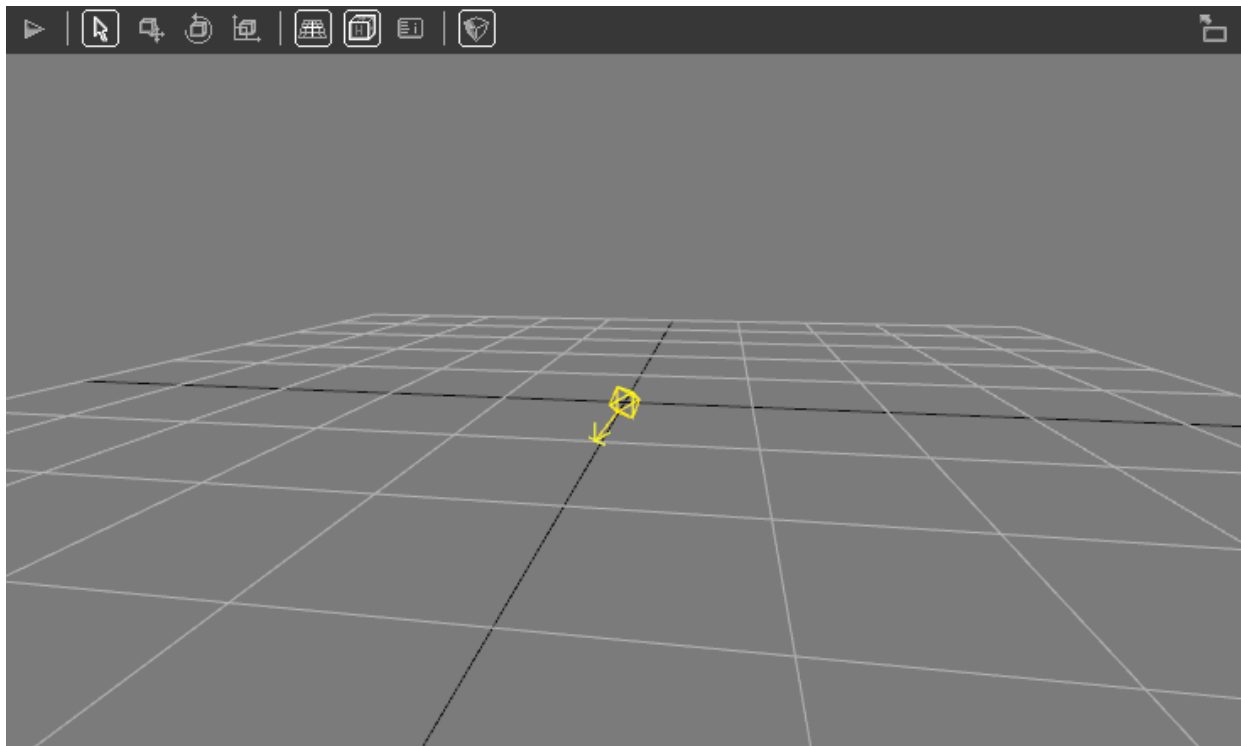
Edition 3D

Interface

C'est la zone 3D permettant de visualiser la scène 3D éditée.












Ainsi, c'est dans cette zone qu'apparaîtront les différents objets.

Cette zone permet également d'aménager les objets importés en modifiant leurs coordonnées spatiales les uns par rapport aux autres en fonction des différents groupes d'objets chargés dans la scène et de la hiérarchie.



L'image ci-dessus nous montre une scène vierge ne contenant que la caméra principale permettant de visualiser la 3D, une lumière et une grille dont le centre représente le point de coordonnées (0, 0,0) de la scène.



	Play : Permet le lancement de l'application 3D (voir Partie édition des fonctions)
	Pause : Stoppe la scène du mode applicatif vers le mode édition
	Move : Permet de modifier la position de l'objet de la scène sélectionnée
	Rotate : Permet de modifier l'orientation de l'objet de la scène sélectionnée
	Scale : Permet de modifier l'échelle de l'objet de la scène sélectionnée
	Show / Hide grid : Affiche ou cache la grille
	Show / Hide helpers : Affiche ou cache les icônes de caméras, lumières et dummies
	Show/Hide Polygon : Affiche ou cache les polygones de la scene (F3)
	Show / Hide 3d infos : Affiche ou cache les informations de rendu de la scène
	Navigate Mode : « Bouton indiquant le mode de navigation dan la 3D en mode édition
	Walk Mode : « Bouton permettant de passer à un mode de navigation vue à la 3 ^{ième} personne »

La barre d'outils se trouvant au dessus de cette zone 3D contient des outils permettant l'édition de cette scène et des options de visualisation de la scène



Déplacement dans la 3D

Le bouton le plus à droite de la barre d'outils permet de choisir le mode de navigation dans la zone 3D.

Ce bouton à deux états (deux modes de navigation) et le clic gauche sur ce bouton permet de passer d'un mode de navigation à l'autre



Déplacement par défaut dans la 3D en mode viewer :

Seule la souris est nécessaire dans ce mode de navigation :

- Clic Gauche + déplacement: Permet la rotation de la vue
- Clic Molette + déplacement : Permet la translation de la vue

Molette avant/arrière : Zoom in/Zoom Out

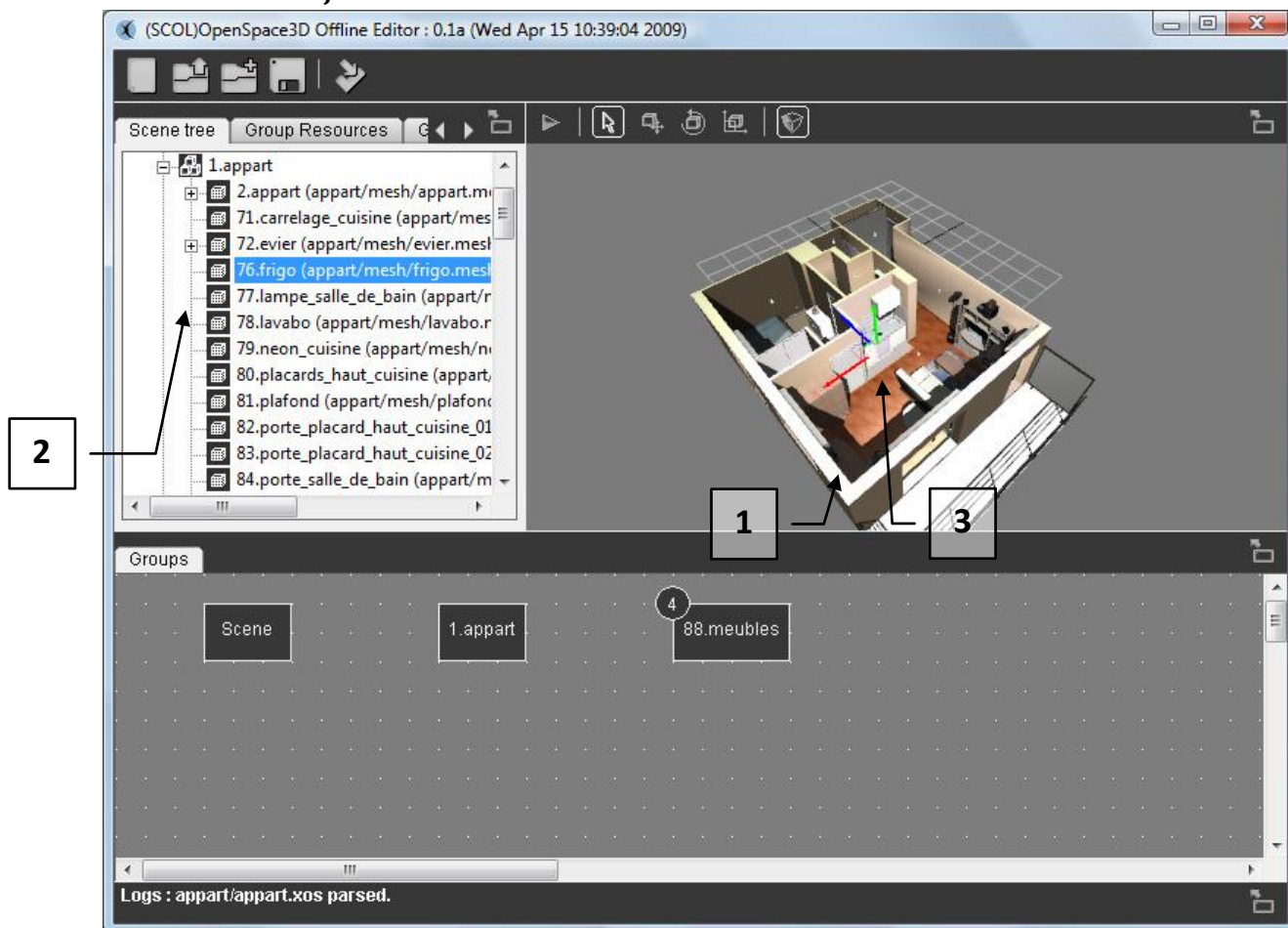


Déplacement en vue 3^{ème} personne :

Comme pour le mode de navigation par défaut seule l'utilisation de la souris est ici nécessaire :

- Clic Gauche + Déplacement vers la Gauche ou flèche de gauche du clavier : Permet d'effectuer une rotation sur la caméra active (je tourne la tête à gauche)
- Clic Gauche + Déplacement vers la Droite ou flèche de droite du clavier : Permet d'effectuer une rotation sur la caméra active (je tourne la tête à droite)
- Clic Gauche + Déplacement vers le haut ou flèche du haut du clavier : Permet de bouger vers l'avant
- Clic Gauche + Déplacement vers le bas ou flèche du bas du clavier : Permet de bouger vers l'arrière
- Bouton Shift appuyer + déplacement : Permet de tourner la camera sur elle-même (je tourne la tête)

Sélection d'un objet



1 / La sélection d'un objet se fait soit dans la 3D par un clic gauche ou dans la zone « arbre de scène ».

2 / Lorsqu'un objet est sélectionné dans la 3D, il est automatiquement sélectionné dans la zone « arbre de scène » à gauche et la caméra se positionne automatiquement en face de cet objet.

3 / A la sélection, un repère permet de visualiser le centre de cet objet

Ce repère est une représentation des axes suivant les directions X (rouge)

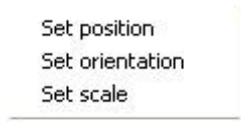
Y (vert) et Z (bleu)

Un clic droit sur un objet dans la zone d'édition 3D permet d'afficher le menu associé à cet élément.

Déplacement d'un objet

Il existe plusieurs manières de déplacer ou de transformer un objet ou un groupe d'objets dans la scène 3D.

- Ainsi, par un clic droit sur un objet sélectionné dans la zone « arbre de scène » on peut avoir accès aux fonctionnalités :



Pour chacune de ces transformations une boîte de dialogue s'ouvre et il est alors possible de saisir les valeurs pour transformer la position, l'orientation ou l'échelle d'un objet dans la 3D.

- Ces mêmes fonctionnalités seront accessibles par les boutons :

Position :  Orientation :  Scale : 

Lorsqu'un objet est sélectionné dans la 3D et que l'un des ces boutons est enclenché alors par l'action de la souris au clic gauche sur les axes du repère + déplacement de la souris provoquera une transformation sur l'objet.

Exemple : Je sélectionne mon objet, je clic gauche sur l'icône de positionnement, dans la 3D je clic sur l'axe X, puis je déplace ma souris en restant cliqué. Mon objet va alors bouger en suivant l'axe X.

Il en va de même sur les autres axes ainsi que sur les transformations en rotation et en échelle lorsque leurs boutons respectifs sont enclenchés.

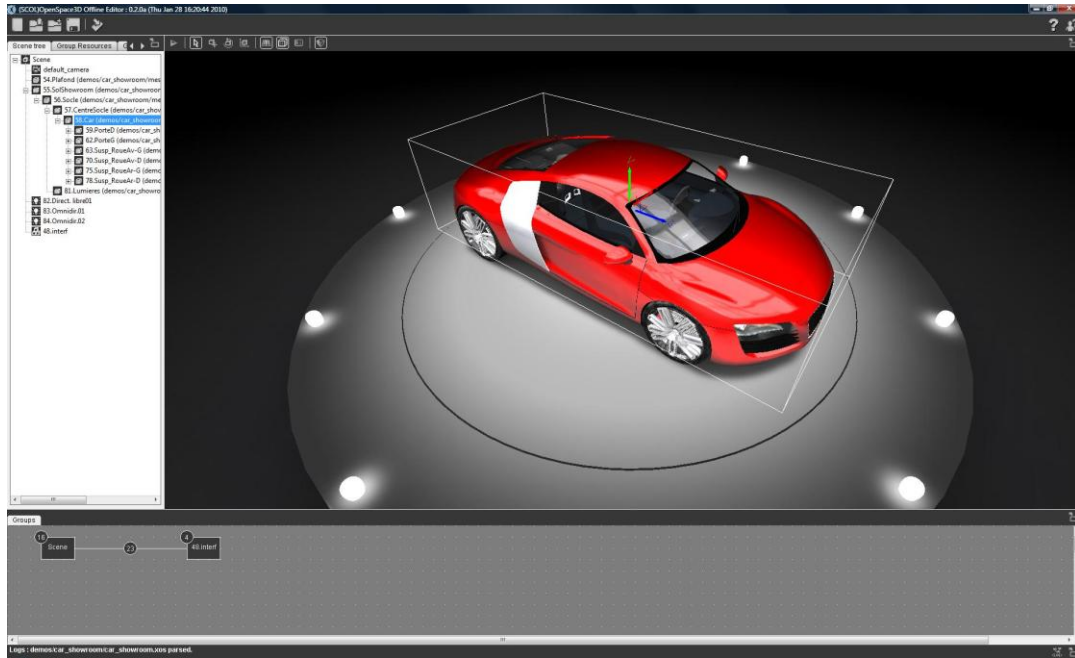
En mode « Scale » maintenez la touche « Alt » lorsque vous utilisez les axes pour redimensionner l'objet de façon homogène.

Enfin, Par un clic droit sur ces boutons, on retrouvera les mêmes boites de dialogue de transformations sur l'objet sélectionné.

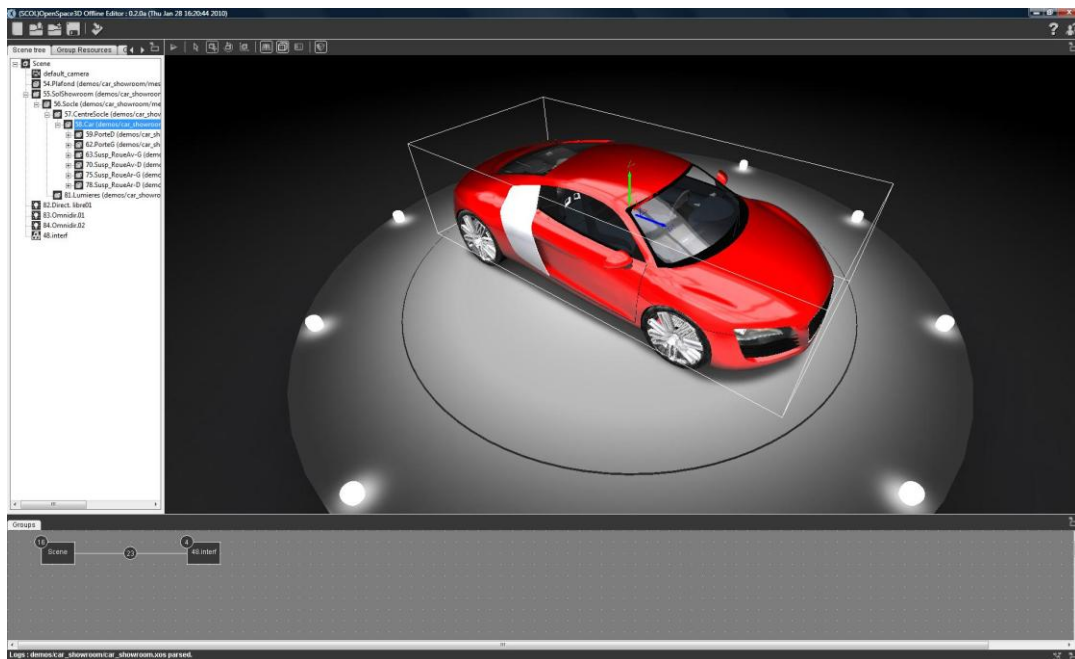
La duplication d'Objets

Dans la fenêtre 3D de l'éditeur, il est possible de dupliquer n'importe quel objet de la scène.

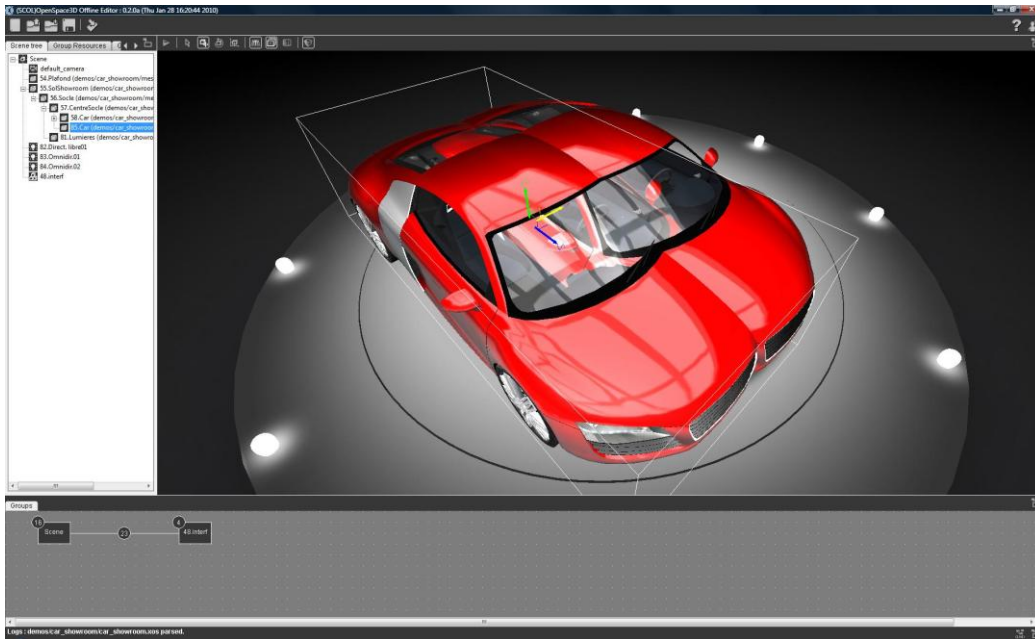
- sélection de l'objet :



- On choisit un mode de déplacement (translation, rotation, scale) :



- En maintenant la touche « shift » enfoncée et en déplaçant notre objet, on peut facilement le dupliquer :





Le mode Play/Pause



Ce bouton va permettre le lancement de la scène en mode application.

Le mode application est défini par les fonctions qui créent les différentes interactions dans la scène 3D.

Pour créer ces interactions on utilise « l'Editeur de Fonctions ».

N.B : Essayer les touches de raccourci CTRL+Back pour activer ou désactiver le mode play/pause

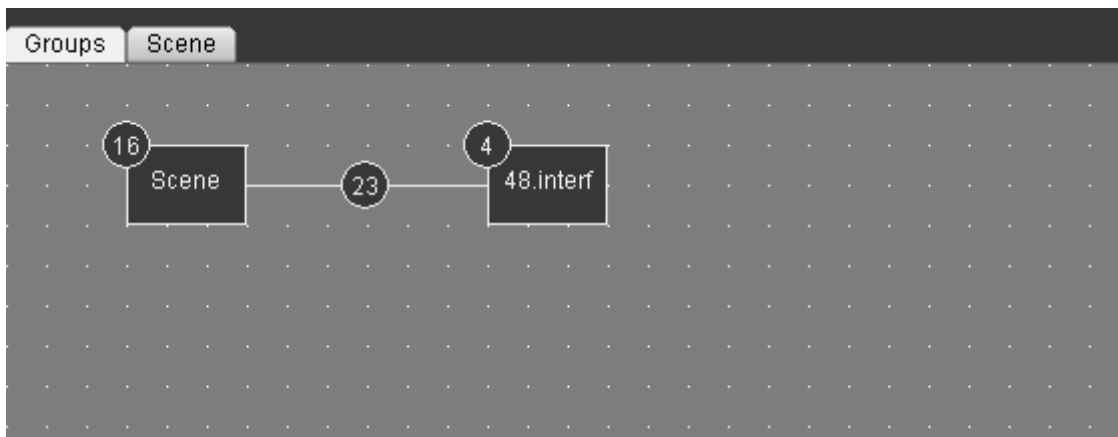
Edition des Fonctions (PlugIT)

Interface

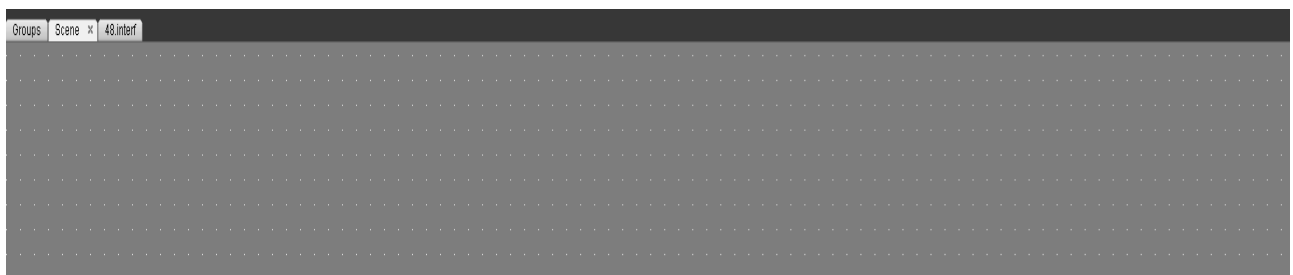
Interface d'Edition des Fonctions organise les ressources par groupes de ressources :

Deux groupes sont représentés :

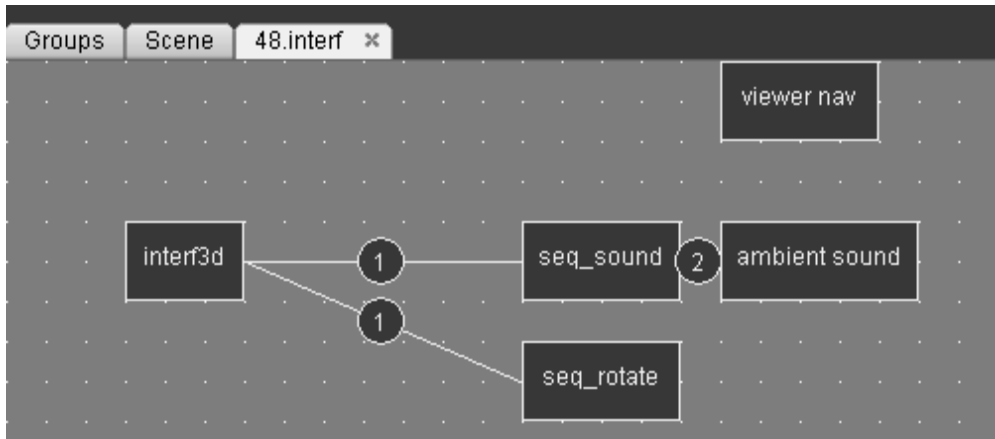
- Scene
- 48.interf



L'accès aux plugIT d'un groupe se fait en double cliquant sur leur représentation (le rectangle).

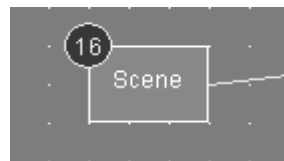


Lorsqu'ils sont ouverts dans l'éditeur de fonctions, les groupes de ressources sont accessibles par les onglets de navigation.

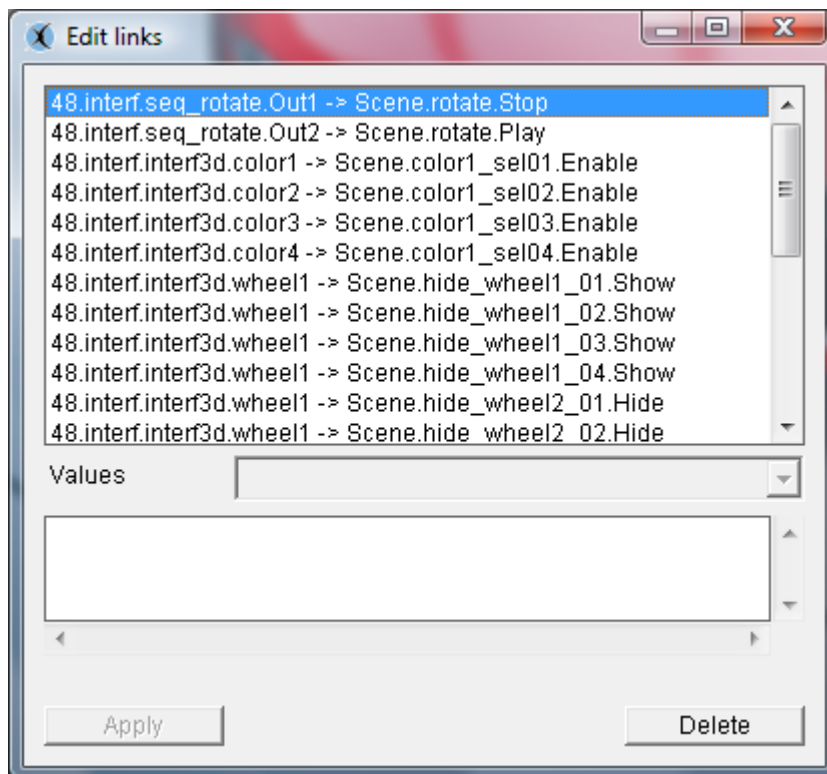


Les liens

Dans l'onglet « Groups » tous les groupes de ressources sont représentés et le nombre de liens (interactions) est affiché



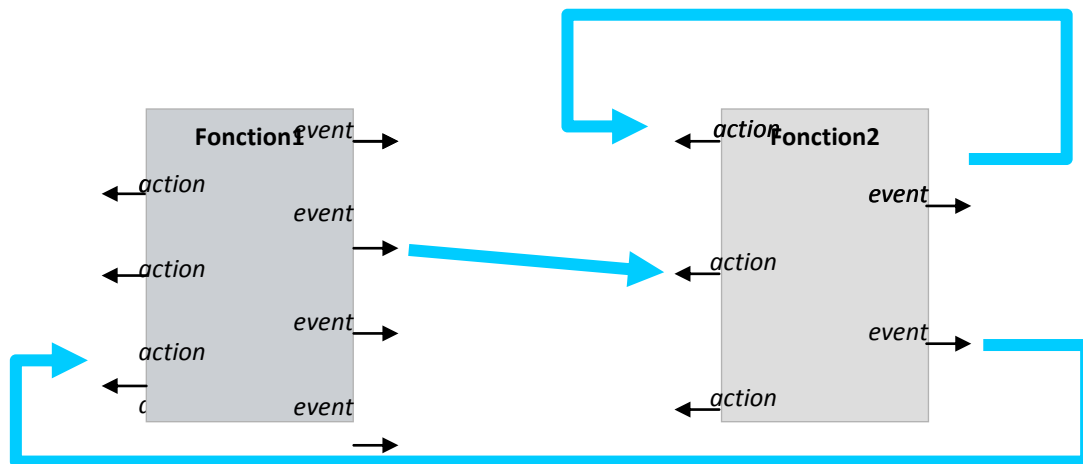
La liste des liens existants dans le groupe est accessible en double-cliquant sur le cercle contenant le nombre de liens.



Explications de la notion de « lien » :

Un **lien** est la connexion d'un événement (**event**) de fonction à une action (**action**) d'une autre ou de la même fonction.

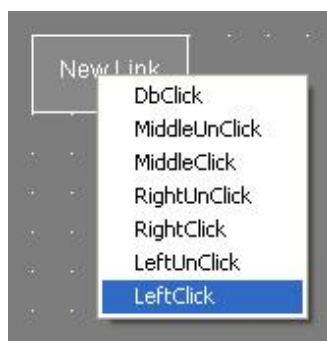
Certains liens proposent des valeurs prédéfinies que l'on peut sélectionner dans la liste des « values »



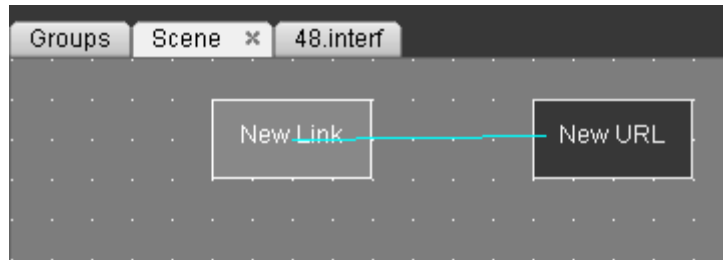
Dans la suite de ce support, la description de lien se fera de la manière suivante :

Group.Fonction.Event → Group.Fonction.Action

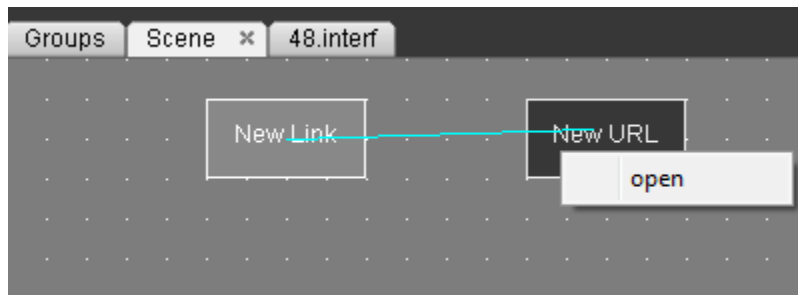
Pour créer des liens entre les fonctions, il faut effectuer un clic droit sur la fonction source puis sélectionner l'évènement voulu, ici « LeftClick ».



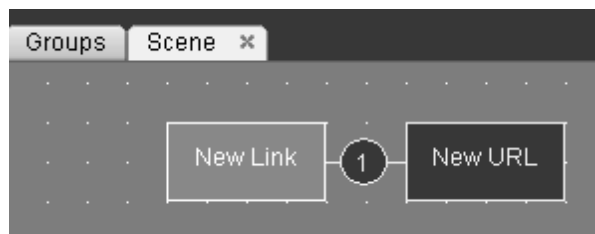
Déplacer le lien bleu depuis la fonction source jusqu'à la fonction de destination



Effectuer un clic droit sur la fonction de destination puis cliquer sur l'action voulue, ici « open »



Le lien est maintenant créé.



Lorsque vous dessinez un lien, vous pouvez annuler sa création en appuyant sur la touche « Echap ».

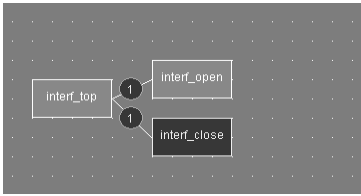
Certaines fonctions permettent de passer un paramètre dans le lien vers l'action déclenchée.

Par exemple une url pour sur l'action « play » de la fonction flash.

Pour supprimer un lien double cliquer sur une liste de lien pour les éditer un à un ou appuyer sur la touche « Suppr » après avoir sélectionné une liste de liens.

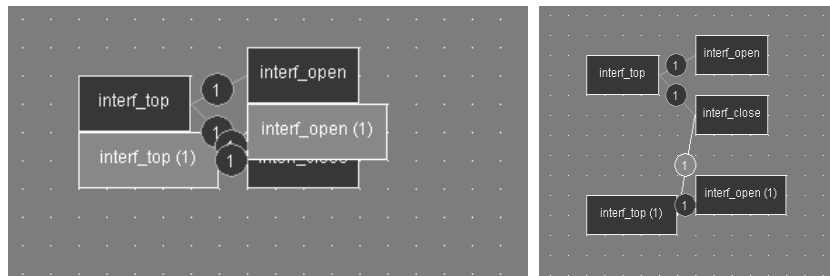
Pour supprimer une instance de fonction, sélectionner une instance puis appuyer sur touche « Suppr ».

Pour sélectionner plusieurs PlugITs effectuez un Ctrl + Clic sur les différents PlugITs à sélectionner



Pour dupliquer un groupe de PlugITs sélectionnez-les puis effectuez un clic droit Copier (Ctrl+C) puis un clic droit Coller (Ctrl+V) dans la zone d'édition ou Couper (Ctrl+X) puis Coller (Ctrl+V)

- plugITs
- Reload (F5)
- Copy (Ctrl + C)
- Cut (Ctrl + X)
- Paste (Ctrl + V)
- anr ▶
- input ▶
- interface ▶
- material ▶
- maths ▶
- media ▶
- misc ▶
- navigation ▶
- network ▶
- object ▶
- physics ▶
- rendering ▶



Vous pouvez alors supprimer les liens inutiles et éditer vos PlugITs

Vous pouvez également faire une multisélection de PlugITs en clic dans la zone de l'éditeur puis en maintenant le clic gauche enfoncé et en déplaçant votre souris



En survolant un lien ou un groupe de lien une info bulle vous décrit les liens entre PlugITet la manière dont ils ont été raccordés :

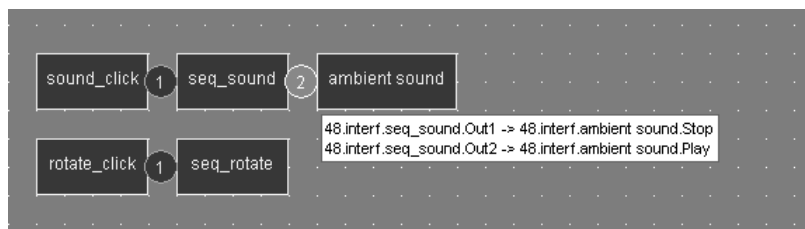
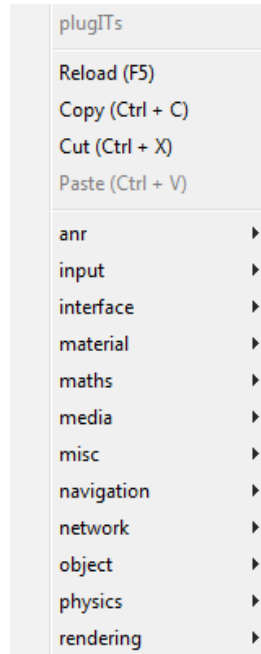


Tableau récapitulatifs d'ergonomie de l'éditeur de PlugITs

Ajouter un PlugIT	Clic droit dans la zone éditeur
Créer un lien	Clic droit sur le PlugIT
Sélectionner un PlugIT	Clic gauche sur le PlugIT
Sélectionner plusieurs PlugITs	Clic gauche puis Ctrl+Clic gauche pour sélectionner les autres
Sélectionner un lien ou groupe de lien	Clic gauche sur le chiffre de lien
Sélectionner un groupe de PlugITs et leurs liens par zone	Clic gauche dans l'éditeur puis sélection de la zone
Couper un ou plusieurs PlugITs et leurs liens	Clic droit "Couper" ou Ctrl+X
Copier un ou plusieurs PlugITs et leurs liens	Clic droit "Copier" ou Ctrl+C
Coller un ou plusieurs PlugITs et leurs liens	Clic droit "Coller" ou Ctrl+V
Dupliquer un ou plusieurs PlugITs	Ctrl + clic gauche + déplacement
Supprimer, un plugIT, un lien ou un groupe de PlugITs et de liens	Suppr
Rafraîchir les PlugITs et les réinitialiser	F5

Documentation des plugIT

Les plugIT sont classés par catégories, suivant le type de fonctionnalités qu'ils représentent :

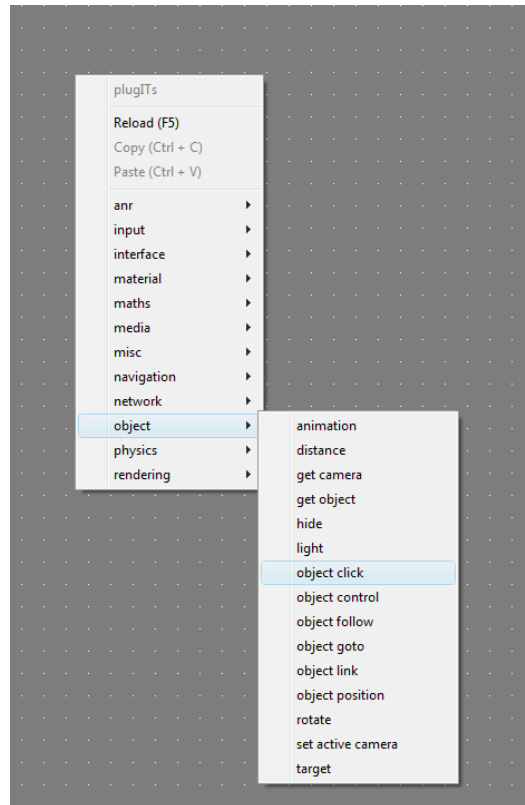


Exemple détaillé : le plugIT ObjectClick

Le PlugIT Object click permet de créer une action souris sur un objet



Pour ajouter cette fonction, il suffit d'effectuer un clic droit dans une zone vide, de sélectionner « object» puis de sélectionner « object click »



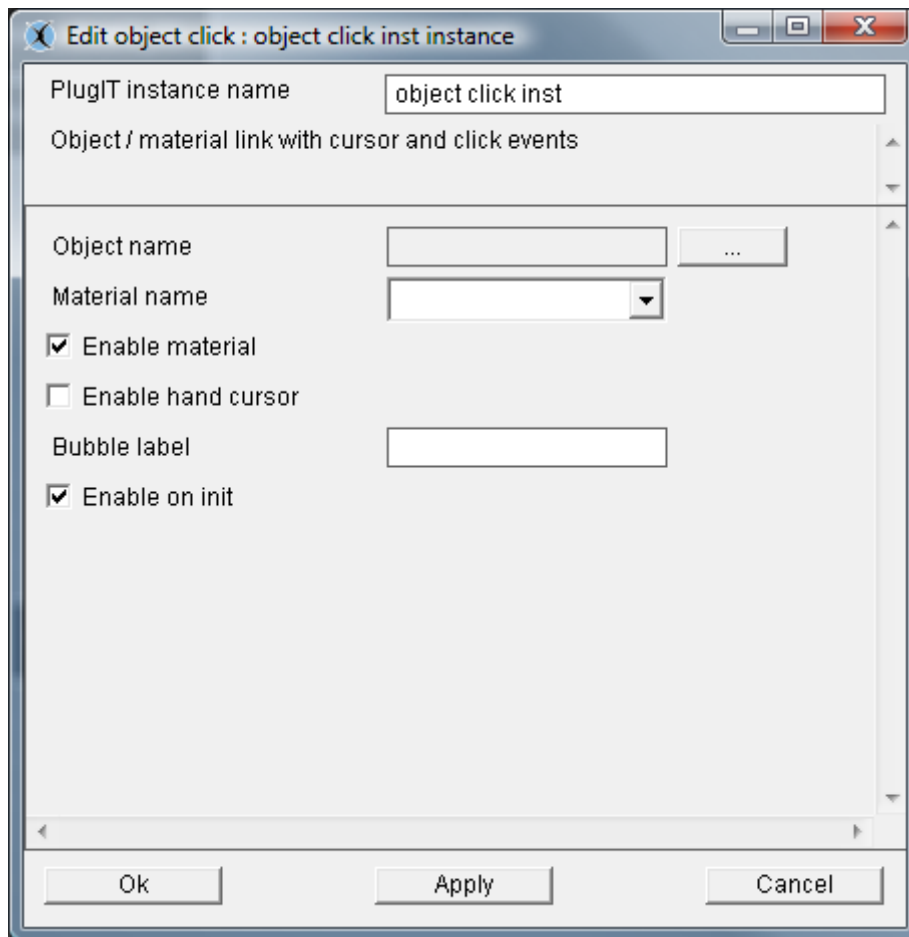
Il est conseillé de nommer précisément les instances afin de faciliter le travail de réédition.



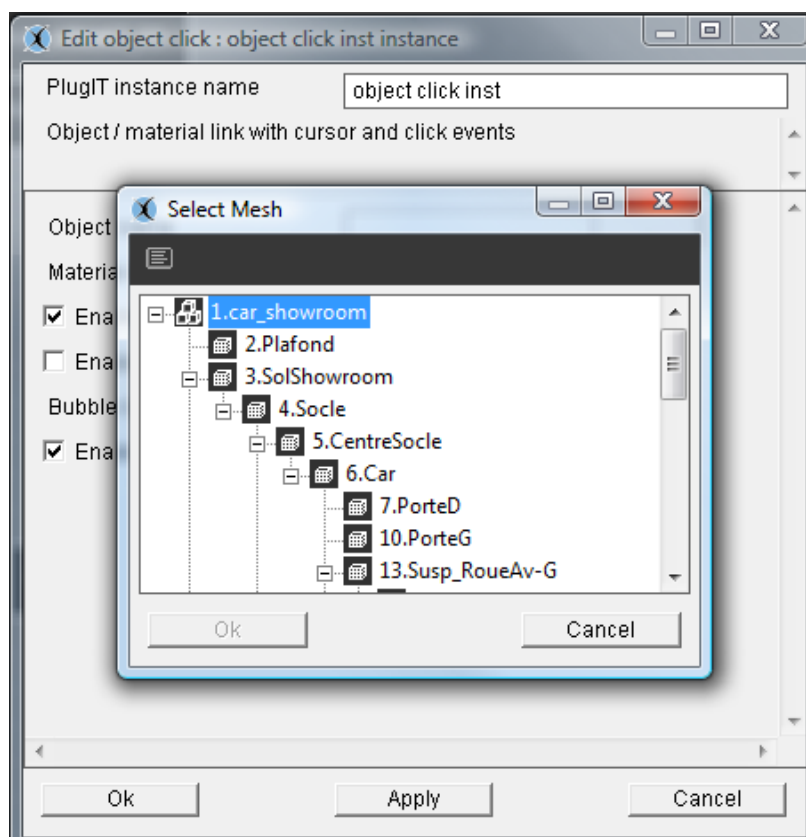
La nouvelle instance de fonction apparaît alors en haut à gauche de l'éditeur, pour la déplacer il suffit de la sélectionner au clic gauche et de déplacer la souris jusqu'à la zone voulue.



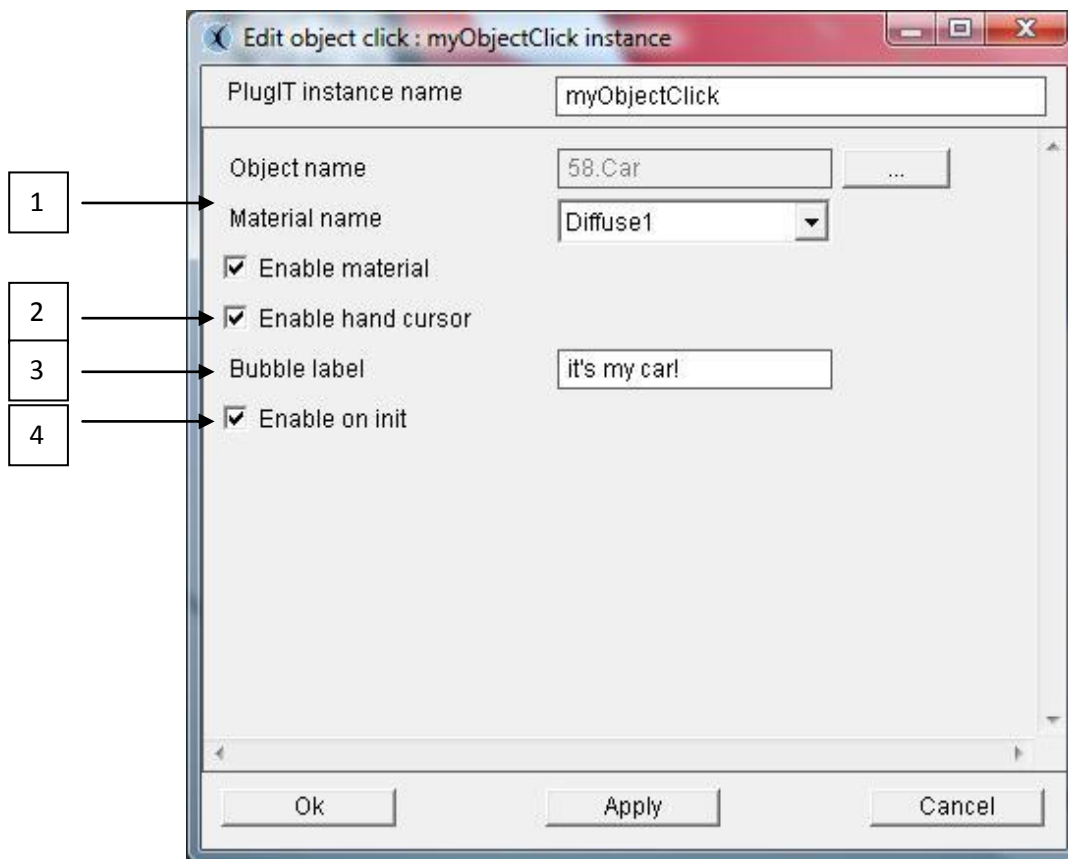
L'instance a correctement été ajoutée, il faut maintenant en éditer les paramètres, pour ce faire un double clic sur l'instance ouvrira la fenêtre d'édition des paramètres de l'instance.



Les paramètres sont vierges, ce qui signifie que pour l'instant, l'instance n'est appliquée à aucune ressource. Pour sélectionner une ressource (mesh/object name) du groupe, il faut cliquer sur l'icône « Parcourir », naviguer dans l'arbre et cliquer sur « ok » une fois que la ressource choisie est en surbrillance.



La ressource choisie est « 58.Car », ici un seul matériau est disponible, si l'on disposait d'un objet avec plusieurs matériaux, on pourrait choisir celui sur lequel on souhaite que la fonction s'applique depuis la zone « material name ».



1 °/Le PlugIT « Object Click » peut être appliqué sur un objet en totalité ou sur un matériau défini, dans ce dernier cas, il faut cocher la case « Enable material ».

2 °/Le paramètre « Enable hand cursor » permet de changer le curseur de la souris en main lorsque la souris passe sur l'objet ou le matériau.

3 °/Le paramètre « Bubble label » permet l'affichage d'une info bulle au survol de l'objet avec le texte rentré dans la zone correspondante : ici « it's my car ! »

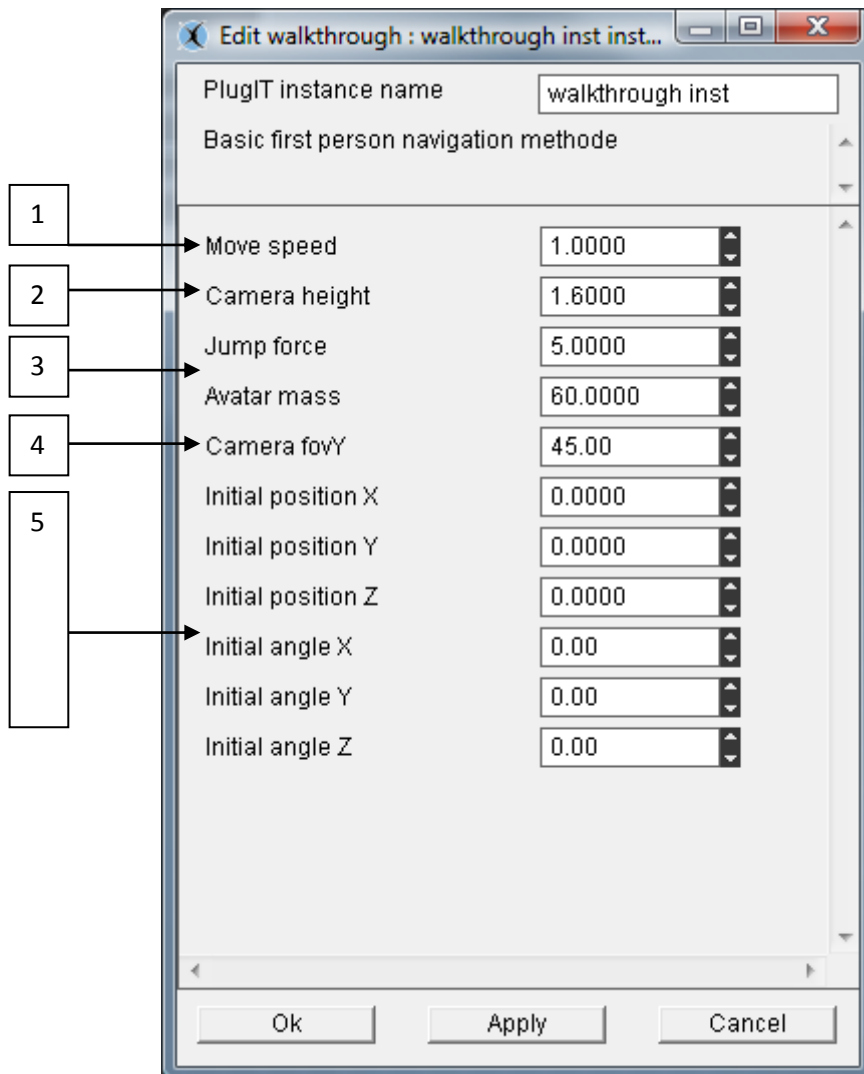
4 °/Le paramètre « Enable on init » permet de d'activer ce plugIT au lancement de l'application

Les plugIT « navigation »

PlugIT Walkthrough

Le PlugIT Walkthrough permet d'activer les déplacements de la caméra à la première personne.

De façon générale cette fonction devrait être ajoutée dans le groupe « Scene » puisque une seule instance suffit et qu'aucune ressource particulière n'est nécessaire. En suivant la même méthode que précédemment, ajouter un PlugIT Walkthrough, puis éditer l'instance.



1 °/Le paramètre « Move Speed » définit la vitesse de déplacement de la caméra.

2 °/ Le paramètre « Camera height» permet de définir la hauteur de la caméra.

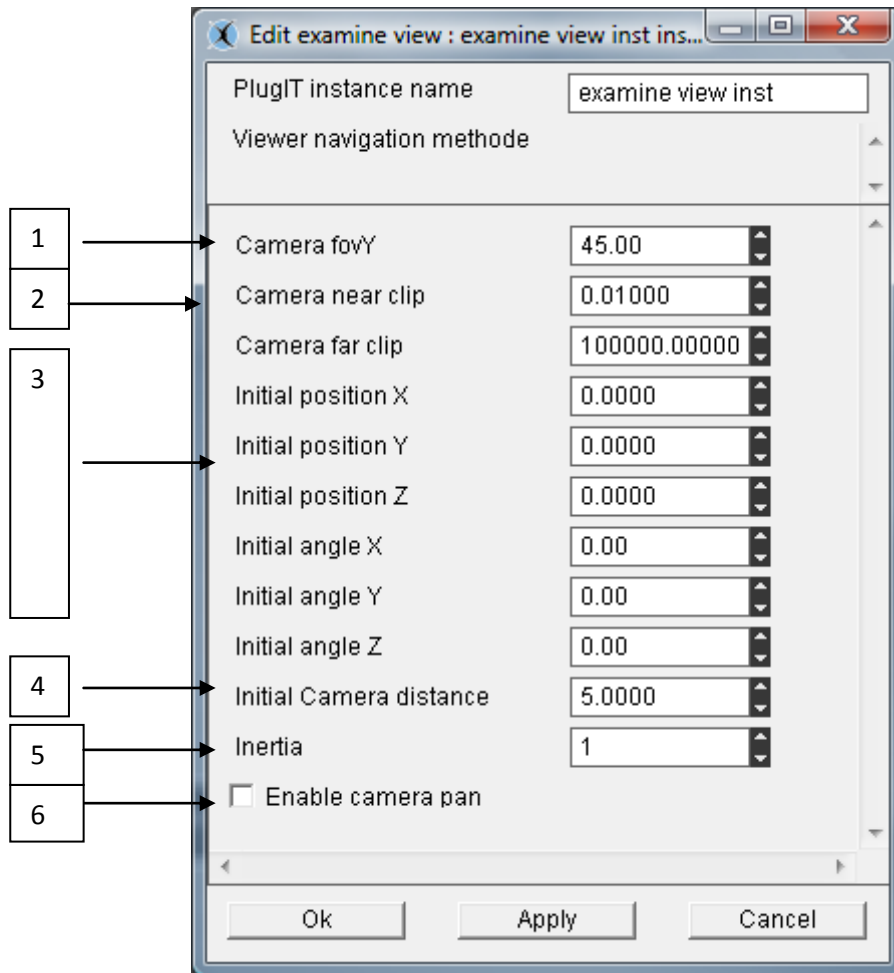
3 °/ Paramètres physiques dans le cas où la physique est activée (« jump » : force du saut (barre espace), « avatar mass » : masse du personnage navigant)

4 °/Le paramètre « Camera fovY» permet de définir l'angle de focale de la caméra.

5 °/ Les paramètres « Initial position » et « Initial angle » définissent la position et l'orientation d'entrée dans la scène en Y X Z.

PlugIT Examine View

Le PlugIT Examine View permet d'activer la navigation en utilisant la même ergonomie que lorsque l'on se trouve en mode édition. De façon générale cette fonction devrait être ajoutée dans le groupe « Scene » puisque une seule instance suffit et qu'aucune ressource particulière n'est nécessaire. En suivant la même méthode que précédemment, ajouter un PlugIT Examine View, puis éditer l'instance.



1 °/Le paramètre « Camera fovY » permet de définir l'angle de focale de la caméra.

2°/ Le paramètre « Camera clip » permet de définir la distance de vision de la caméra

3°/ Les paramètres « Initial position » définissent la position d'entrée dans la scène en Y X Z.

Les paramètres « Initial angle » définissent l'orientation d'entrée dans la scène en Y X Z.

4 °/Le paramètre « Initial Camera distance » définit la distance par défaut de la caméra par rapport à la position initiale.

5 °/Le paramètre « Inertie » définit la valeur d'inertie lorsque la physique est activée

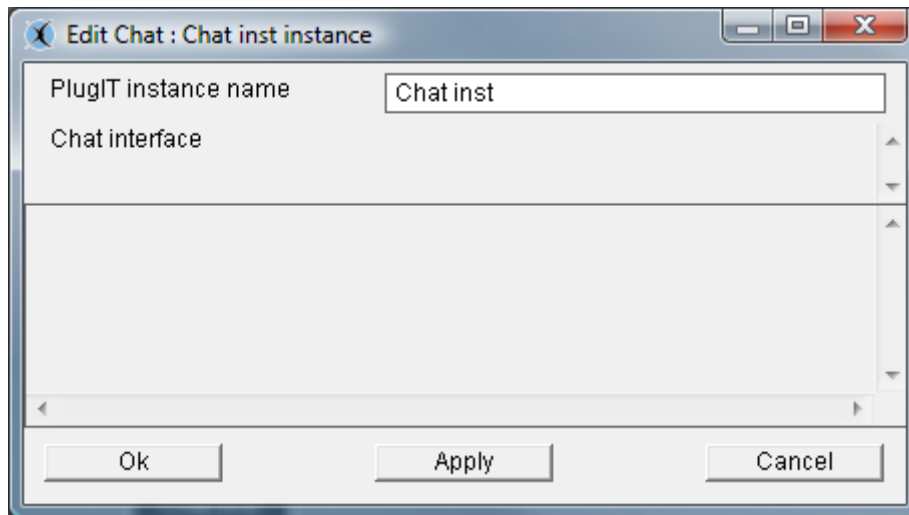
6°/Le paramètre « Enable camera pan » active ou désactive la translation de la caméra

Les plugIT « Network »

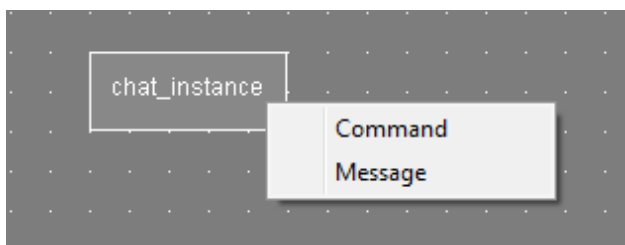
PlugIT chat

(Voir exemple fourni avec l'installation d'OpenSpace3D : chat)

Ce plugIT permet l'ajout d'une interface gérant un chat en flash.

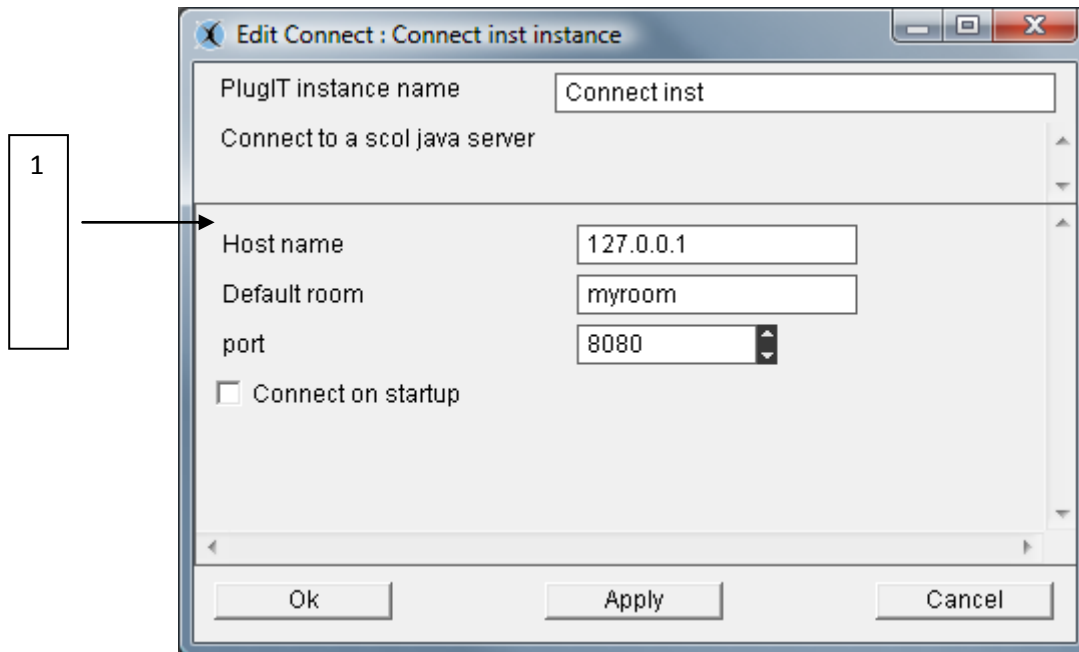


Les événements associés correspondent aux messages reçus par le flash ou les commandes envoyées à celui-ci :



PlugIT connect

Ce plugIT gère les connexions à une chat room du server java pour OpenSpace3D.



1 °/ « host name » : adresse du serveur associé

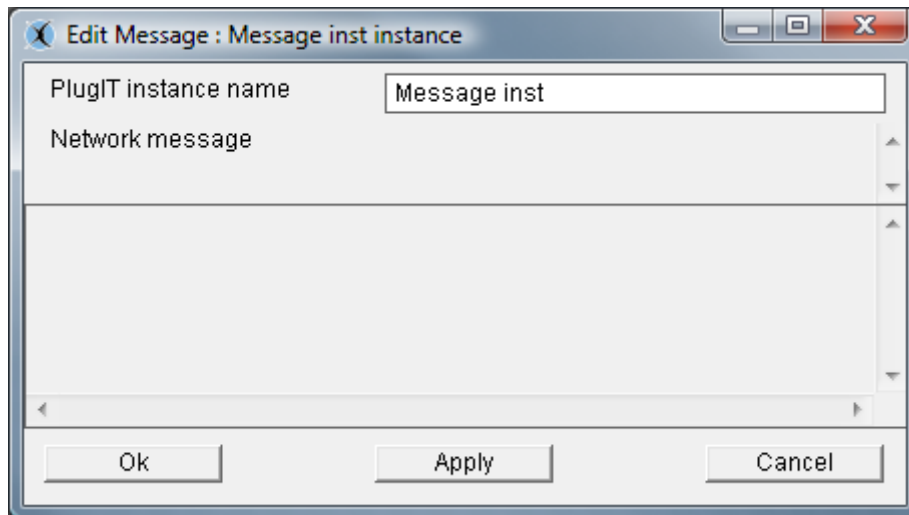
« Default room » : Nom de la chat room sur laquelle on cherche à se connecter

« Port » : le port associé

« Connect on Start up » : connexion automatique au lancement de l'application

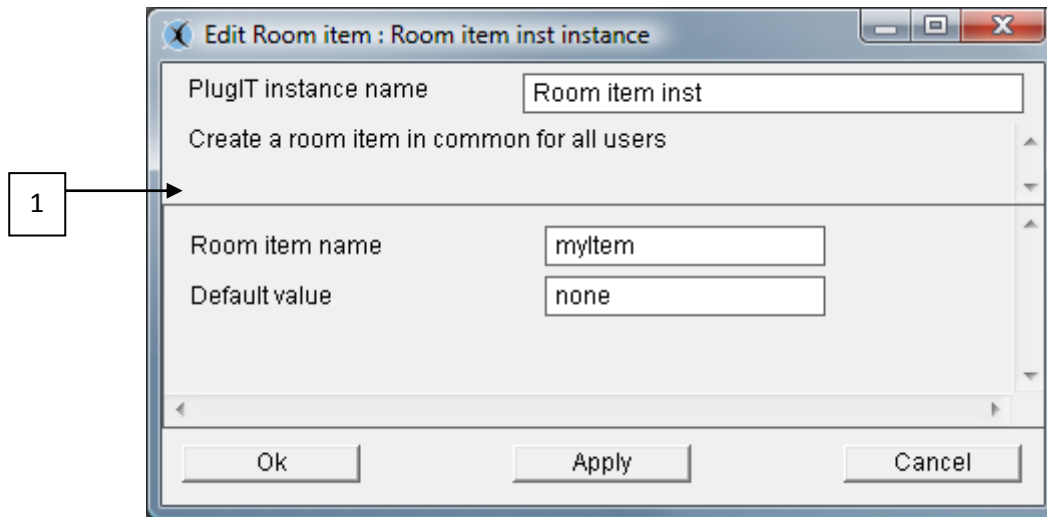
PlugIT message

Ce plugIT permet au client (application) d'envoyer un message réseau.



PlugIT room item

Ce plugIT permet de créer un paramètre d'item commun.

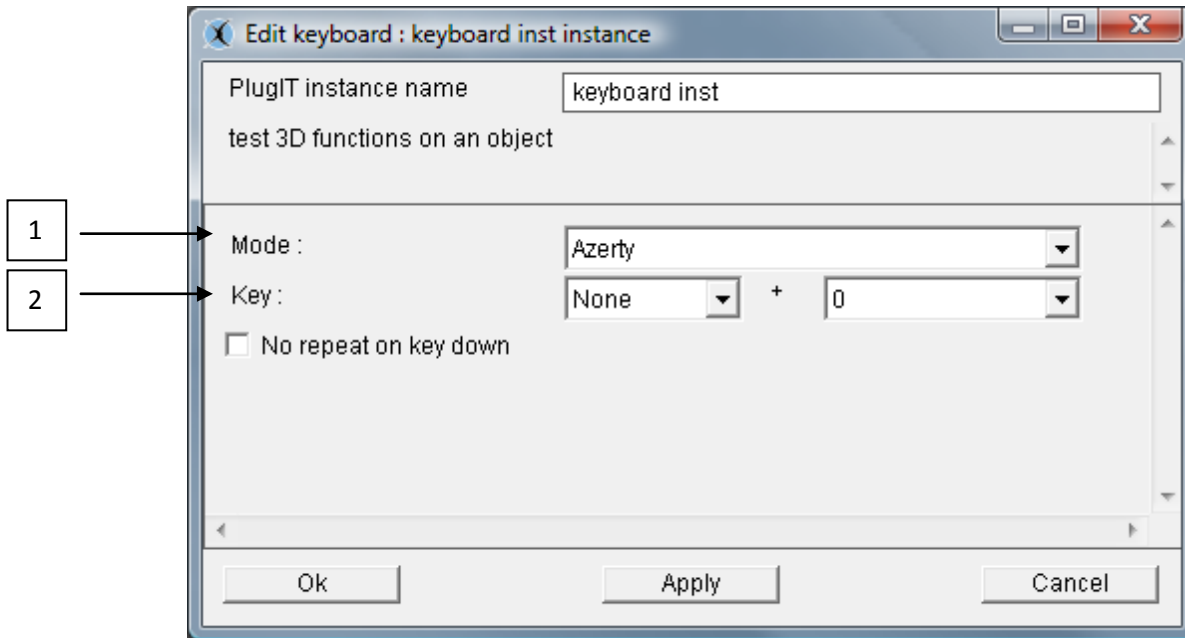


1 °/ On renseigne le nom de l'item de la room ainsi que la valeur par défaut de cet item

Les plugIT « Input »

PlugIT Keyboard

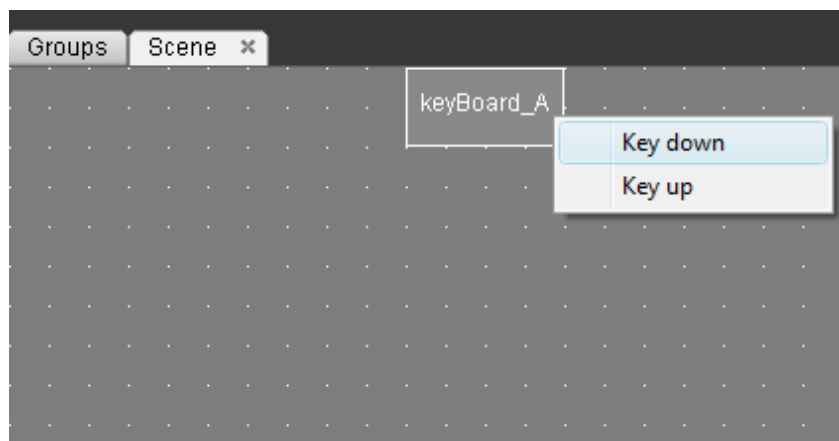
Le PlugIT Keyboard permet de paramétrer une touche de clavier en événement afin de provoquer une action sur l'application.



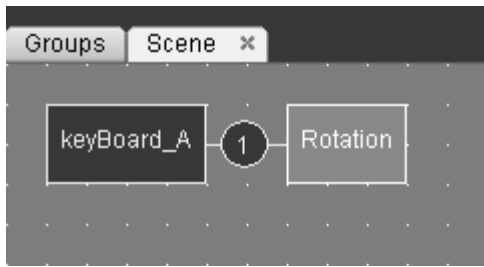
1 °/ Permet de choisir le type de clavier en input (AZERTY ou QWERTY)

2 °/ Le second paramètre correspond à l'appui sur la touche désirée. Le premier paramètre permet de paramétrer l'état courant des touches (CTRL, SHIFT, CTRL+SHIFT ou NONE)

Exemple :



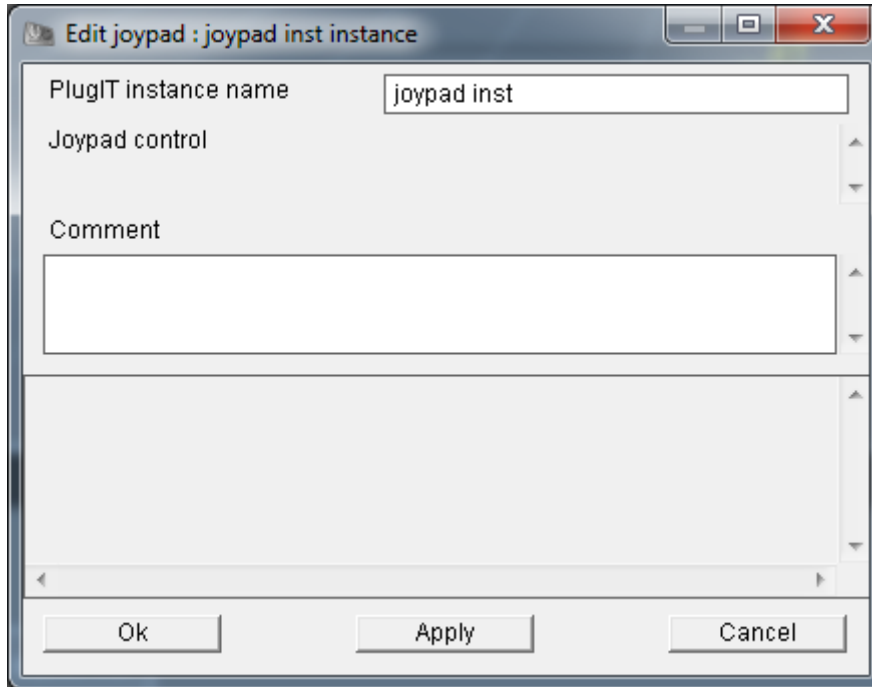
En événements, on retrouve notre clavier ainsi paramétré, ce qui nous permet de créer une action par un nouveau lien sur un autre plugIT en fonction de l'état de la touche.



Ici, on lance l'action sur le plugIT rotate (rotation d'un objet) lorsque la touche A est pressée.

PlugIT Joypad

Le PlugIT joypad vous permet d'utiliser les contrôles d'un joypad pour interagir avec l'application 3D.



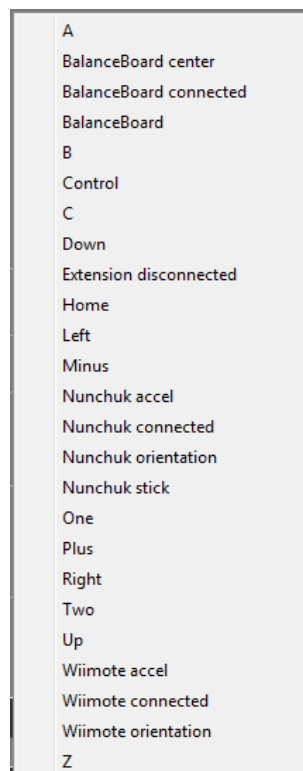
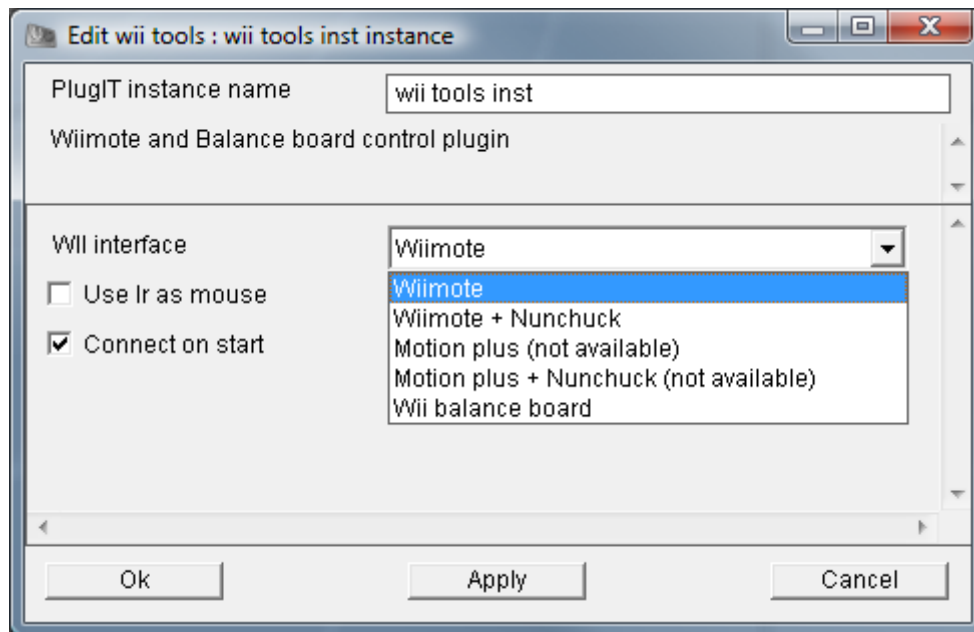
Ce PlugIT ne contient pas de paramètres particuliers, c'est en le reliant avec un autre plugIT que vous pourrez sélectionner le contrôle à utiliser en entrée et le raccorder à une action en sortie.



PlugIT Wiimote

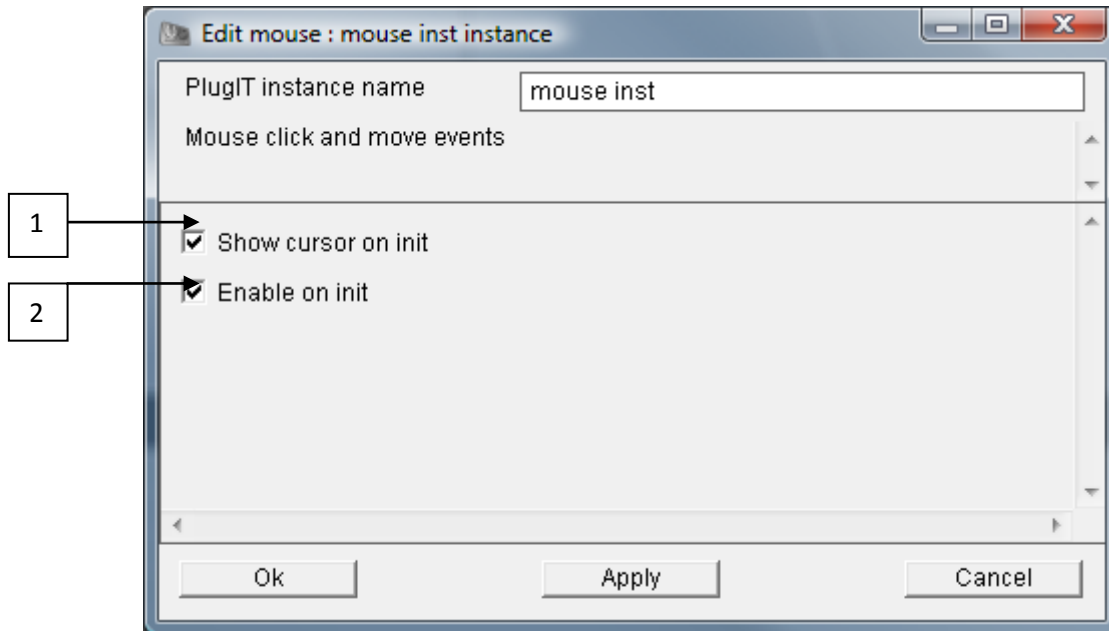
Wii Tools

Le PlugIT Wii Tools vous permet de paramétrer des interactions qui vous permettront ensuite d'utiliser la wiimote ou tout autre accessoires Wii (Nunchuck, balance Board...) comme interface input avec votre application :



PlugIT Mouse

Le PlugIT mouse vous permet de définir les paramètres de contrôle de la souris comme afficher ou cacher le curseur, activer ou désactiver le control de la souris dans l'application.



1° / On définit ici si le curseur est visible au démarrage

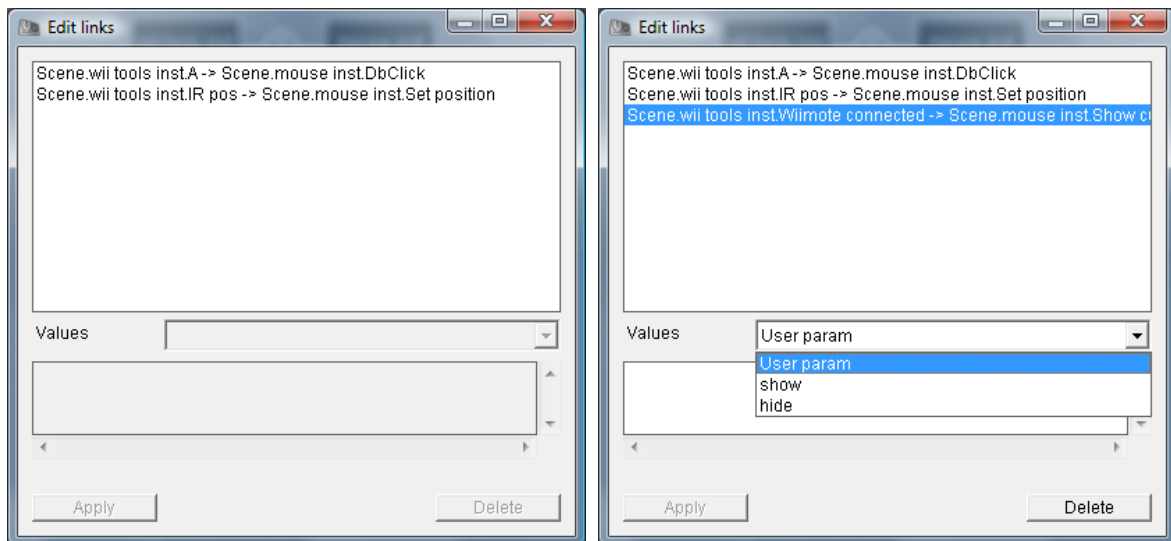
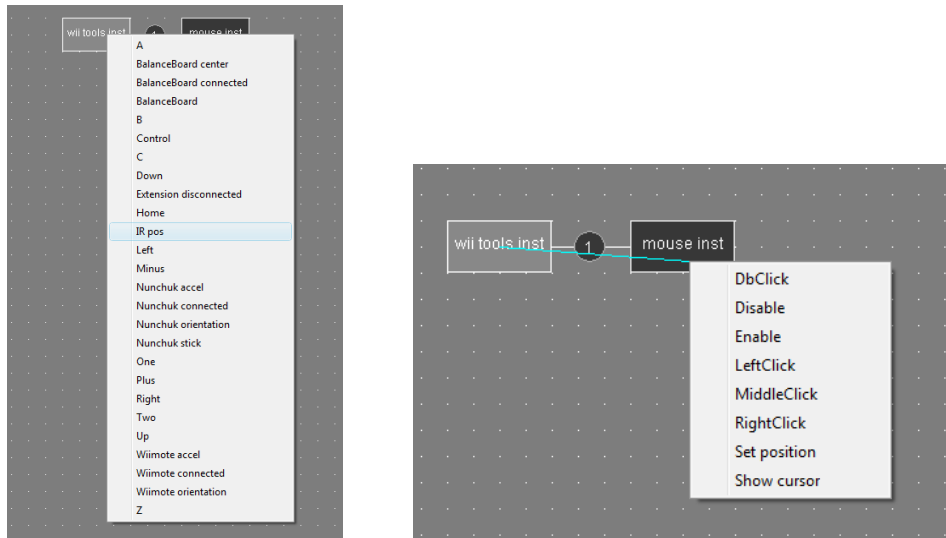
2° / On définit ici si la souris est active au démarrage

Liste des contrôles disponibles en sortie du plugIT Mouse :

Double-clic gauche	→	DbClick
Clic gauche	→	LeftClick
Clic milieu (mollette)	→	MiddleClick
Déplacement de la souris	→	MouseMove
Clic droit	→	RightClick

Exemple d'utilisation du plugIT mouse avec le plugIT Wii Tools:

On définit ici la Wiimote comme étant l'interface homme-machine de l'application avec un comportement de souris. Pour chaque contrôle de la wiimote, on crée un lien depuis le plugIT Wii Tools vers le plugIT mouse par exemple la touche A correspond au double clic d'une souris et la position de l'Infrarouge (IR pos) correspond à la position du curseur d'une souris.

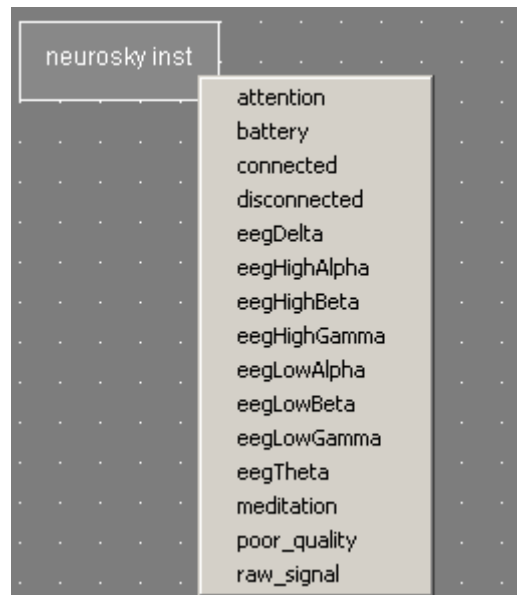


Pour l'affichage du curseur (lorsque la wiimote est connecté) on définit si qu'on affiche celui-ci en éditant les paramètres du lien :

PlugIT NeuroSky

Le neurosky est une BCI (Brain computer interface), elle peut-être utilisée dans n'importe laquelle des applications OpenSpace3D.

Plus d'informations : <http://www.neurosky.com/>



Voici les événements capables de renvoyer le plugIT :

Attention : Concentration du porteur de l'IHM (0 à 100)

Meditation : Méditation du porteur de l'IHM (0 à 100)

Battery : Niveau de batterie du neurosky

Connected : Renvoie un événement lorsque le neurosky est connecté

Disconnected : Renvoie un événement lorsque le neurosky est déconnecté

Les EEG : http://fr.wikipedia.org/wiki/Rythme_cérébral

EegDelta : Fréquence delta du cerveau

EegHighAlpha : Fréquence high alpha du cerveau

EegHighBeta : Fréquence high bêta du cerveau

EegHighGamma : Fréquence high gamma du cerveau

EegLowAlpha: Fréquence low alpha du cerveau

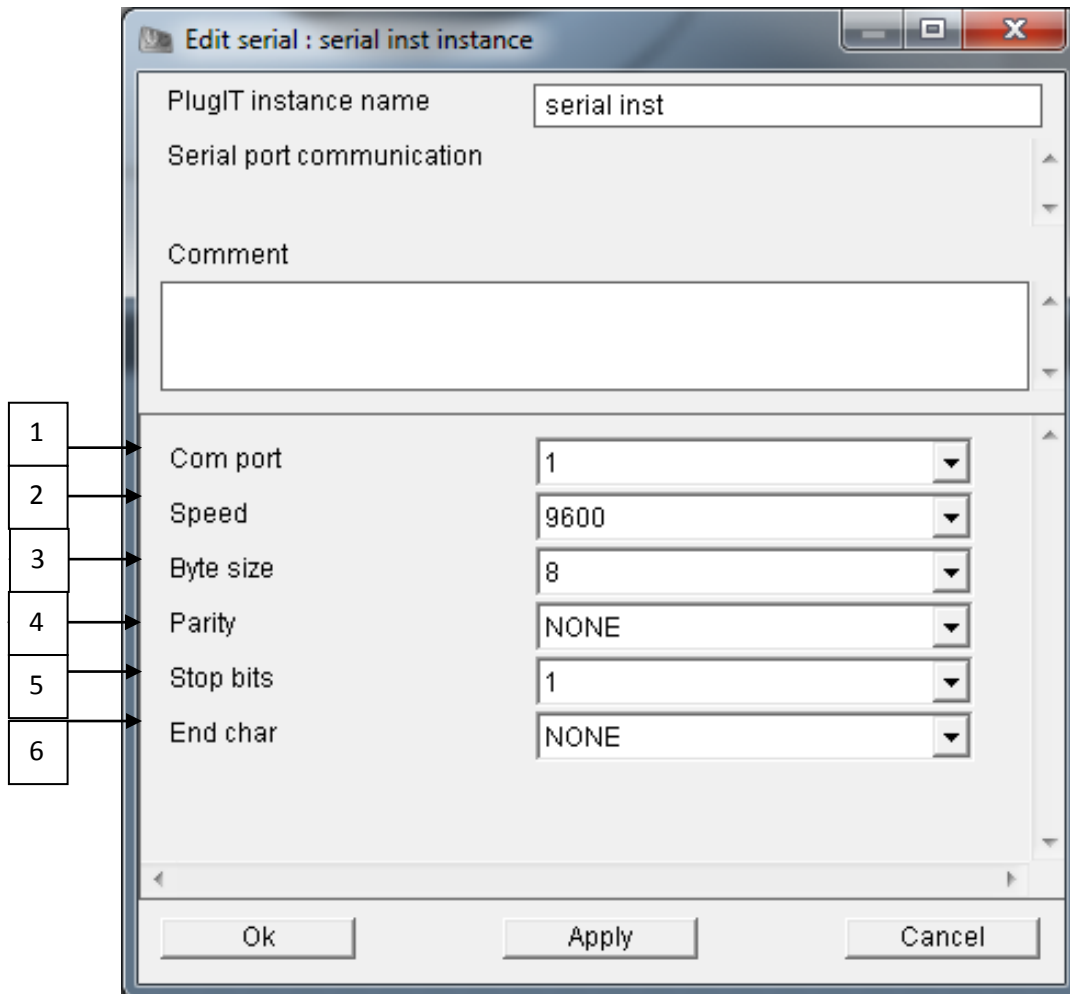
EegLowBeta : Fréquence low beta du cerveau

EegTheta : Fréquence Thêta du cerveau

Poor_quality : (0 à 200) Rapporte la qualité du signal, si la valeur est 0 le casque est bien positionné si la valeur est de 200 alors le casque ne reçoit aucun signal

Raw_signal : Signal brut issu du casque

PlugIT Serial



Le PlugIT Serial vous permet de faire communiquer votre application OpenSpace3D avec un port de communication série

1° / Numéro du port serie sur lequel vous branchez votre appareil

2° / Vitesse de transfert des données en baud

3° / Taille des paquets en Byte

4° / contrôle de parité

5° / Bits d'arrêt

6° / Caractère de fin de message



Les plugIT « Interface »

PlugIT flash Interface

Ce plugIT très complet et puissant, permet l'ajout d'interface flash dans l'application. Cette fonctionnalité peut-être appliquée sur un objet 3D ou encore en interface 2D classique.

1 °/ Si la case, « apply on texture » est cochée, cela signifie que l'on souhaite appliqué le plugIT sur un objet 3D, il est donc nécessaire de renseigner les champs « Object name » et « material name »

2 °/ On choisit ici si l'on souhaite chargé le control flash via une url (« url ») ou en local (« flash control path »)

3 °/ Ces paramètres permettent de positionner et de dimensionner le flash contrôle.

4 °/ Le « z Order » correspond à l'ordre d'affichage en profondeur du flash contrôle. Il ne sera pris en compte que dans le cas ou le flash contrôle est une interface 2D.

5 °/ Ces paramètres permettent de configurer des options de comportements sur notre interface.

« Transparency » : Permet d'afficher la transparence du flash contrôle

« KeyBoard on create » : Donne le focus clavier à la création du flash contrôle (possibilité d'utiliser le clavier directement au lancement)

« Draggable » : Permet le déplacement du flash contrôle par le clic droit de la souris pendant le déroulement de l'application

« Top on Focus » : Permet au flash control de passer par-dessus les autres lorsque l'on clic dessus.

« Always on Top » : Permet au flash control de toujours être au dessus lors des manipulations de l'ordre d'affichage par le clic souris.

« Create on start » : créer le flash control dès le lancement de l'application

« Enable Keyboard » permet d'autoriser le control clavier sur l'interface flash

« Enable Mouse » permet d'autoriser le control souris sur l'interface flash

6 °/ Cette partie permet la communication avec le flash contrôle.

Les « flash callback » permettent de récupérer des événements issus du flash vers un autre plugIT

Les « flash Call » permettent d'envoyer à flash des paramètres.



- 1 →
- 2 →
- 3 →
- 4 →
- 5 →
- 6 →

Edit Flash Interface : Flash Interface inst instance

PlugIT instance name:

Flash interface interface:

Material setting

Apply on texture

Object name:

Material name:

Material technique:

Technique pass:

Pass texture:

File setting

Url:

Flash control path:

Position / Size setting

Scale mode:

Pos X: (%)

Pos Y: (%)

Width: (%)

Height: (%)

Misc setting

Opacity:

Initial Z order:

Transparency

Keyboard focus on create

Draggable

Top on get focus

Always on top

Enable KeyBoard

Enable Mouse

Create on start

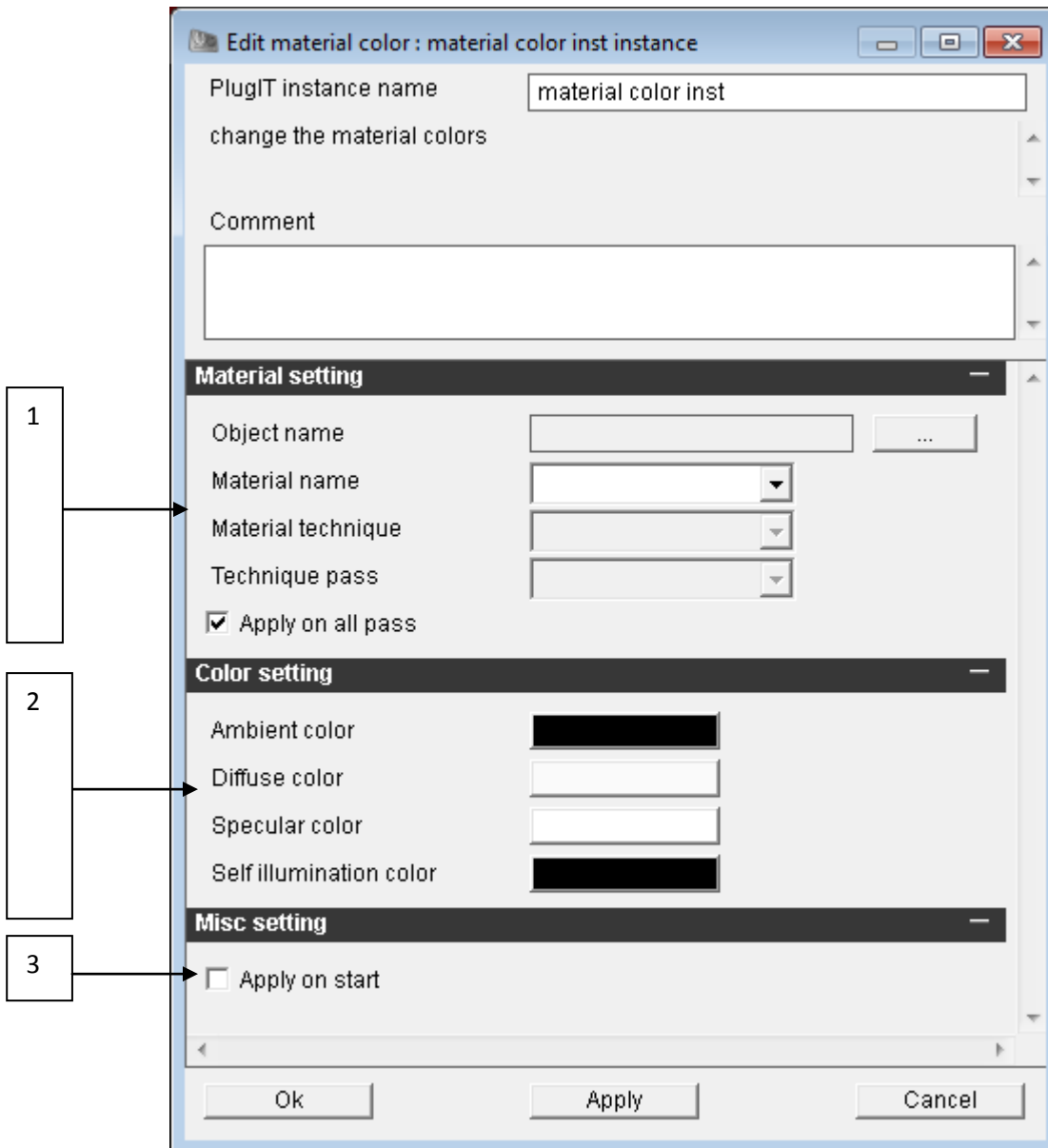
Flash communication

Flash callback	Flash call
<input style="width: 95%;" type="text"/>	<input style="width: 95%;" type="text"/>
<input type="text"/> + <input type="text"/>	<input type="text"/> + <input type="text"/>

Les plugIT « Material »

PlugIT Material color

Le PlugIT Material color permet de modifier dynamiquement les couleurs sur toutes les passes d'un material.



1 °/Cliquer sur le bouton parcourir « ... » pour sélectionner l'objet puis le matériau qui sera modifié.

Il est également possible de choisir la couleur sur une passe en particulier (L'application sur toutes les passes est activée par défaut)

2 °/Ambient color est la couleur ambiante du matériau qui sera appliquée lorsque le PlugIT est activé.

Diffuse color est la couleur diffuse du matériau qui sera appliquée lorsque le PlugIT est activé.

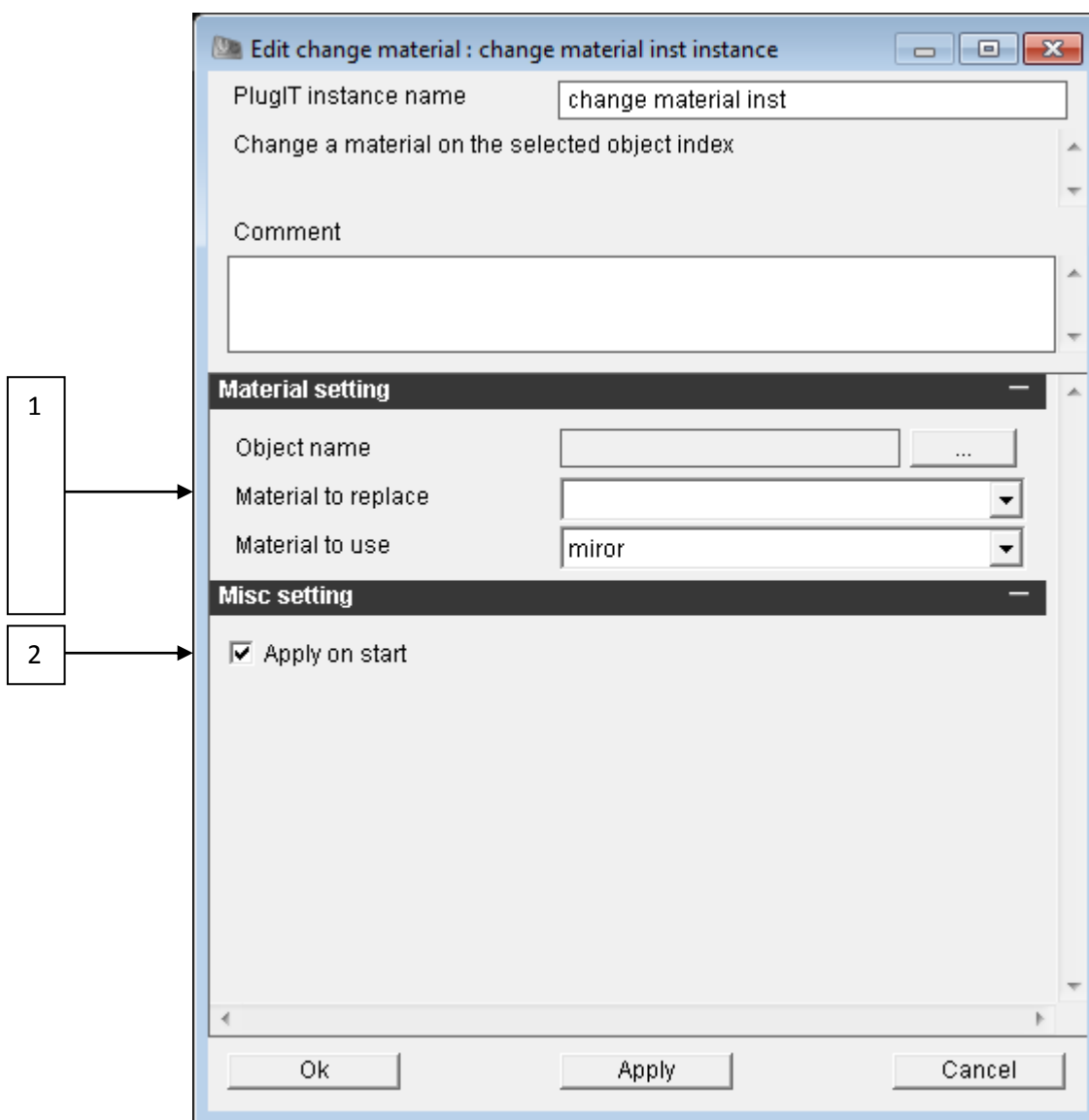
Specular color est la couleur spéculaire du matériau qui sera appliquée lorsque le PlugIT est activé.

Self illumination color est la couleur d'auto illumination du matériau qui sera appliquée lorsque le PlugIT est activé.

3 °/Activation du changement de couleur au démarrage de l'application ou si décoché à l'activation d'une action.

PlugIT change Material

Ce plugIT permet de changer un material par un autre.

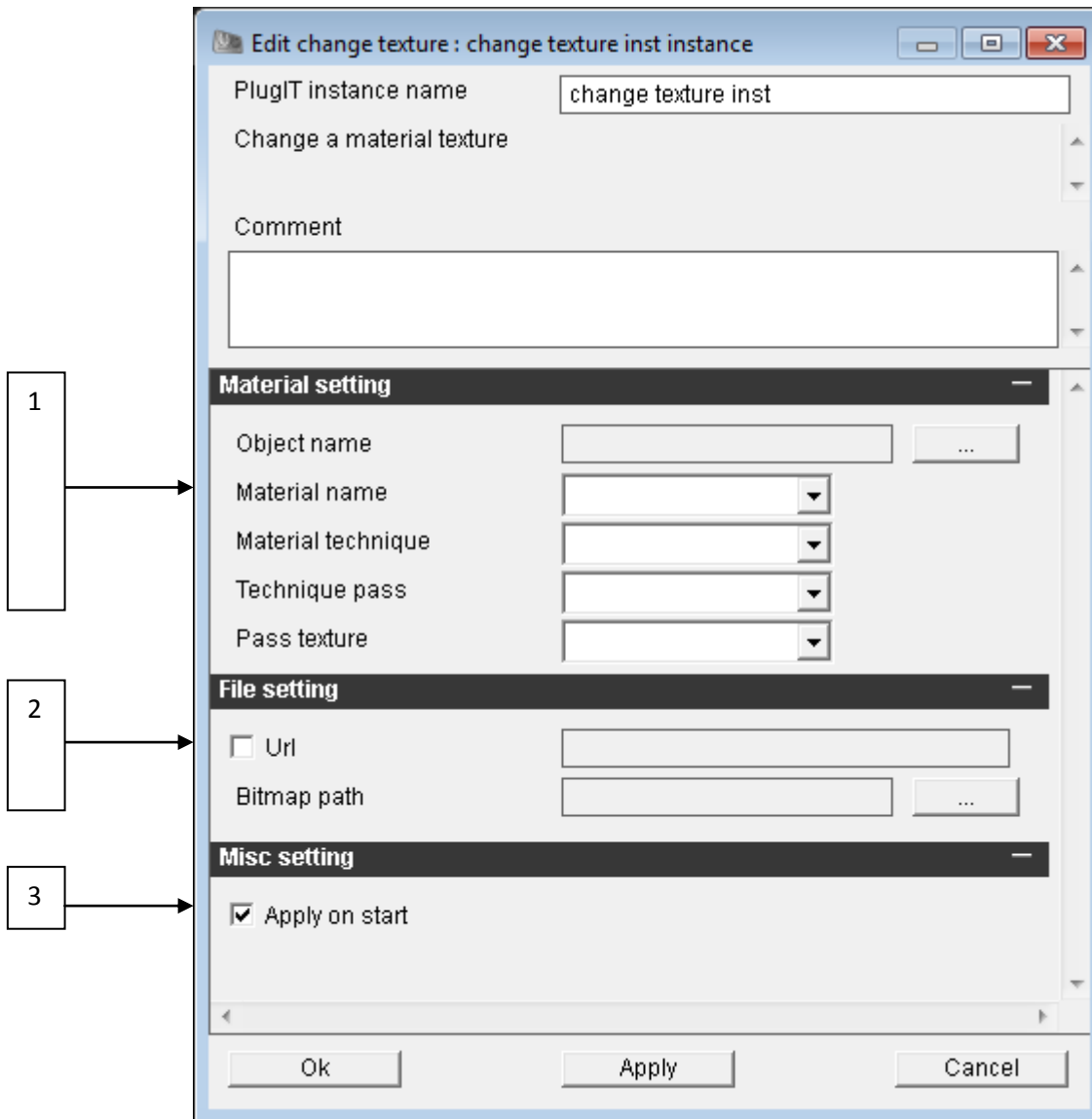


1 ° / On renseigne ici le nom de l'objet 3D ainsi que le material à remplacer et celui à utiliser

2 ° / On choisit ici si l'on souhaite l'activation de cette modification au démarrage.

PlugIT change texture

Ce plugIT permet de changer une texture sur un material



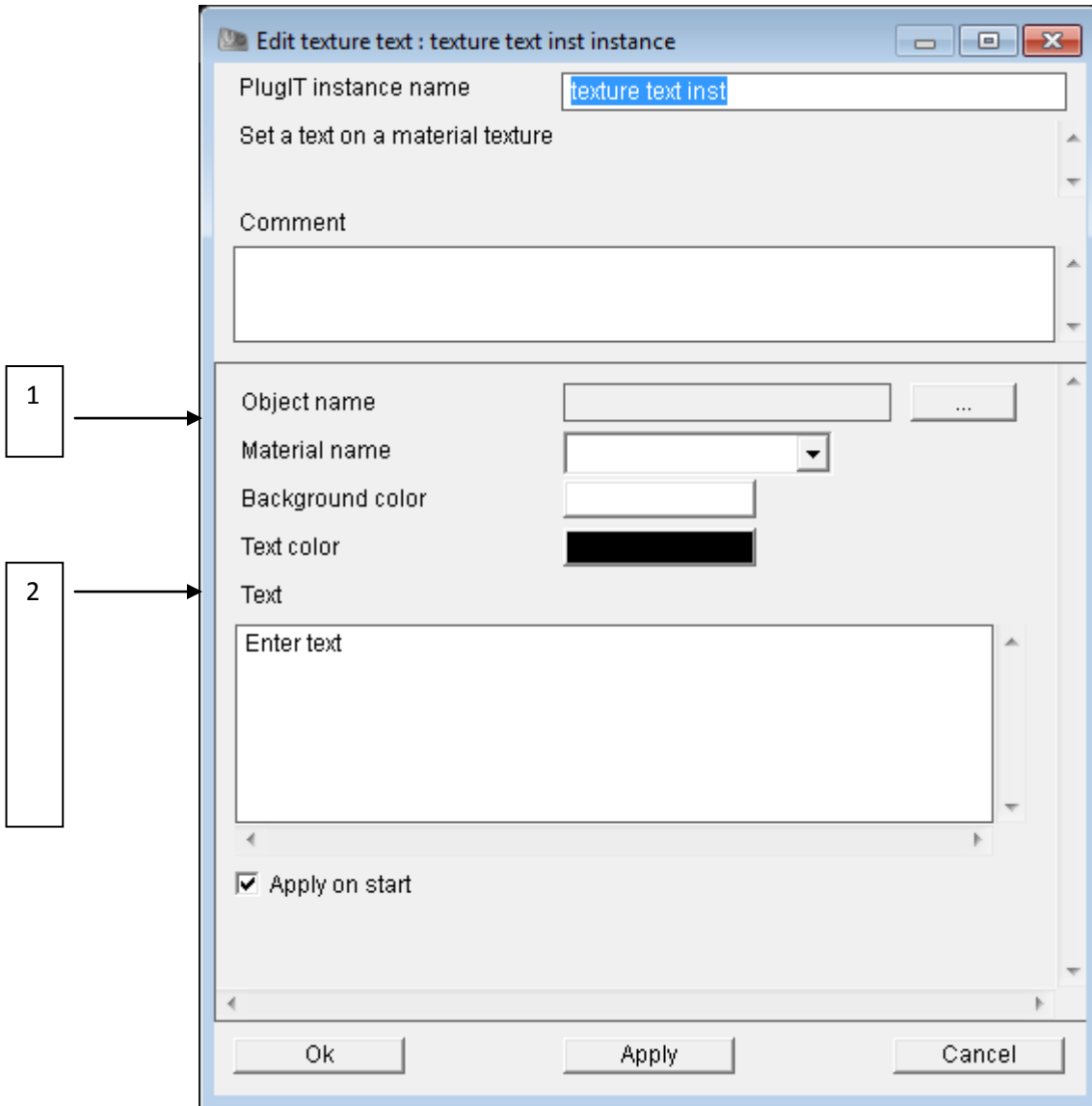
1 ° / On renseigne ici le nom de l'objet 3D ainsi que le material à remplacer et celui à utiliser. On renseigne également sur quelle technique, quelle passe et quelle texture du material, on souhaite appliquer la nouvelle texture

2 ° / On choisit ici si l'on souhaite appliquer une image issue d'une URL ou appliquer une image présente sur votre disque

3 ° / On choisit ici si l'on souhaite l'activation de cette modification au démarrage.

PlugIT texture Text

Ce plugIT permet d'écrire du texte en dynamique sur une texture



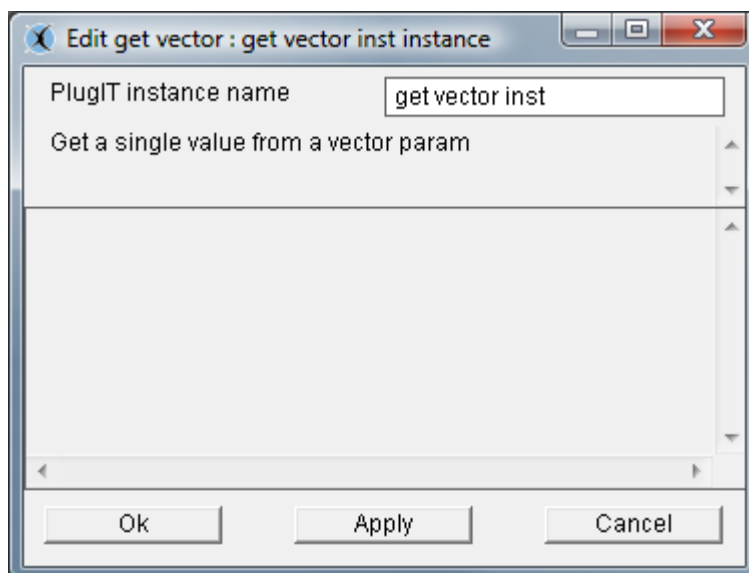
1 °/ On renseigne ici le nom de l'objet 3D ainsi que le matériau sur lequel le texte sera appliqué

2 °/ On renseigne simplement la couleur de fond du texte, la couleur de la police et le texte que l'on souhaite intégrer sur la texture.

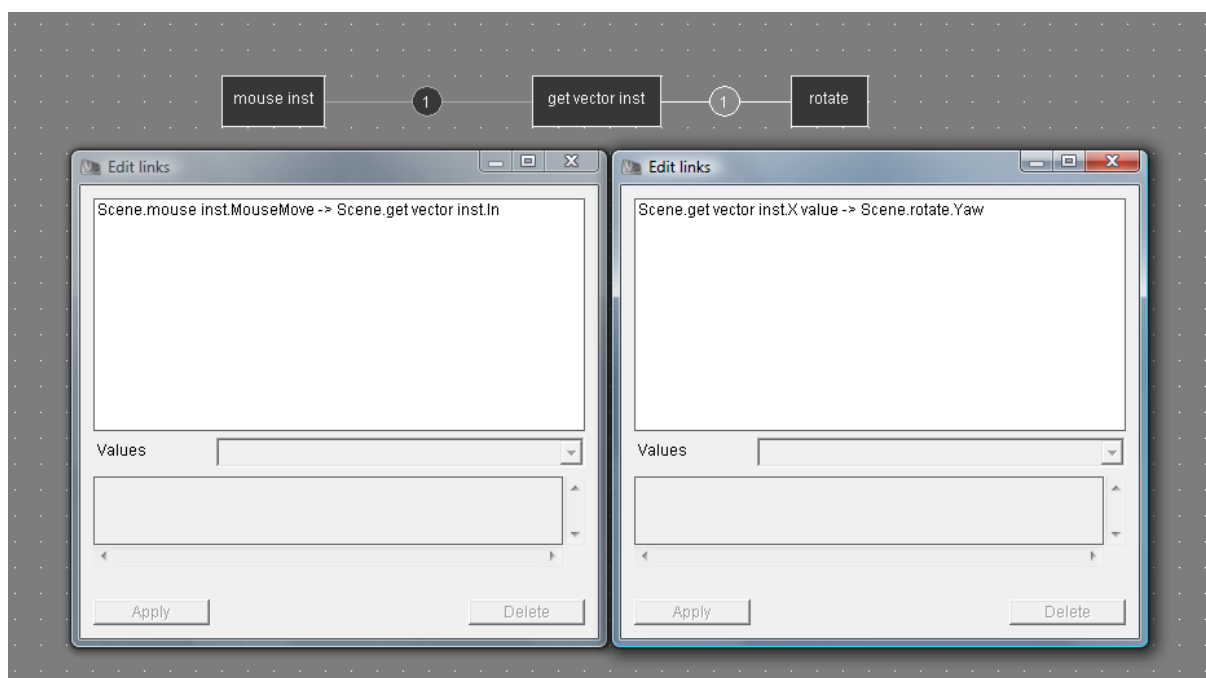
Les plugIT « Maths »

PlugIT Get vector

Ce PlugIT permet de renvoyer la valeur de vecteur X, Y ou Z en sortie.

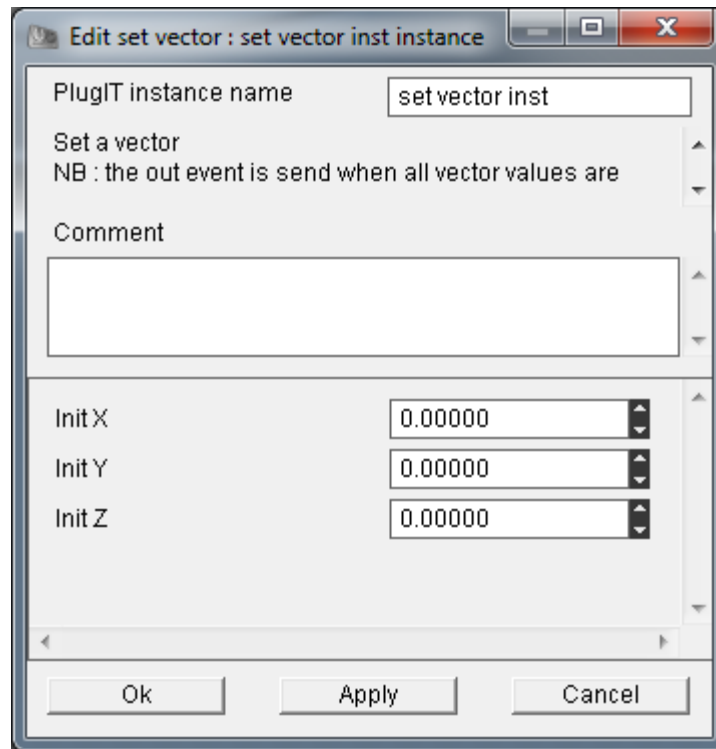


Exemple :



Dans cet exemple, la valeur X de déplacement de la souris est récupérée par le plugIT Get Vector qui l'envoie ensuite pour modifier la valeur Yaw sur le plugIT Rotate

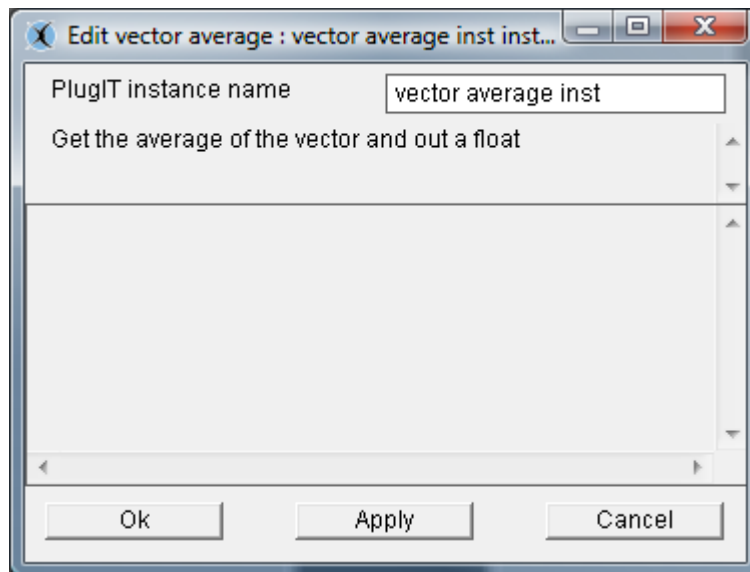
PlugIT Set Vector



Le PlugIt Set Vector permet de créer ou de modifier la valeur d'un vecteur en fonction de la valeur de vecteur d'entrée et la nouvelle valeur de ce vecteur peut alors être utiliser en sortie

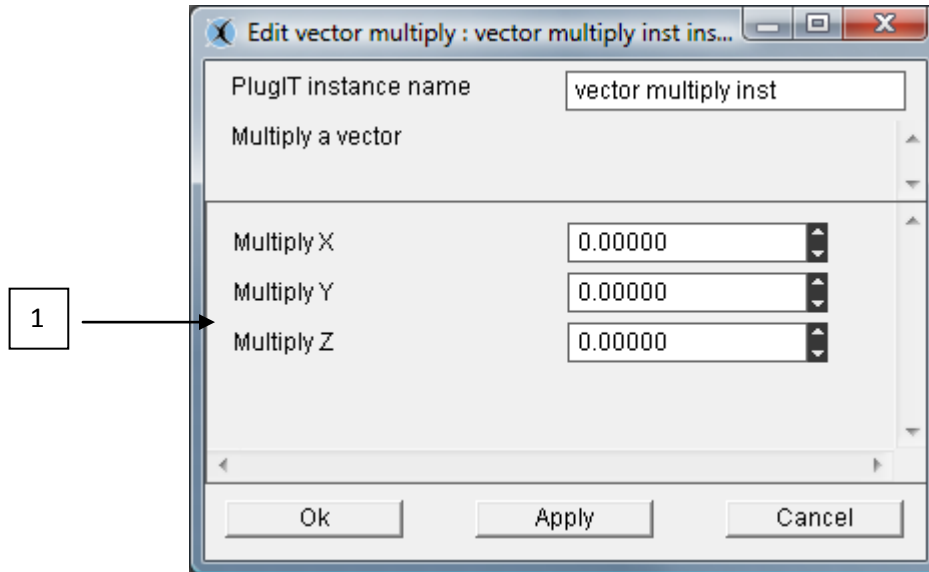
PlugIT vector average

Ce plugIT permet simplement de faire la moyenne sur les valeurs d'un vecteur en entrée. A la sortie du plugIT on récupère cette valeur.



PlugIT vector multiply

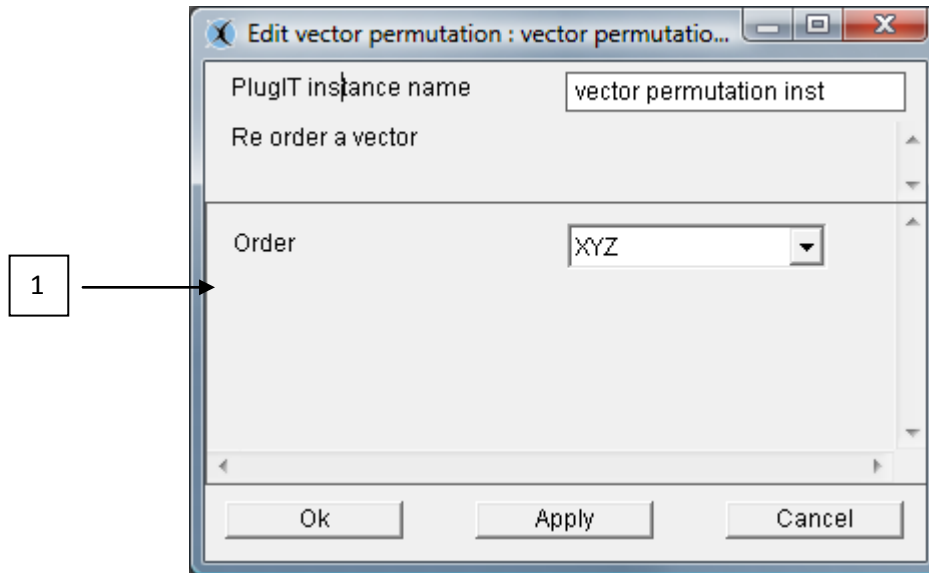
Ce plugIT permet la multiplication de deux vecteurs pour renvoyer le vecteur final en sortie.



1 °/ Vecteur d'entrée

PlugIT vector permutation

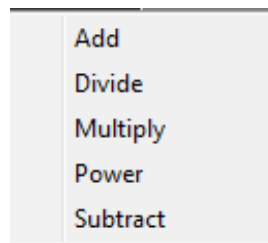
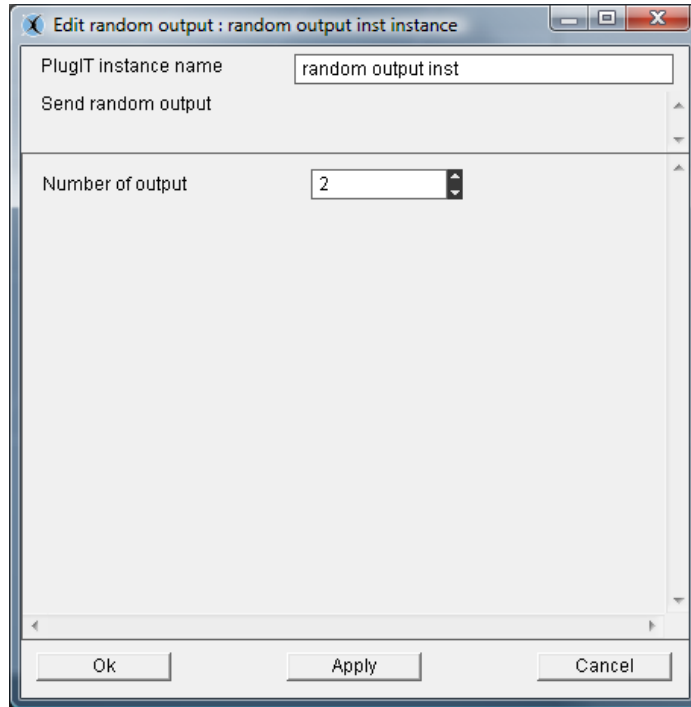
Ce plugIT permet des transformations sur l'ordonnement des valeurs sur un vecteur. Il permet de transformer un vecteur suivant le type de transformation souhaitée.



1 ° Les différents types de transformations sur le vecteur

PlugIT operator

Le PlugIT operator vous permet d'effectuer une opération de calcul sur une valeur : multiplication, division, soustraction, addition.

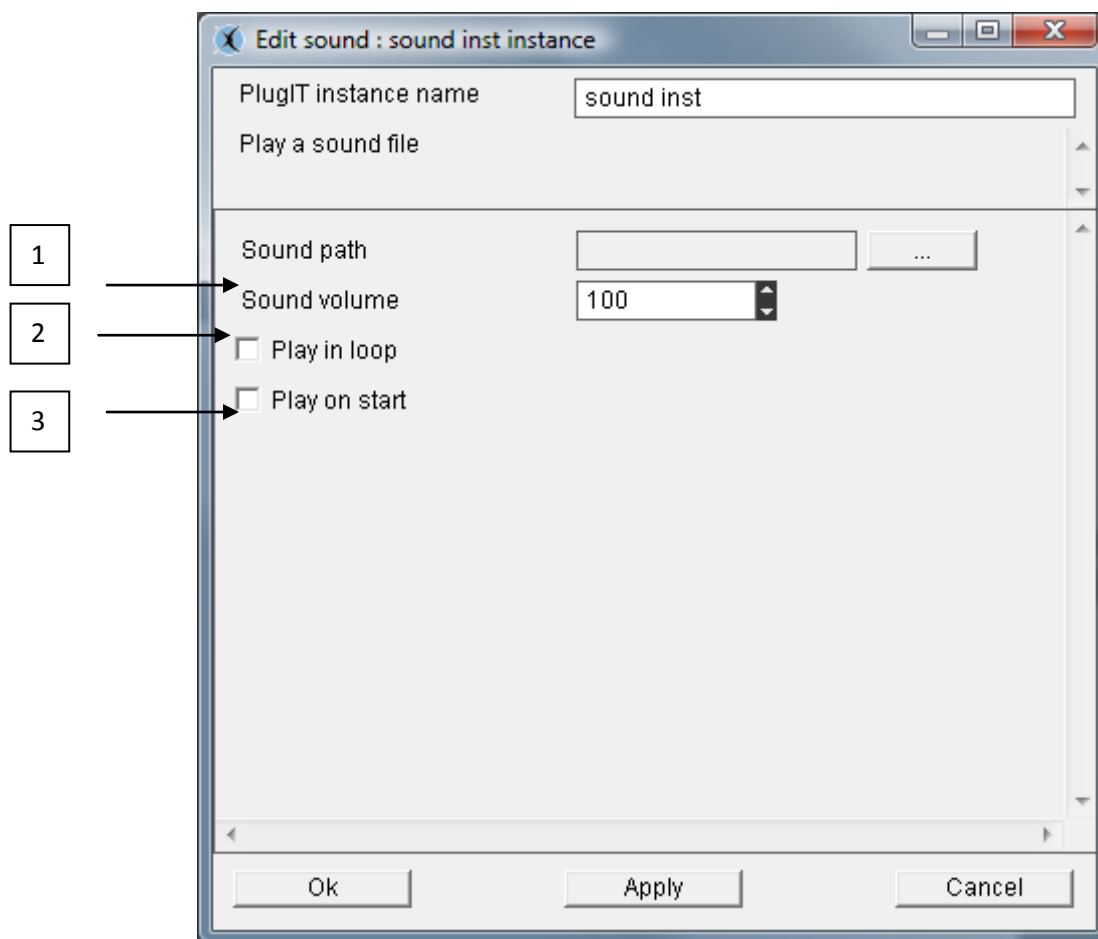


Les plugIT « Media »

PlugIT Sound

Le PlugIT Sound permet de déclencher un son au lancement de l'application ou lors de l'action « play ».

En suivant la même méthode que précédemment, ajouter un PlugIT Sound, puis éditer l'instance.



1 °/ Sélectionner le fichier son, qui devra se trouver dans votre partition scol, en cliquant sur le bouton parcourir « ... ». Les formats Mp3 et wav sont reconnus.

2 °/ Choix du volume du son

3 °/ Le paramètre « Play in loop » définit si le son sera lu en boucle.

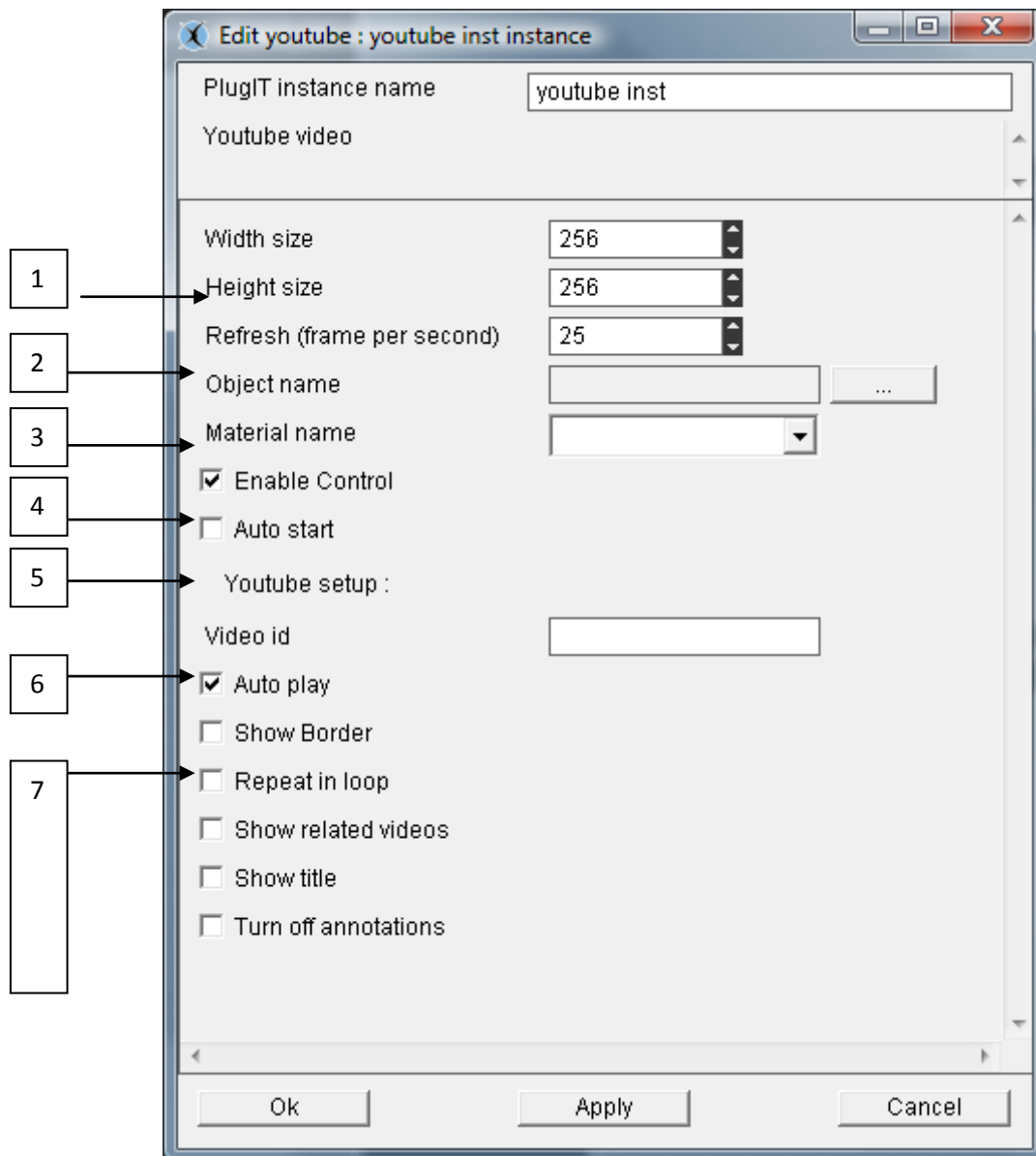
Le paramètre « Play on start » permet de définir si le son sera démarré automatiquement au lancement de l'application.

PlugIT Youtube

Le PlugIT Youtube permet d'afficher une vidéo youtube sur un objet 3d.

En suivant la même méthode que précédemment pour ajouter un PlugIT Youtube, puis éditer l'instance.

Clic droit dans une zone vierge : media, sélectionner « Youtube



1 °/Les paramètres « Width size » et « Height size » permettent de définir la taille utilisée pour la texture appliquée sur le matériel. Plus la taille sera élevée et plus la qualité sera accrue, mais au dépend des performances de l'application.



2 °/Le paramètre « Refresh » définit le nombre maximum d'image par seconde pour afficher le contenu du flash, plus la valeur est élevée et plus l'animation sera fluide, mais au dépend des performances.

3 °/Cliquer sur le bouton parcourir « ... » pour sélectionner l'objet puis le matériau qui serviront de support pour le flash.

4 °/Le paramètre « Enable Control » permet d'activer ou de désactiver le contrôle de la souris sur l'animation flash.

5 °/Le paramètre « Auto start » permet d'activer le démarrage automatique du PlugIT au lancement de l'application.

6 °/Le paramètre « Video id » est l'identifiant de la vidéo youtube.

7 °/Le paramètre « Show border » affiche une bordure autour de la vidéo.

Le paramètre « Repeat in loop » permet de définir si la vidéo sera lue en boucle ou non.

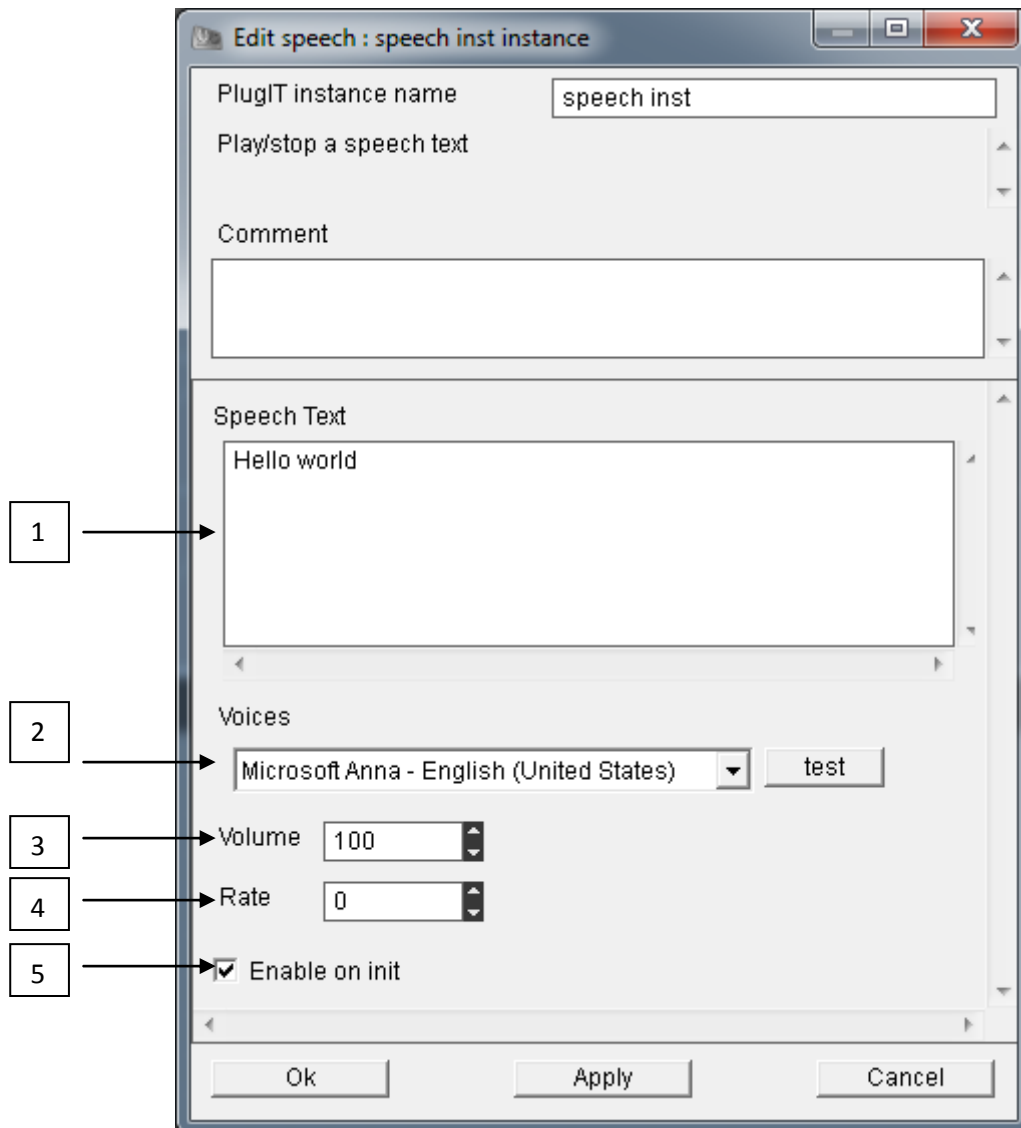
Le paramètre « Show related videos » affiche les vidéos similaires.

Le paramètre « Show title » affiche le titre de la vidéo.

Le paramètre « Turn off annotations » Désactive les annotations sur la vidéo.

PlugIT Speech

Le PlugIT Speech vous permet d'utiliser l'Api Text-to-Speech de Windows pour générer une voix qui sera émise en fonction de l'évènement associé.



1°/ Zone de saisie du texte qui doit être lu par la synthèse vocale

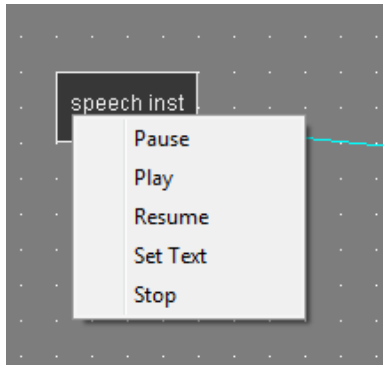
2°/ Sélection de la voix à utiliser pour lire le texte parmi celles qui sont installées sur l'ordinateur

3°/ Volume de lecture du texte

4°/ Taux de lecture, ou vitesse à laquelle le texte doit être lu

5°/ Cochez cette case si vous souhaitez que ce texte soit lu à l'initialisation de la scène.

Les liens en entrée du PlugIT Speech permettent de lier un événement sortant d'un autre PlugIT pour engendrer un action au PlugIT Speech

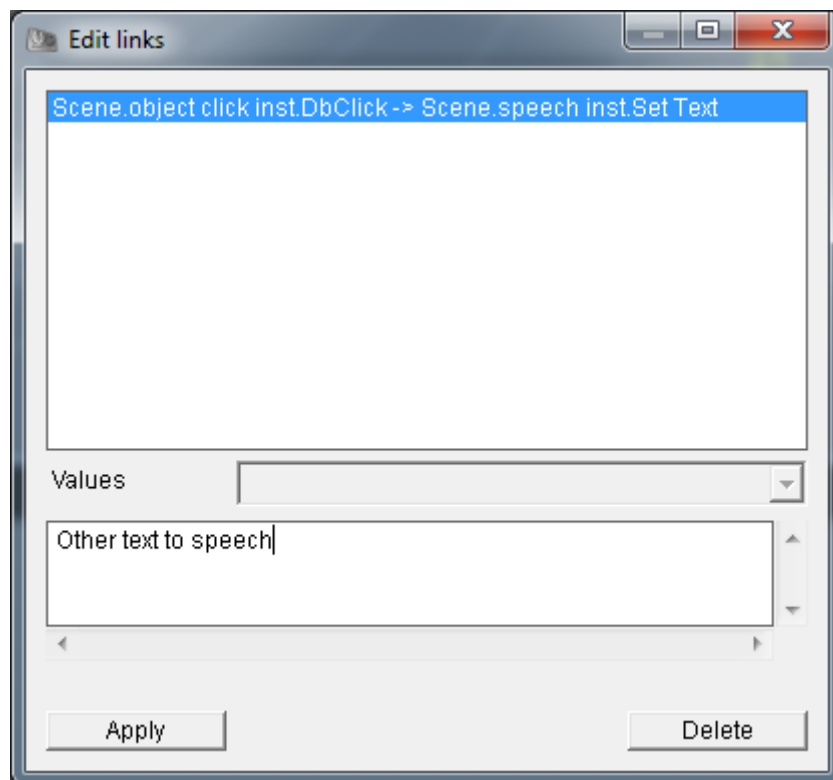


1°/ Pause permet de faire une pause sur la lecture du texte

2°/ Play permet de lancer la lecture du texte

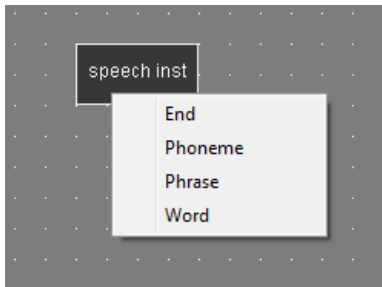
3°/ Resume reprendre la lecture du texte là où il avait été mis en pause

4°/ Set text permet de changer le contenu du texte en passant un autre texte en paramètre du lien "set text"



5°/ Stop permet d'arrêter la lecture du texte.

Les liens en sortie du PlugIT Speech permettent de lier des évènements de ce plugIT pour engendrer des actions vers d'autres PlugITs



exemple sur un texte du type Bonjour je suis la

1°/ End reconnaît la fin de la lecture du texte

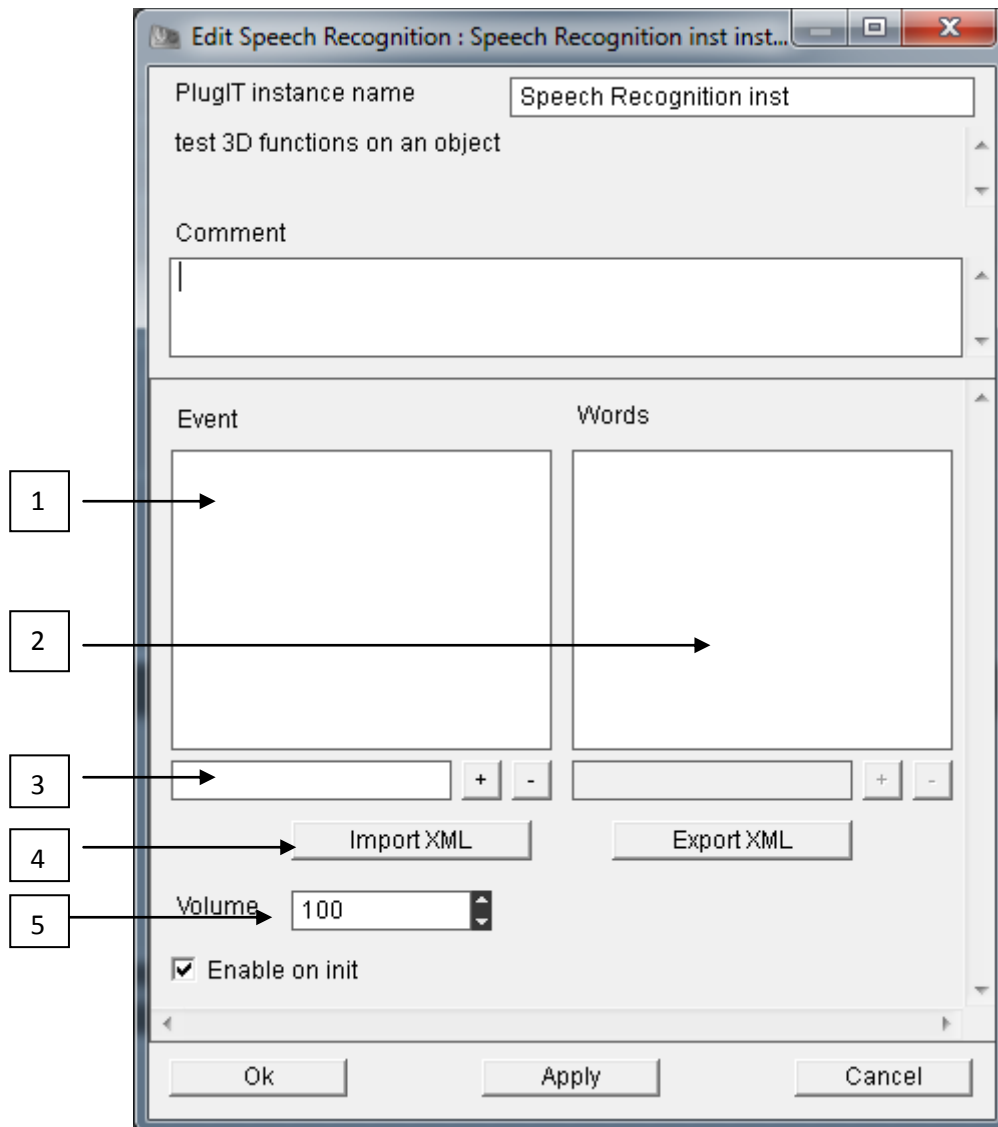
2°/ Phoneme reconnaît une syllabe comme "bon", "jour", "je", "suis", "la"

3°/ Phrase "bonjour je suis la"

4°/ Word "bonjour"

PlugIT Speech Recognition

Le PlugIT Speech Recognition vous permet d'utiliser la reconnaissance vocale pour intégrer avec l'application pour exemple déclencher une action sur un PlugIT lorsqu'un mot est détecté



1°/ Event sont les événements

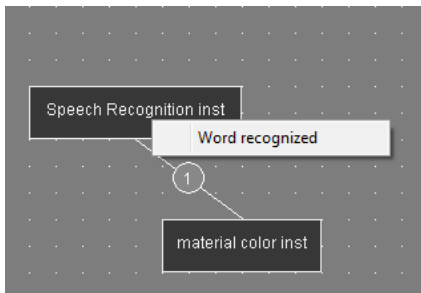
2°/ Words correspondent aux mots reconnus qui seront associés aux événements

3°/ Zone de saisie de nouveaux événements ou de mots à reconnaître

4°/ Import/Export de fichier .xml

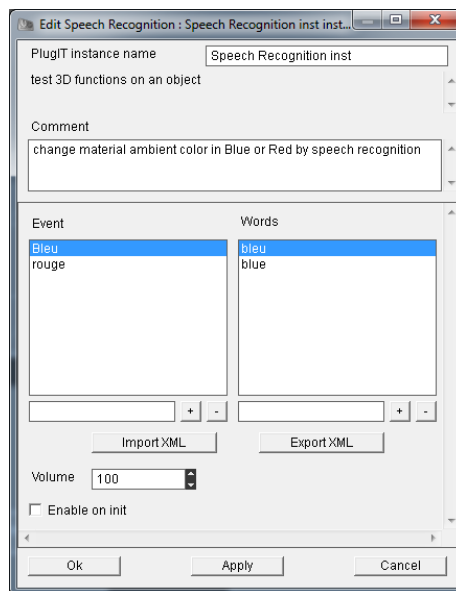
5°/ Volume

Tous les évènements créés dans le PlugIT Speech recognition sont ensuite disponible en sortie du PlugIT



Exemple d'intégration du PlugIT Speech Recognition pour changer la couleur d'un material lorsque le mot reconnu correspond à une couleur dans les événements du PlugIT :

Les événements Bleu et Rouge ont été créés dans l'interface du PlugIT Speech Recognition, à chacun d'eux les mots anglais et français qui seront reconnus ont été associées aux événements respectifs



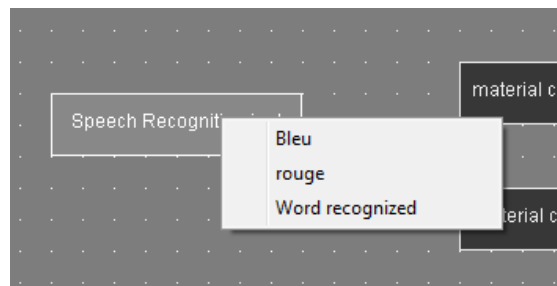
Voici la structure de l'xml qui peut en être exporté :

```

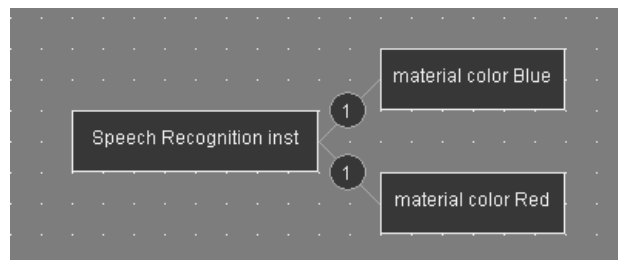
=<wordlist>
=<event value="Bleu">
<word value="bleu" />
<word value="blue" />
</event>
=<event value="rouge">
<word value="red" />
<word value="rouge" />
</event>
</wordlist>

```

Les événements disponibles en sortie du PlugIT Speech Recognition sont donc :



Les PlugITs permettant de changer la couleur de l'ambient du material ont été ajoutés et liés aux événements du PlugIT Speech recognition :

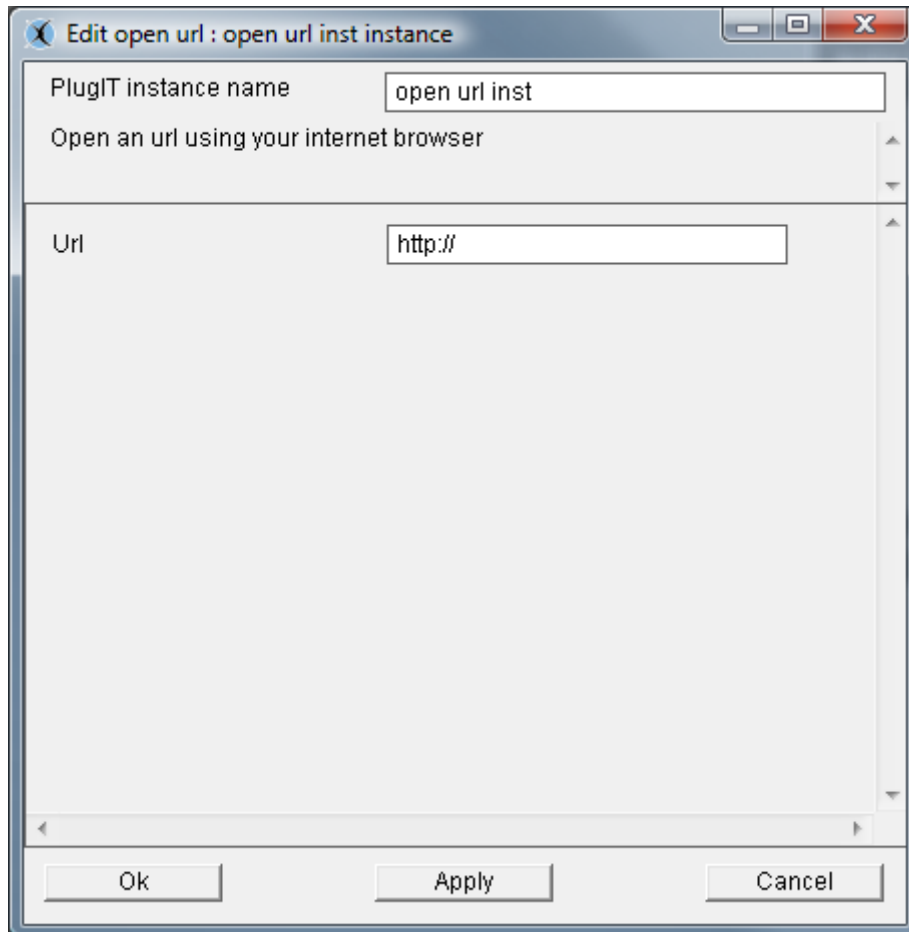


Désormais lorsque l'utilisateur prononcera par exemple le mot "blue" l'ambient du material sera changée en bleu

Les plugIT « Misc»

PlugIT Open Url

Le PlugIT Open Url permet d'ouvrir une url (page web)



Le paramètre « Url » permet de saisir l'url qui devra s'ouvrir.

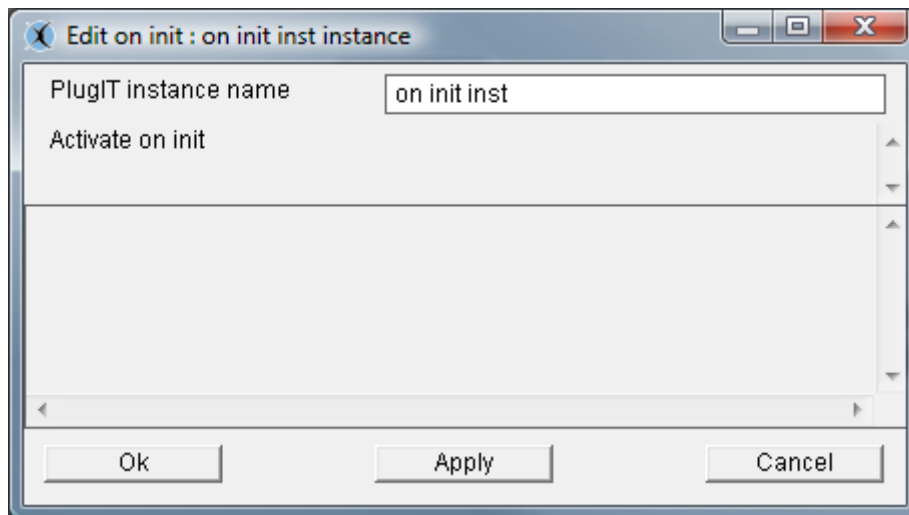
PlugIT On init

Le PlugIT On Init permet de déclencher un évènement lorsque toute la scène est initialisée.

En suivant la même méthode que précédemment, ajouter un PlugIT On Init, puis éditer l'instance.

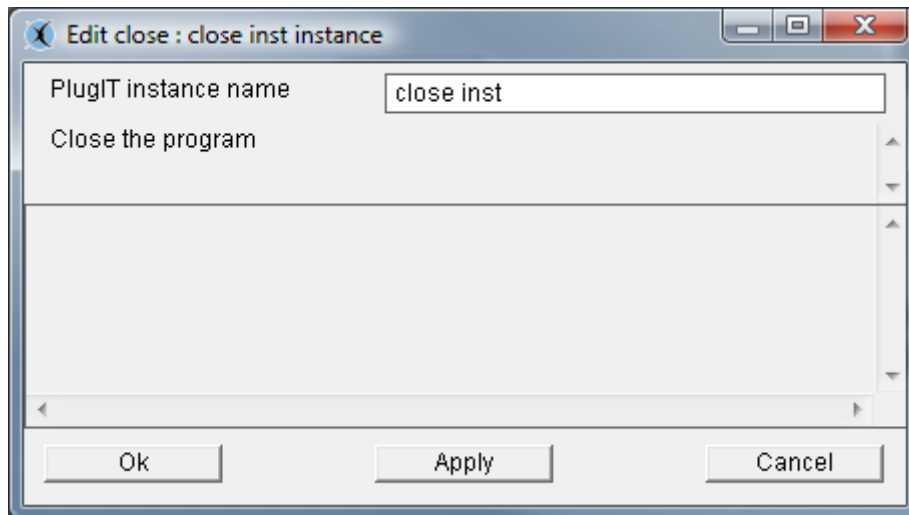
Cette fonction n'a pas de paramètre à éditer.

Il sera utilisé avec la plupart des autres PlugITs pour exécuter une action au lancement de l'application.



PlugIT Close

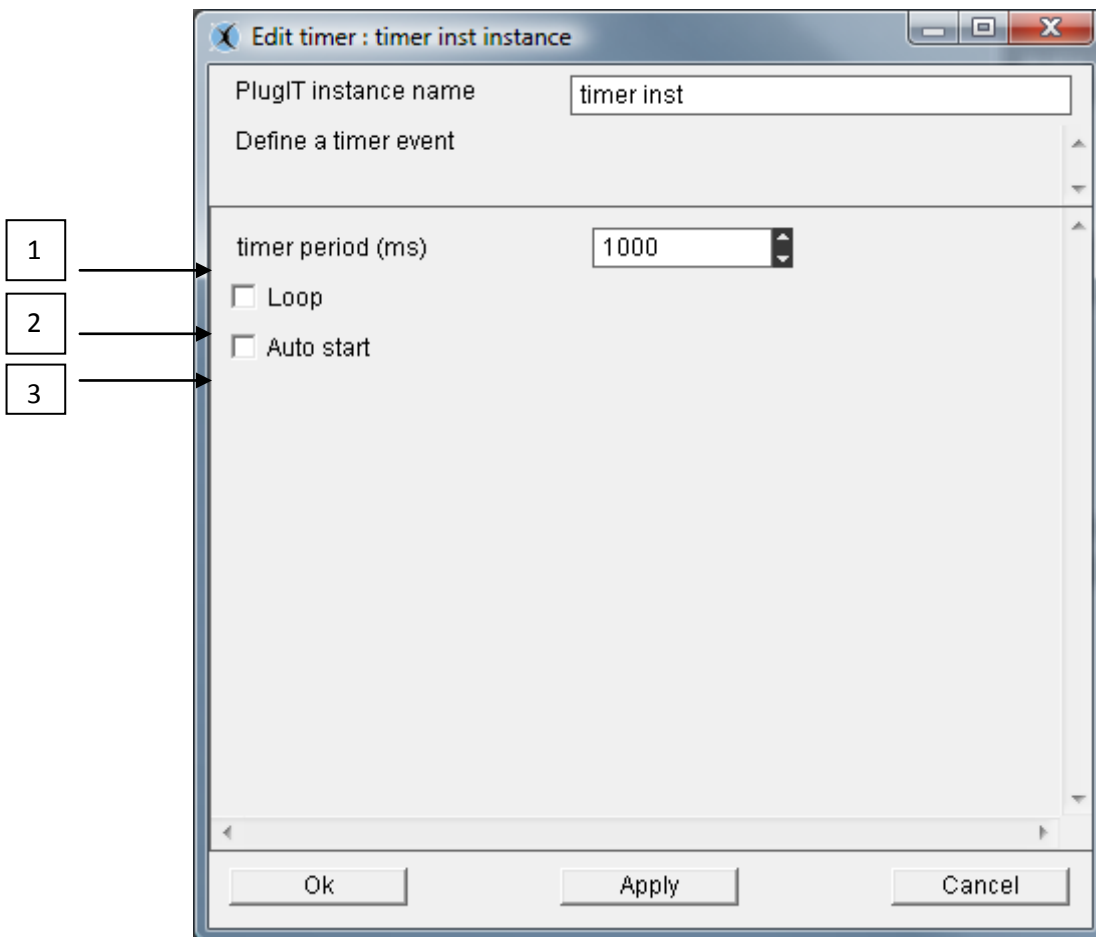
Le PlugIT Close permet de fermer l'application.



Cette fonction n'a pas de paramètres à éditer.

PlugIT Timer

Le PlugIT Timer permet de déclencher un évènement à une période donnée.



1 °/Le paramètre « timer period » permet de définir en millisecondes la durée de la période avant le déclenchement de l'évènement.

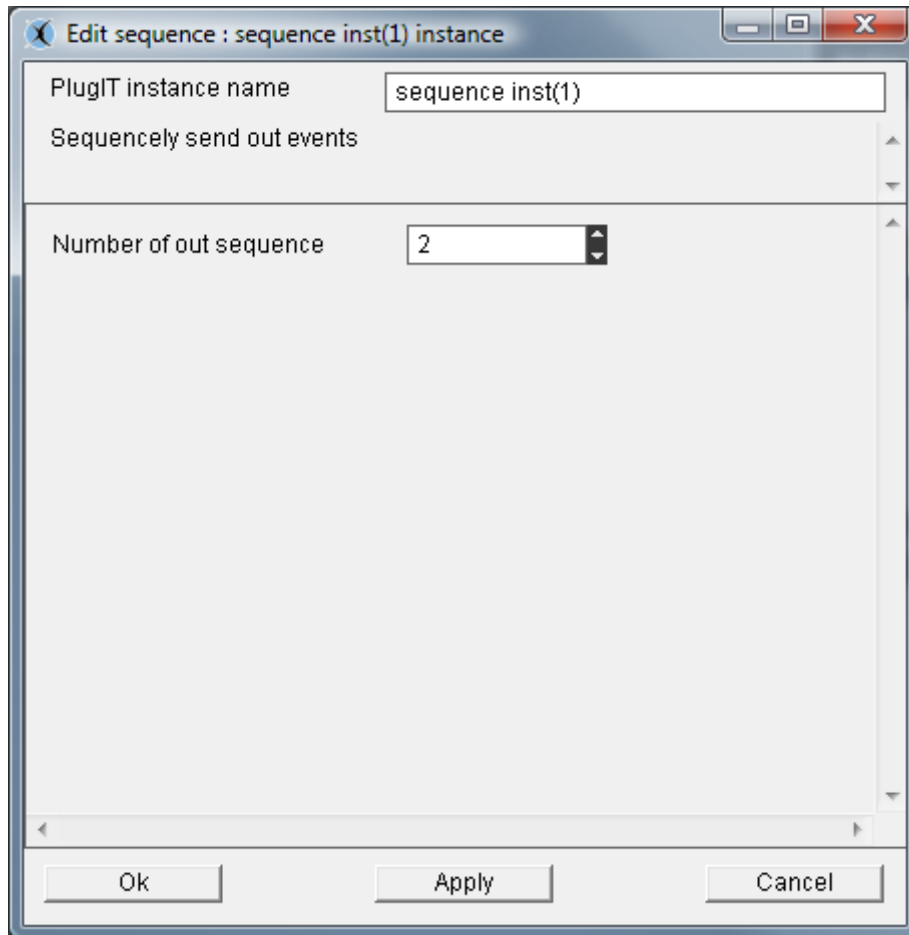
2 °/Le paramètre « Loop » définit si l'évènement sera déclenché une seule fois après la période donnée ou toujours à intervalle régulier de cette même période.

3 °/Le paramètre « Auto start » permet de définir si le timer sera lancé automatiquement au lancement de l'application.

PlugIT Séquence

Le PlugIT Séquence permet de déclencher deux évènements de façon alternative.

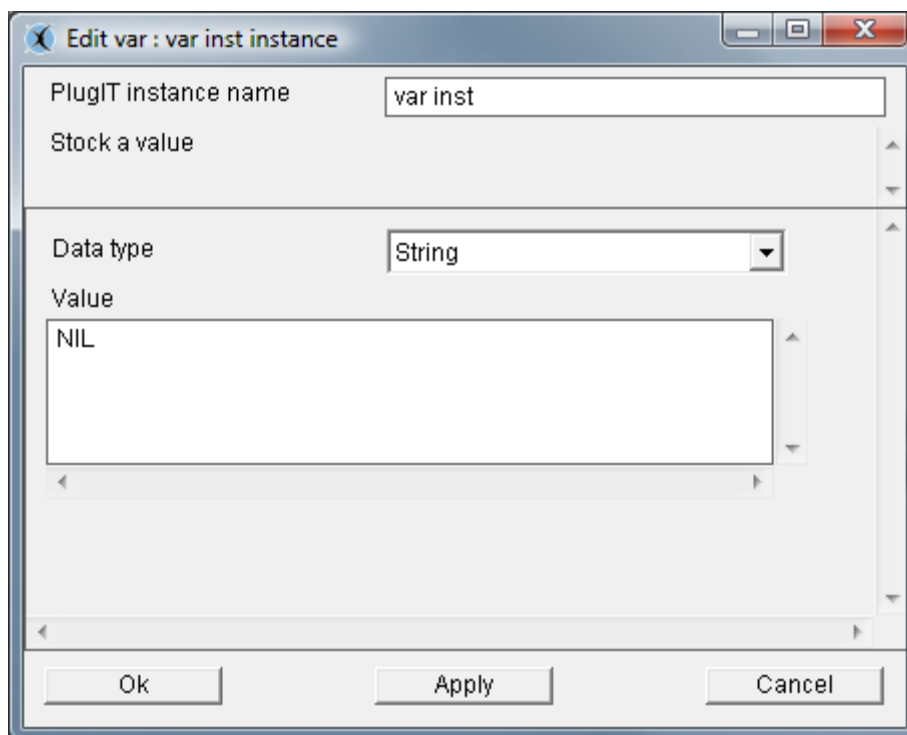
Cette fonction ne permet que d'éditer le nombre de séquences pouvant être raccordées à un autre plugit



Par exemple cette fonction peut servir à déclencher sur un même clic via le PlugIT Object click alternativement le play ou le stop d'une animation ou d'une vidéo.

PlugIT Var

Le PlugIT Var permet de créer une variable.

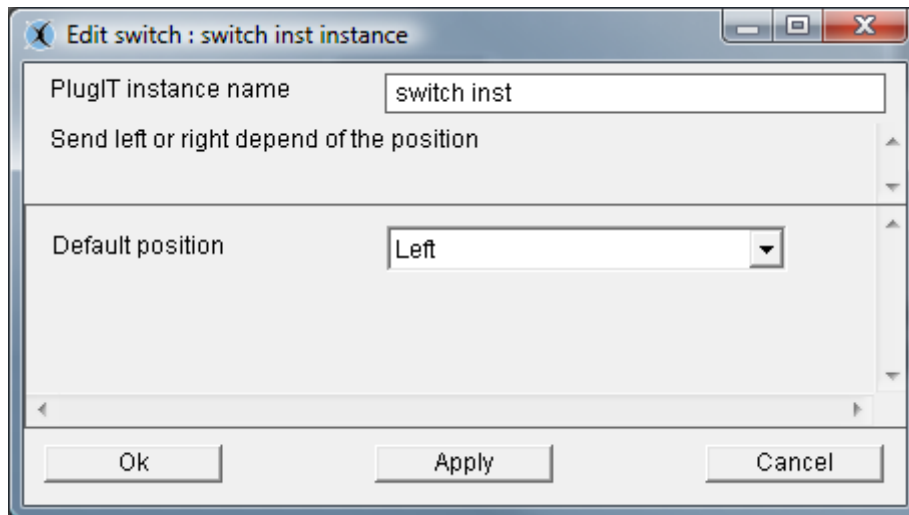


Le paramètre « Data type » est le type de variable souhaitée.

Le paramètre « Value » est la valeur de la variable à l'initialisation du PlugIT.

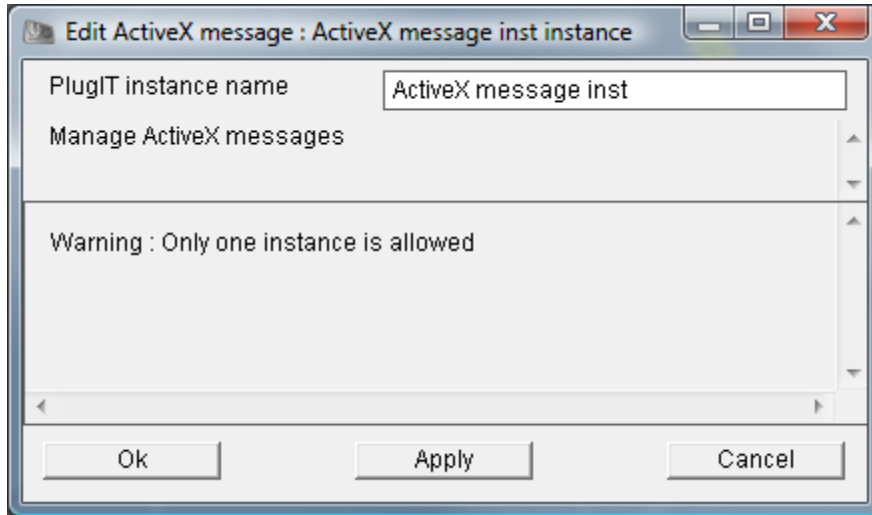
PlugIT Switch

Le PlugIT Switch permet de définir deux sorties et de choisir quelle sortie utiliser à un instant donné.



Le paramètre « Default Position » Left ou Right permet de définir la position de l'interrupteur à l'initialisation du PlugIT.

PlugIT ActiveXmessage



Voici un exemple de code de page php fonctionnant avec le PlugIT ActiveXmessage

```
<?php
include "conf.inc.php";

?>

<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

<link rel="stylesheet" href="style.css" type="text/css">

<script language="JavaScript">
<!--
function Load()
{
<?php
if($WEBINTEGRATED == 1)
echo
"document.scol.LaunchMachine("\\$browser$%5fload+%22locked%2fstduser%2epkg%22%0amain+%22" .
$SCOL_SITE_URL_OLD . "%22+ffffff+NIL CSDMRWK 1000000\",1,0);";
else
echo "document.scol.LaunchMachine("\\$browser$%5fload+%22locked%2flink%2epkg%22%0amain+%22" .
$SCOL_SITE_URL_OLD . "%22%0a CSDMRWK 1000000\",1,0);";
?>
}
//-->
</script>

</head>
```



```

<body onLoad="Load()">
<div align="center">
<OBJECT classid="clsid:7A96FF35-4937-11D1-8F2C-00609779BDA3"
  codebase="<?echo $WIN_32_PLUGIN_URL?>"
  width="<?php if($WEBINTEGRATED == 1) echo $WIDTH; else echo '0'?>"
  height="<?php if($WEBINTEGRATED == 1) echo $HEIGHT; else echo '0'?>"
  id="scol">
  <PARAM name="ForcelInstall" value="no" />
  <PARAM name="ScolVersionNeeded" value="<?php echo $MINIMUMSCOLVERSION;?>" />
  <EMBED
  TYPE="application/x-scol"
  width="<?php if($WEBINTEGRATED == 1) echo $WIDTH; else echo '0'?>"
  height="<?php if($WEBINTEGRATED == 1) echo $HEIGHT; else echo '0'?>"
  name="scol"
  ForcelInstall="no"
  ScolVersionNeeded="<?php echo $MINIMUMSCOLVERSION;?>"
  onScolEnd="ScolEnd"
  onScolMessage="MessageFromScol"
  PLUGINSURL="<?php echo $WIN_32_PLUGIN_URL;?>"
  PLUGINSPAGE="<?php echo $WIN_32_PLUGIN_URL;?>"
  </EMBED>
</OBJECT>

<script>
function ScolEnd()
{
  //Your code here
  document.the_form.messval.value="Scol closed";
}

function MessageFromScol(msg)
{
  document.the_form.messval.value=msg;
  //Your code here
}

function setCarColor(nb)
{
  document.scol.SendScolMessage('AXMessage "color'+nb+'");
}
</script>

<!-- CALLBACK FOR IE !-->
<SCRIPT FOR=scol EVENT=onScolMessage(msg)>
  MessageFromScol(msg);
</SCRIPT>
<!-- CALLBACK FOR IE !-->
<SCRIPT FOR=scol EVENT=onScolEnd>
  ScolEnd();
</SCRIPT>

<form name="the_form">
<INPUT TYPE="text" NAME="messval" VALUE="" SIZE="25">

```



```
<input type=button value="Color1" onclick="setCarColor('1')">
<input type=button value="Color2" onclick="setCarColor('2')">
<input type=button value="Color3" onclick="setCarColor('3')">
<input type=button value="Color4" onclick="setCarColor('4')">
</form>

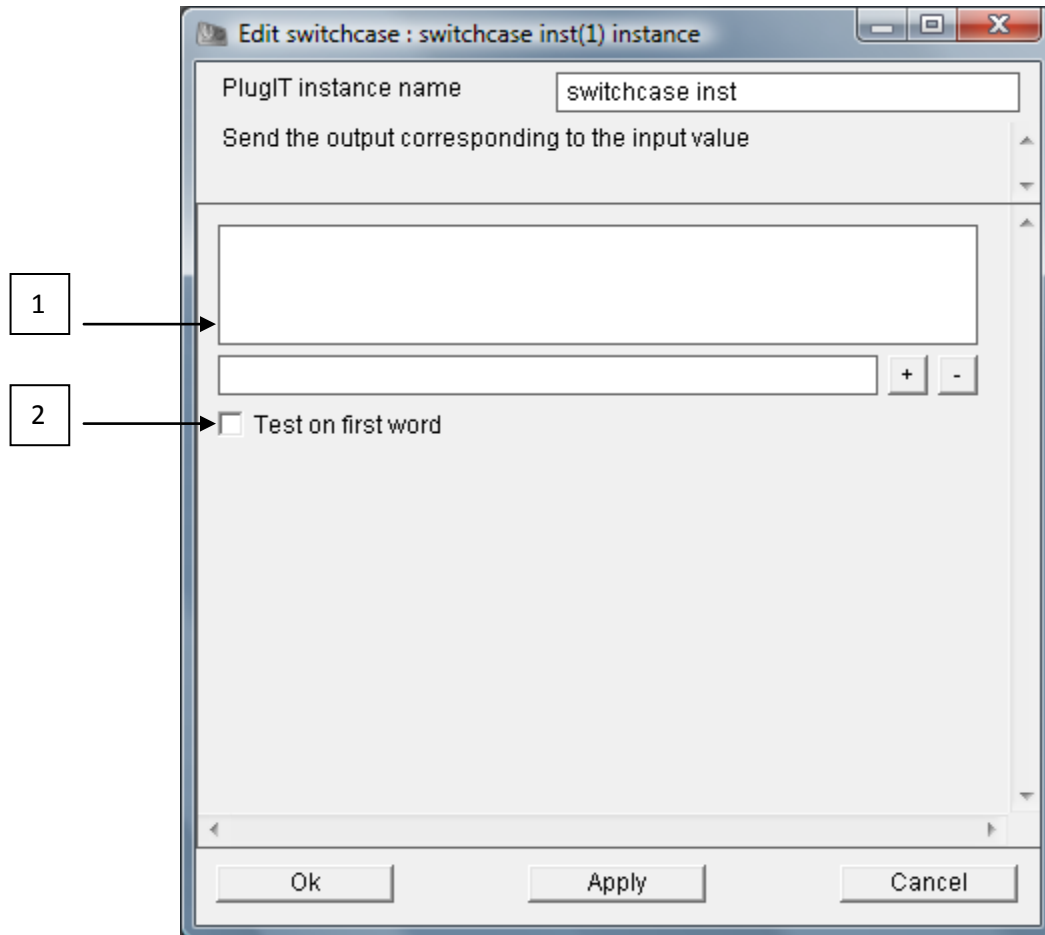
</div>
<p align="center"></p>

</body>

</html>
```

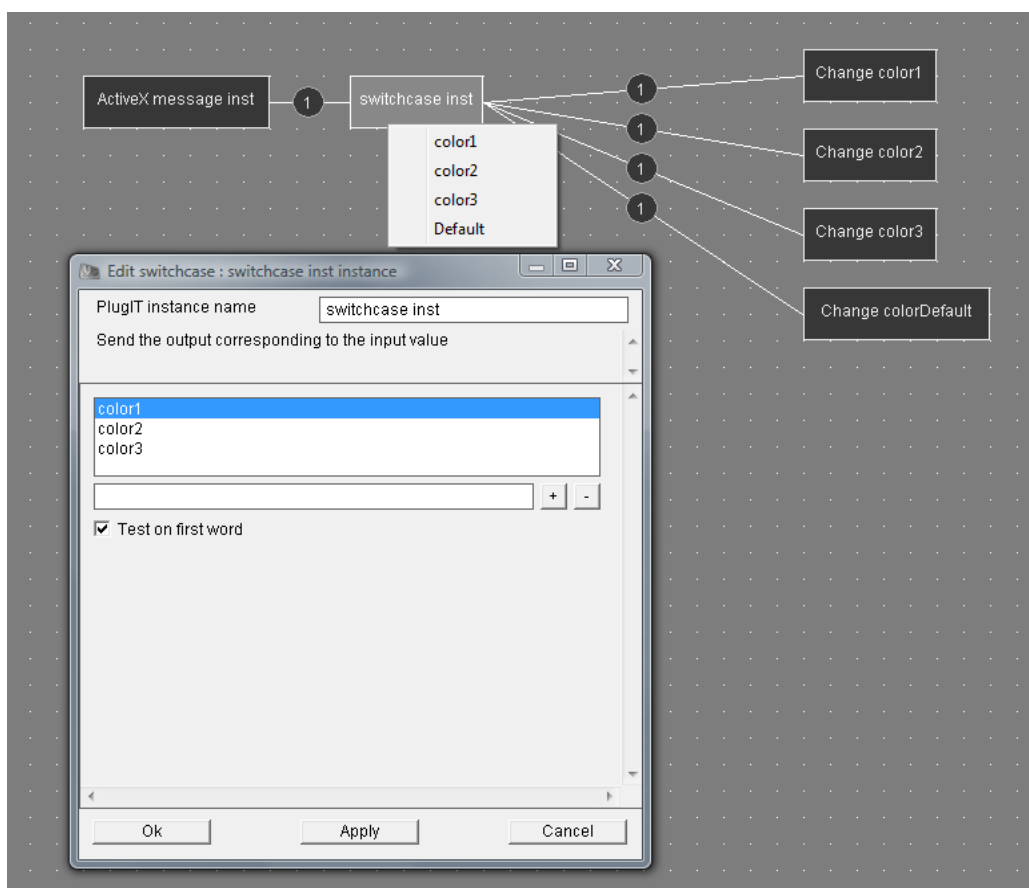
PlugIT Switchcase

Le plugIT switchcase permet de récupérer un paramètre en entrée pour ensuite exécuter l'évènement correspondant à la valeur un test sur un mot et affecter une action en fonction de la valeur



1° / Liste des paramètres

2° / activer le test sur le premier mot du paramètre uniquement

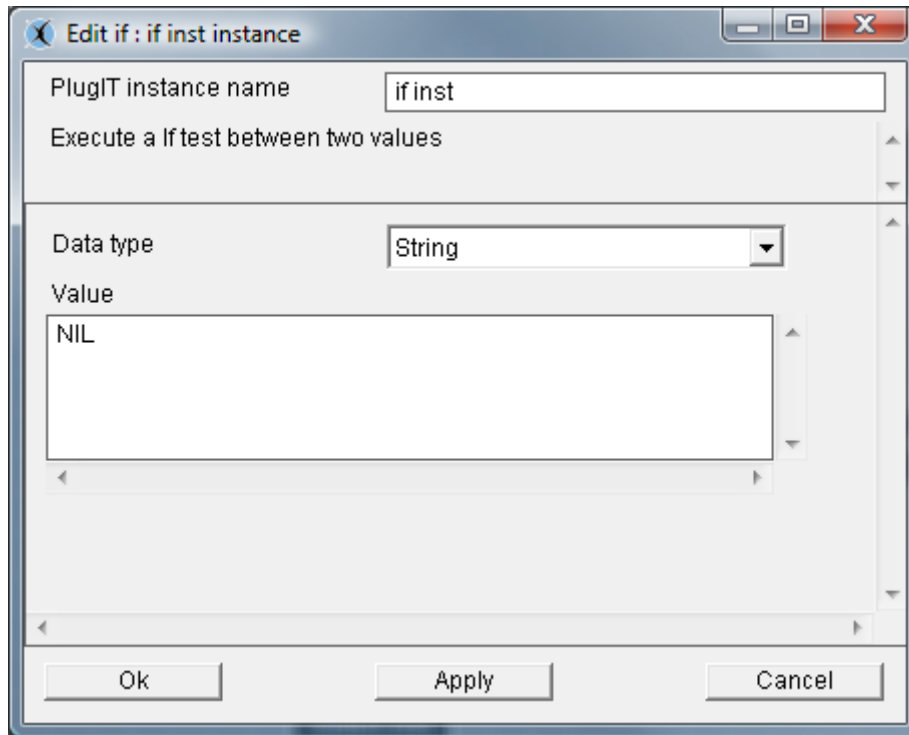


Dans cet exemple, le PlugIT ActiveXmessage envoie des paramètres de couleurs au plugIT switchcase, Lorsque le plugIT switchcase reconnaît l'un des termes qui ont été définis dans sa liste dans le premier mot qui lui est adressé par l'ActiveX alors il déclenche l'évènement qui y est associé, sinon il déclenche l'évènement défini par défaut.

Ex : Lorsque le message ActiveX contient le terme « color1 » dans ses paramètres, le plugIT Switchcase déclenche le changement de couleur défini pour la valeur color1 (Change color1).

PlugIT If

Le PlugIT If permet de tester des valeurs définies par exemple avec le PlugIT Var et de renvoyer un évènement en fonction du résultat.

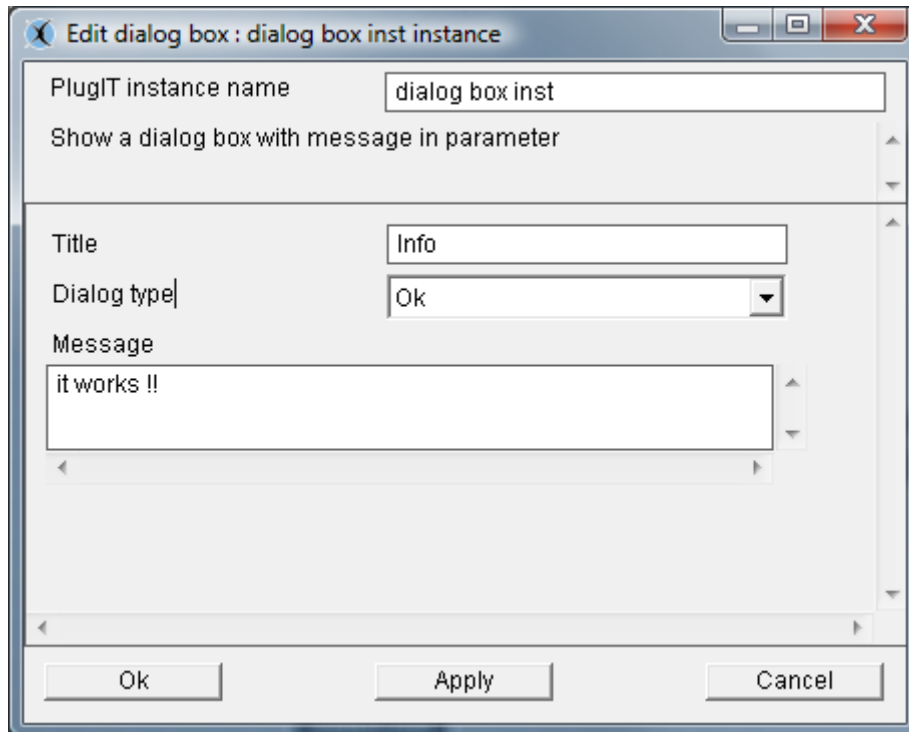


Le paramètre « Data type » est le type de variable souhaitée.

Le paramètre « Value » est la valeur de la variable à comparer.

PlugIT Dialog box

Le PlugIT Dialog box permet de créer une boîte de dialogue.



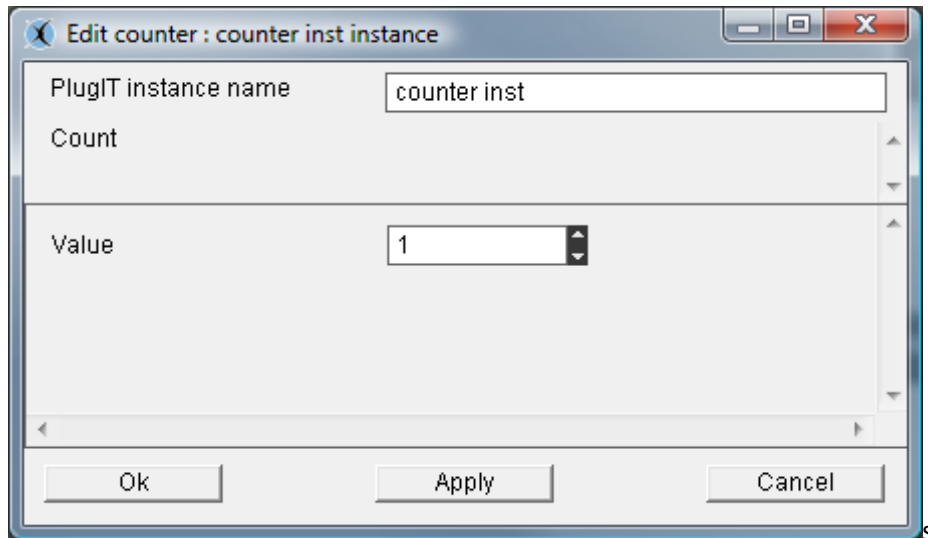
Le paramètre « Title » est le titre de la boîte de dialogue.

Le paramètre « Dialog type » est le type de boîte de dialogue souhaité, simple avec le bouton « OK » ou avec deux réponses possibles « OK / Annuler » ou « Oui / Non ».

Le paramètre « Message » est le message par défaut affiché dans la boîte de dialogue.

PlugIT Counter

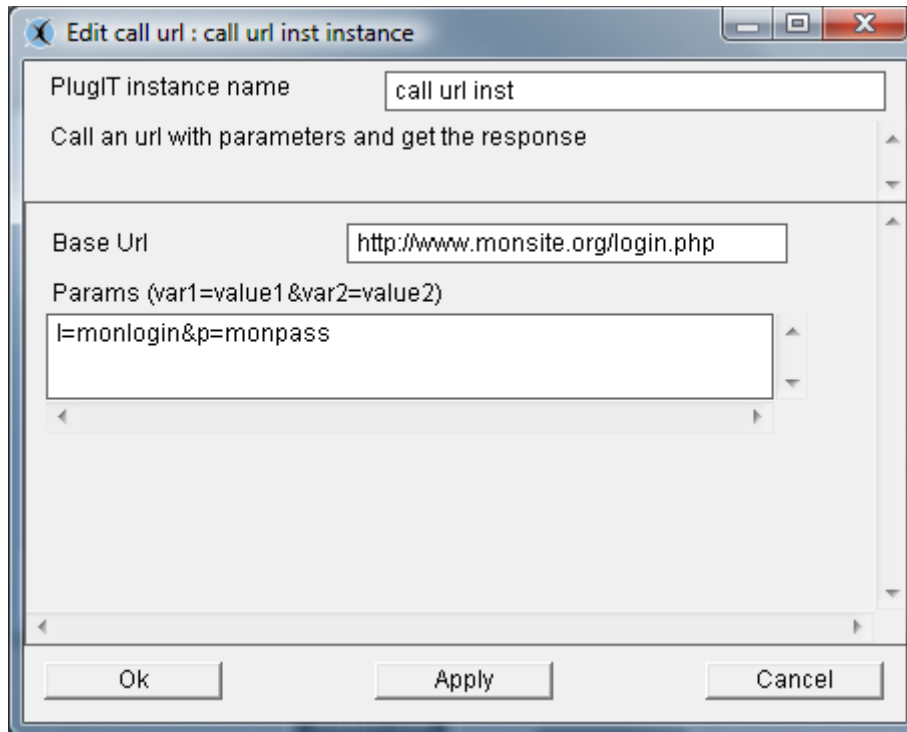
Le PlugIT Counter permet de créer une variable et de déclencher un évènement lorsque le compte est atteint.



Le paramètre « Value » est la valeur initiale du compteur.

PlugIT Call Url

Le PlugIT Call Url permet d'envoyer une requête http et de récupérer le résultat, ce PlugIT peut par exemple servir à interroger un site php.

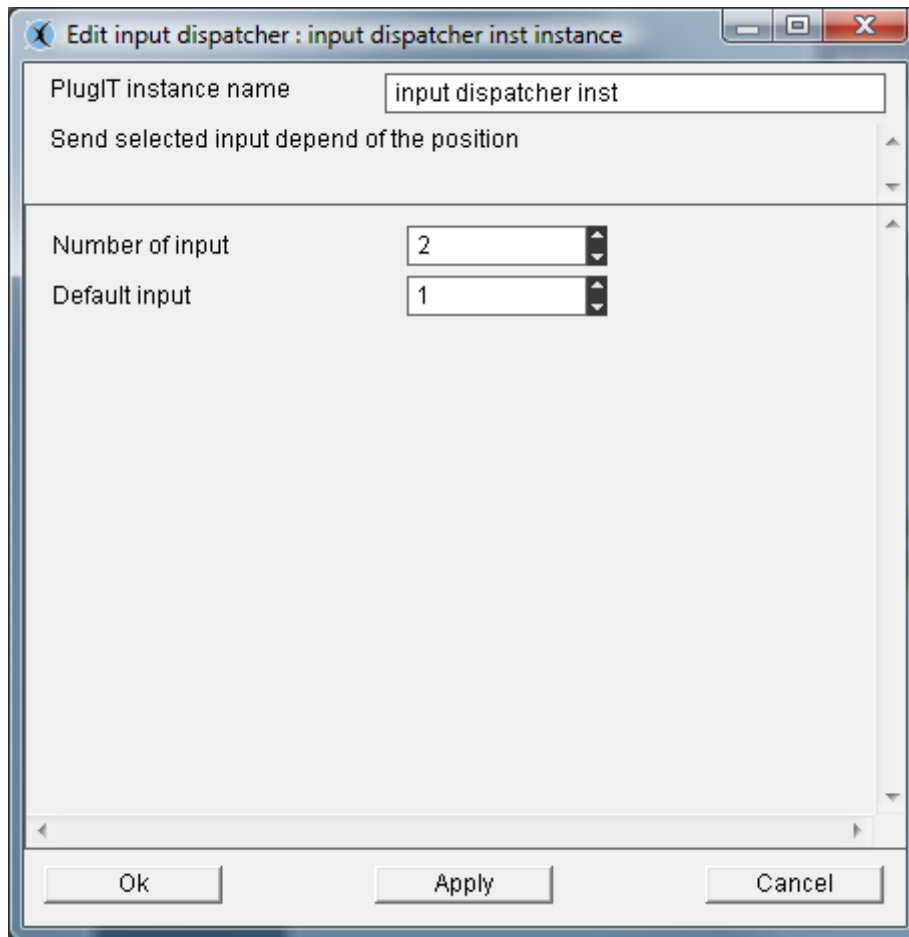


Le paramètre « Base Url » est l'url de la requête.

Le paramètre « Params » permet de préciser les paramètres à envoyer à la suite de l'url.

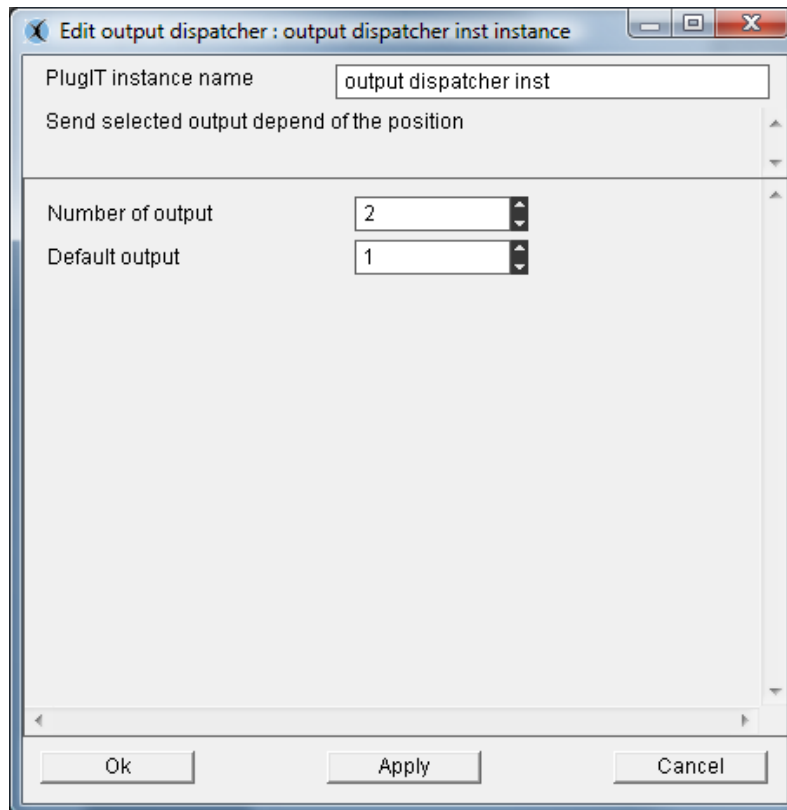
plugIT input Dispatcher

Ce plugIT logique permet de distribuer différentes entrées dépendant de la position



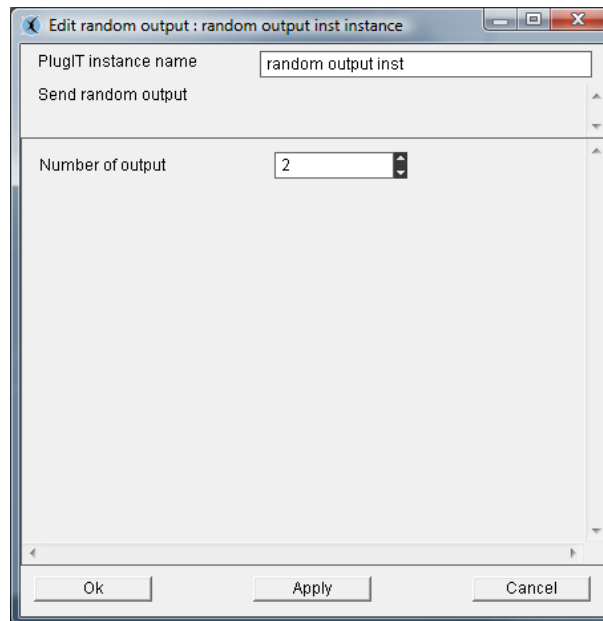
PlugIT output Dispatcher

Ce plugIT logique permet de distribuer une sortie dépendant de la position.



PlugIT Random Output :

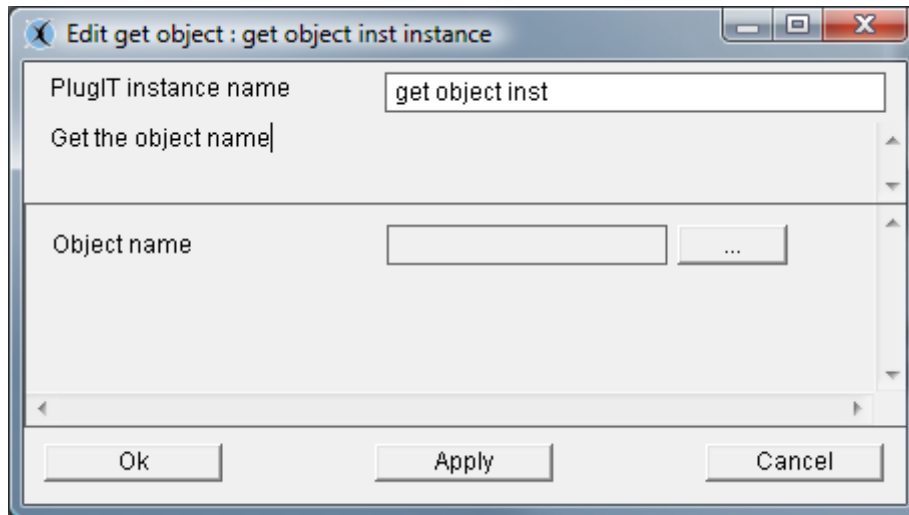
Le plugIT random output vous permet des définir un nombre d'output qui seront exécutés de manière aléatoire



Les plugIT « Object »

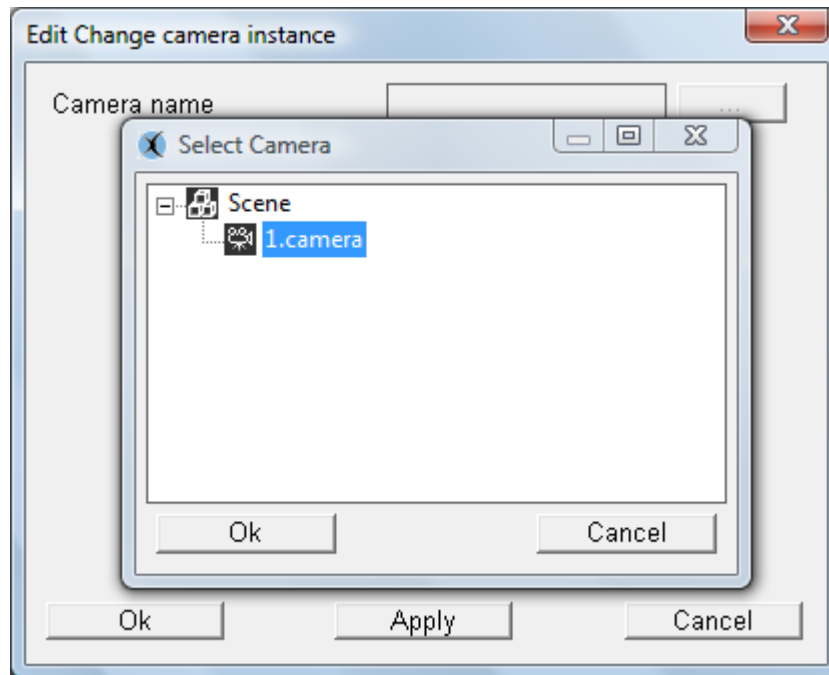
PlugIT getObject

Ce plugIT permet la récupération d'un objet de l'application par son nom, celui-ci pourra être un paramètre sur un lien vers un autre plugIT



PlugIT Set Active camera

Le PlugIT Set Active camera permet de changer la caméra utilisée.



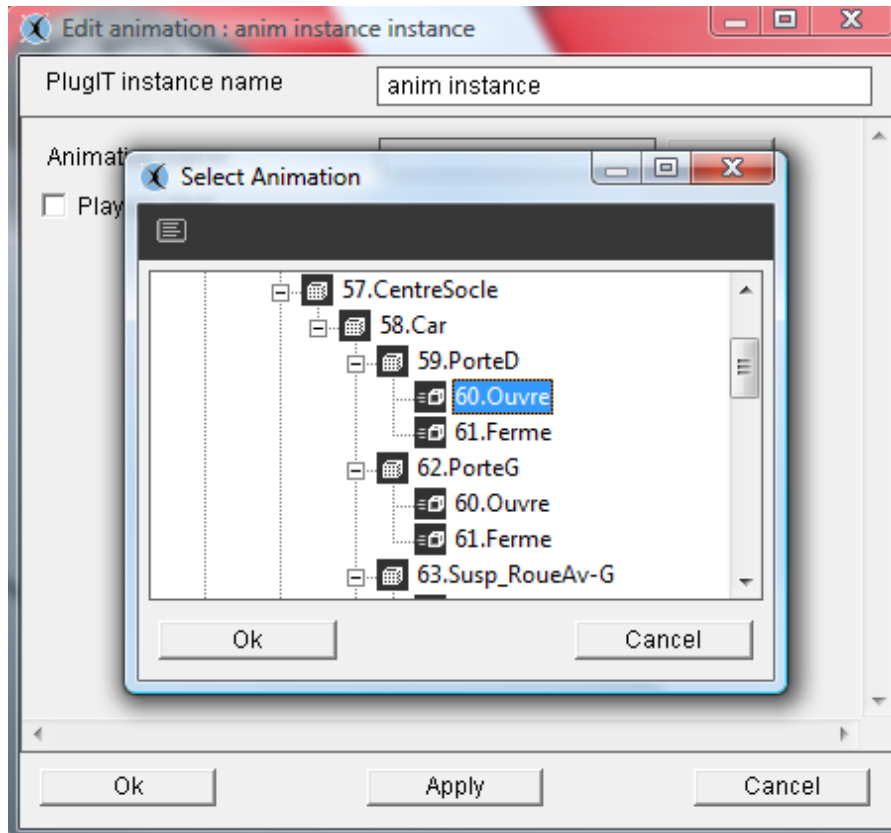
Sélectionner la caméra voulue via le bouton parcourir.

Vous pouvez ensuite choisir d'activer ou de restaurer la caméra via l'édition des liens.

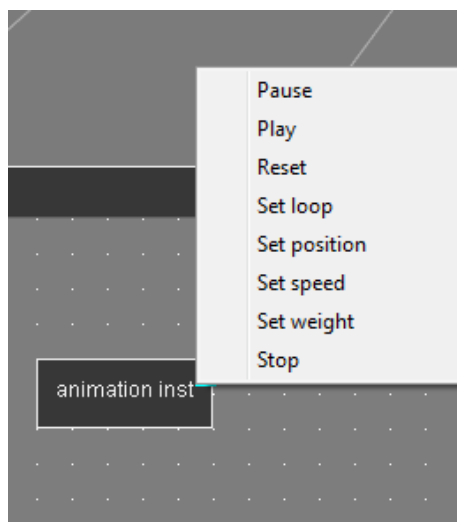
PlugIT Animation

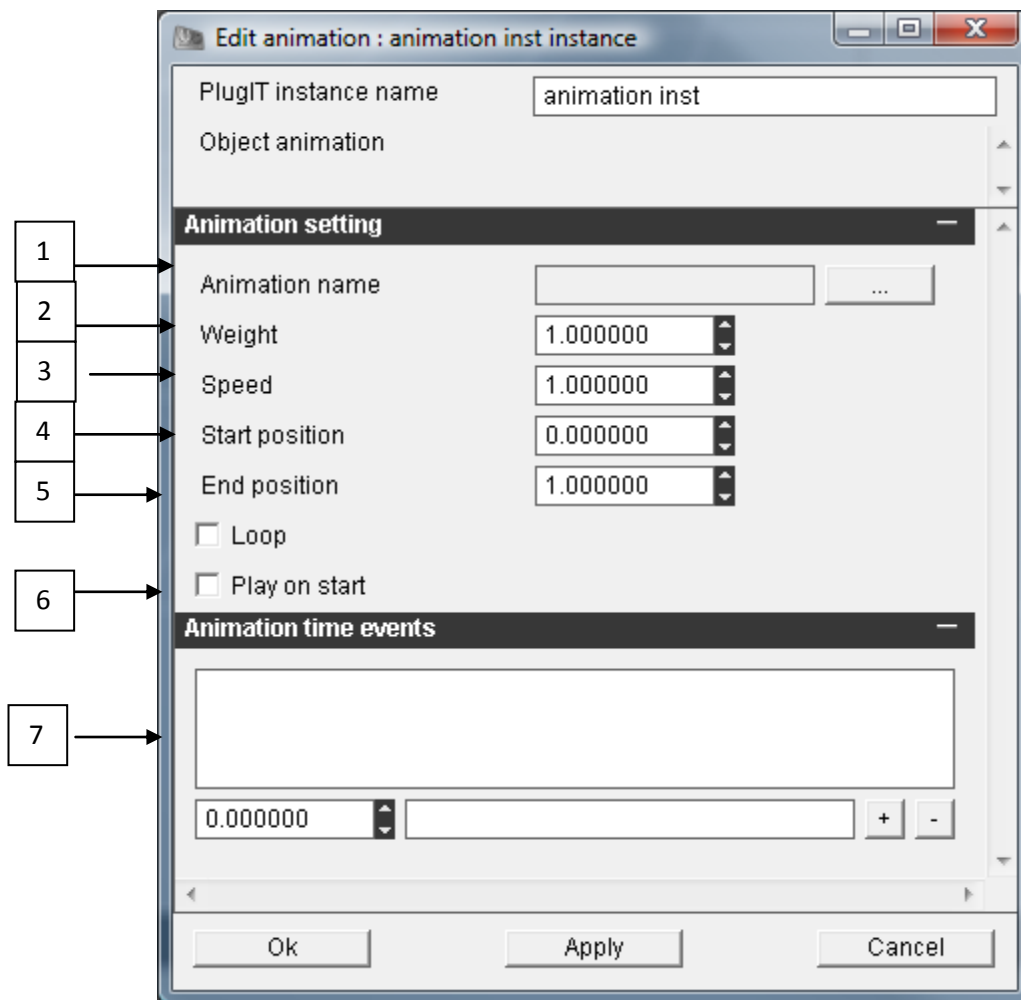
Le PlugIT Animation permet de lire ou d'arrêter une animation.

Pour contrôler une animation via la fonction Animation il est conseillé de désactiver l'animation dans l'arbre de scène.



Sélectionner l'animation via le bouton parcourir





1°/ nom de l'animation

2°/ poids de l'animation

3°/ vitesse de l'animation : permet d'accélérer ou de ralentir une animation par rapport à sa vitesse d'origine et si la valeur est négative, l'animation sera lue à l'envers

4°/ définit la valeur sur laquelle doit démarrer l'animation

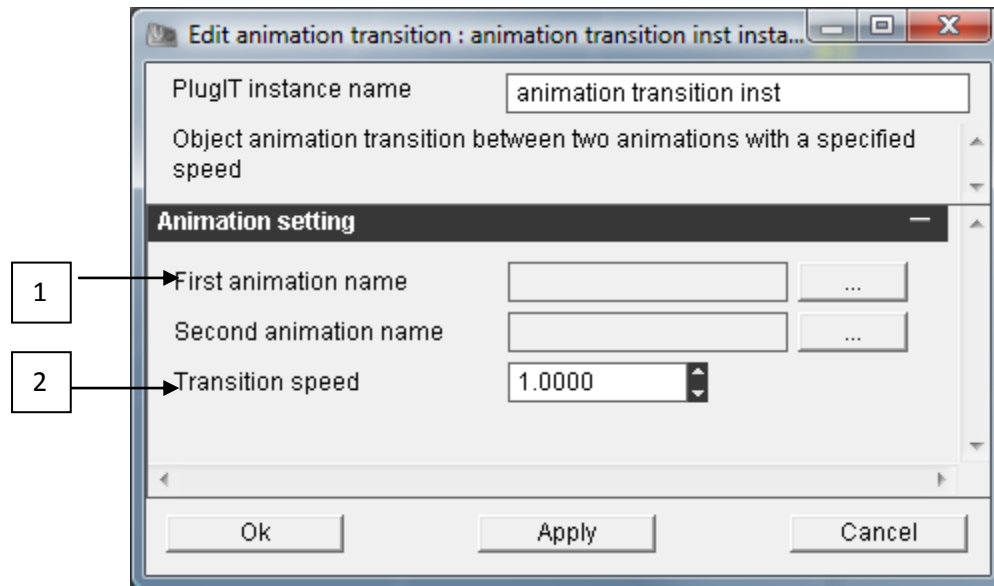
5°/ définit la valeur sur laquelle doit s'arrêter l'animation

6°/ Le paramètre « loop » définit si l'animation doit être lue en boucle. Le paramètre « Play on start » permet de définir si l'animation sera lue automatiquement au lancement de l'application.

7°/ Permet de définir des événements en fonction d'une donnée temporelle dans l'animation

PlugIT : Animation Transition

Permet de faire des transitions d'animations entre deux animations

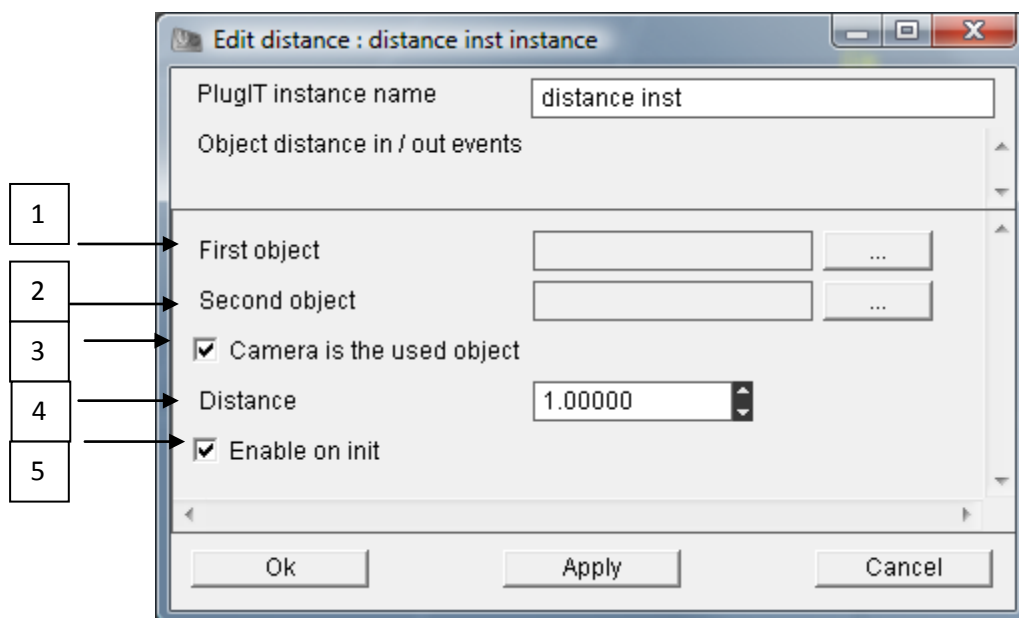
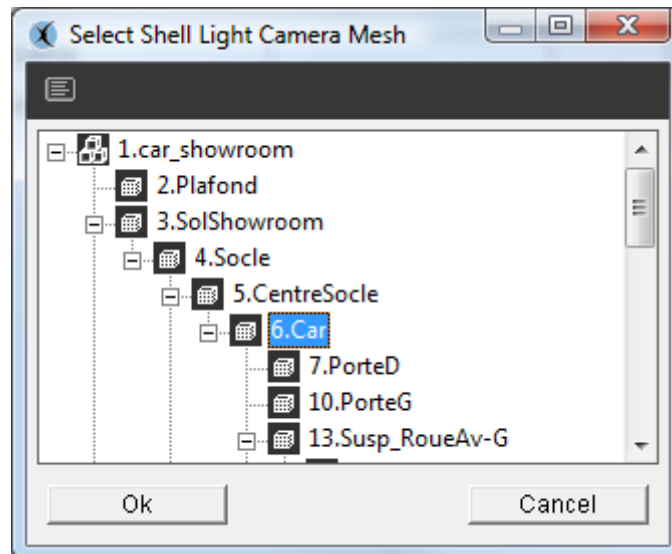


1°/ sélection de l'animation 1 et 2

2°/ Vitesse de la transition

PlugIT Distance

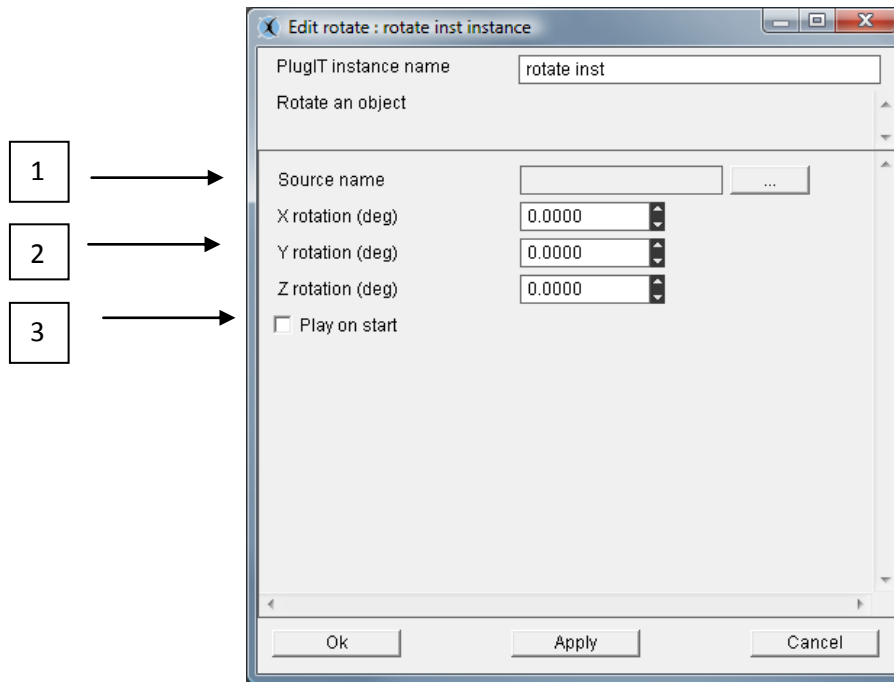
Le PlugIT Distance permet de déclencher un évènement lorsque l'on entre ou sort d'une zone autour d'un objet (radar).



- 1°/Sélectionner l'objet qui servira de source de la zone via le bouton parcourir.
- 2°/Sélectionner l'objet qui servira de delimitation de la zone via le bouton parcourir.
- 3°/Choisissez si la camera est l'objet courant de repère
- 4°/Le paramètre « distance » définit le rayon de la zone autour de l'objet source.
- 5°/Activer au démarrage

PlugIT Rotate

Le PlugIT Rotate permet d'appliquer une rotation continue sur un objet ou un nœud.

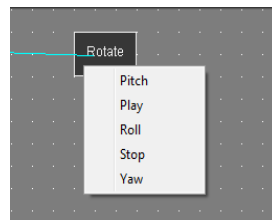


1 °/ Sélection de l'objet source

2 °/Les paramètres « rotation » définit l'angle de rotation appliqué par axe et par image rendue en degré.

3 °/Le paramètre « Play on start » permet de définir si la rotation sera démarrée automatiquement au lancement de l'application.

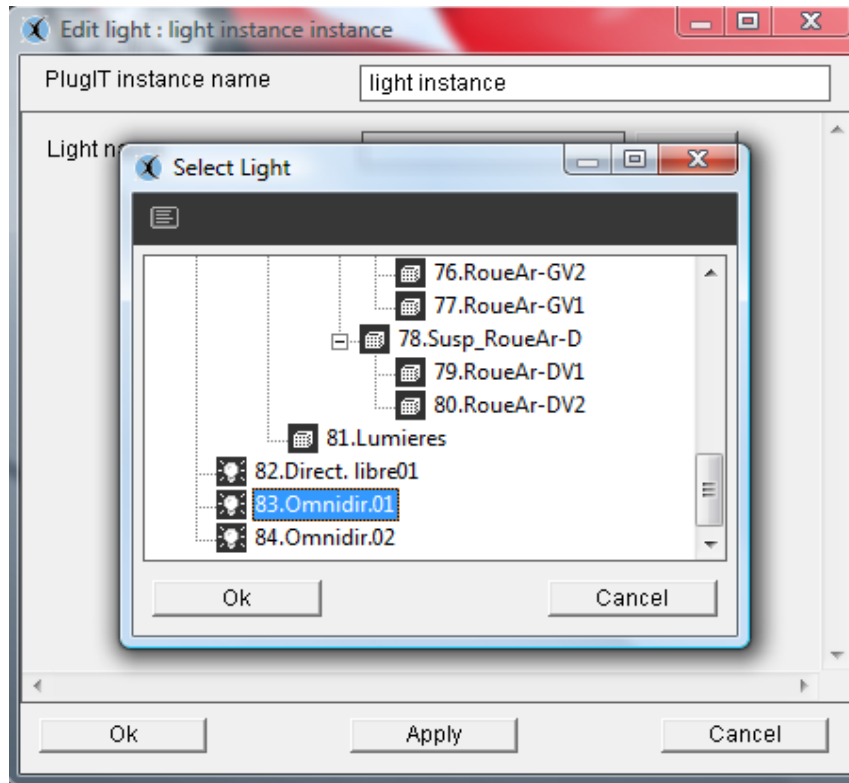
Il est possible de modifier les valeurs de rotation dynamiquement via les actions « Yaw, Pitch , Roll »



PlugIT Light

Le PlugIT Light permet d'activer ou désactiver une lumière en dynamique.

En suivant la même méthode que précédemment, ajouter un PlugIT Light, puis éditer l'instance.

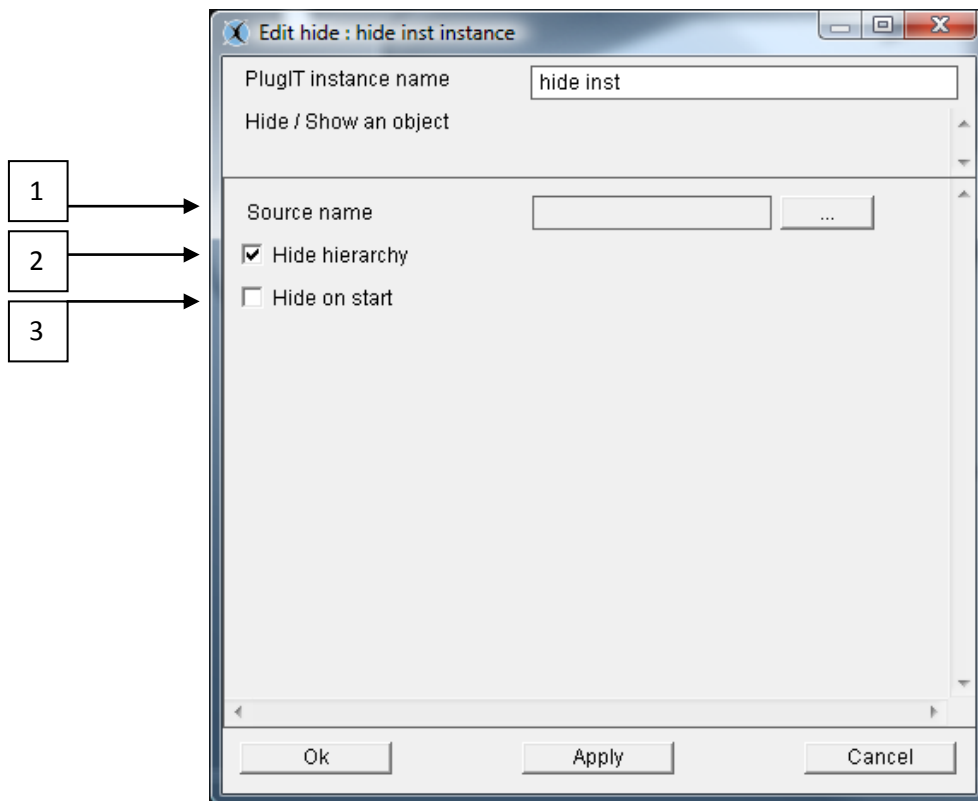


Sélectionner la lumière sur laquelle vous souhaitez agir via le bouton parcourir.

La fonction Light ne possède pas de paramètre particulier, vous pouvez activer ou désactiver la lumière via les actions « On » et « Off » en utilisant les liens.

PlugIT Hide

Le PlugIT Hide permet de cacher ou afficher un élément dans la 3d.



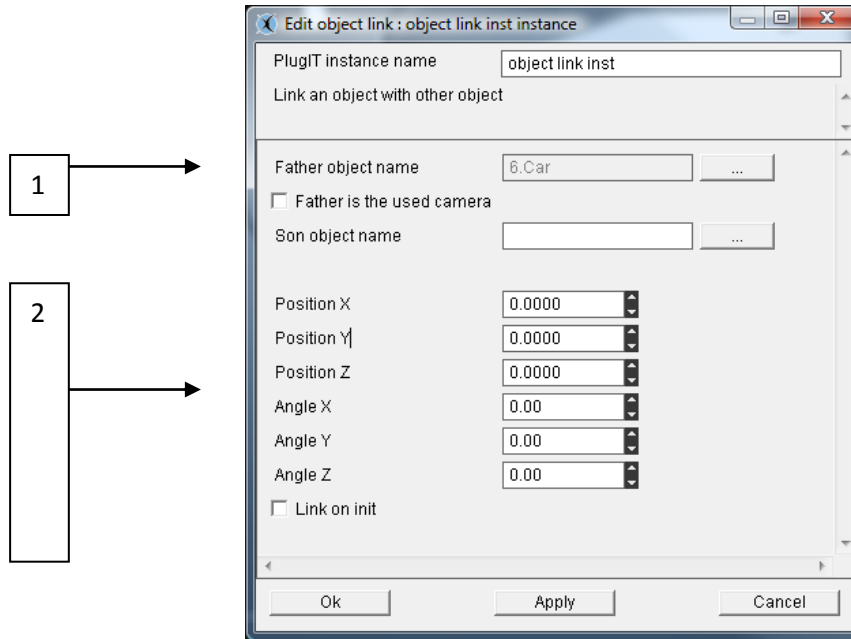
1 °/Sélectionner l'élément sur lequel vous souhaitez agir via le bouton parcourir.

2 °/ Le paramètre « Hide hierarchy » permet de définir si les fils de l'objet seront aussi cachés ou non.

3 °/ Le paramètre « Hide on start » permet de définir si l'élément sera caché automatiquement au lancement de l'application.

PlugIT Object link

Le PlugIT Object link permet de lier un objet à un autre objet ou à la caméra active.



1 °/Sélectionner l'objet père auquel votre autre objet sera lié via le bouton parcourir ou cochez « Father is the used camera » si vous souhaitez lier un objet à la caméra.

2 °/Les paramètres « Position » permet de définir la position par rapport à l'objet père.

Les paramètres « Angle » permet de définir l'orientation par rapport à l'objet père.

Pour utiliser ce PlugIT il vous faudra utiliser un PlugIT Get Object ou Get Camera et faire le lien :

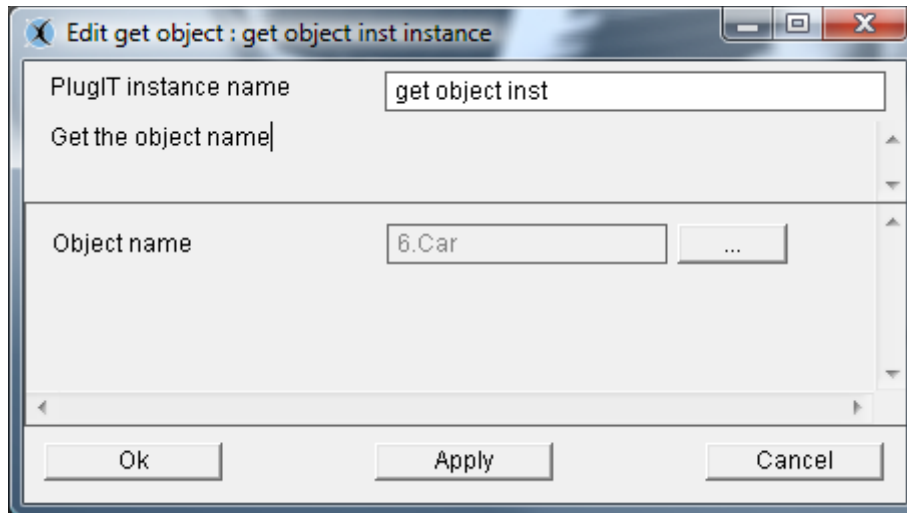
`get object.Object -> object link.Link`

ou passer le nom de l'objet en paramètre du lien.

PlugIT Get Object

Le PlugIT Get Object permet de récupérer le nom d'un objet puis de le renvoyer en paramètre. Il n'est utile que lorsqu'il est utilisé avec un autre PlugIT, par exemple avec le PlugIT Object Link.

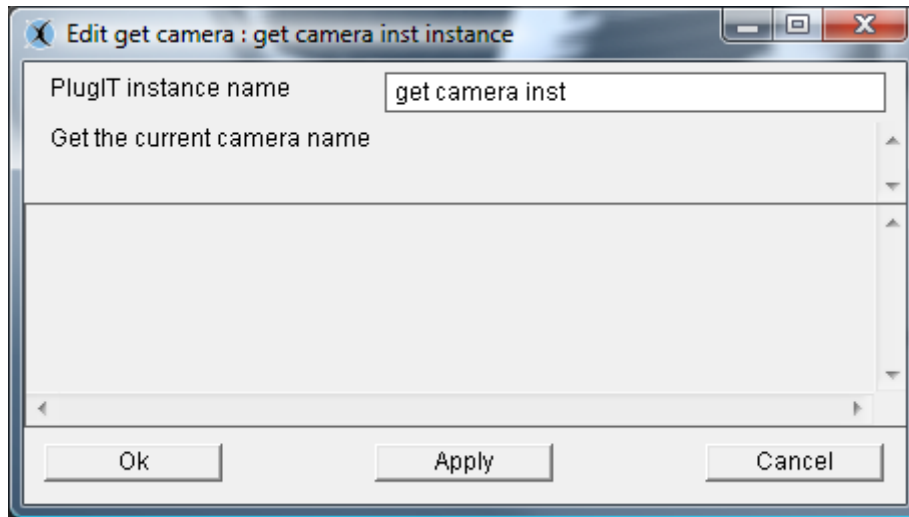
En suivant la même méthode que précédemment, ajouter un PlugIT Get Object, puis éditer l'instance.



Sélectionner l'élément sur lequel vous souhaitez agir via le bouton parcourir.

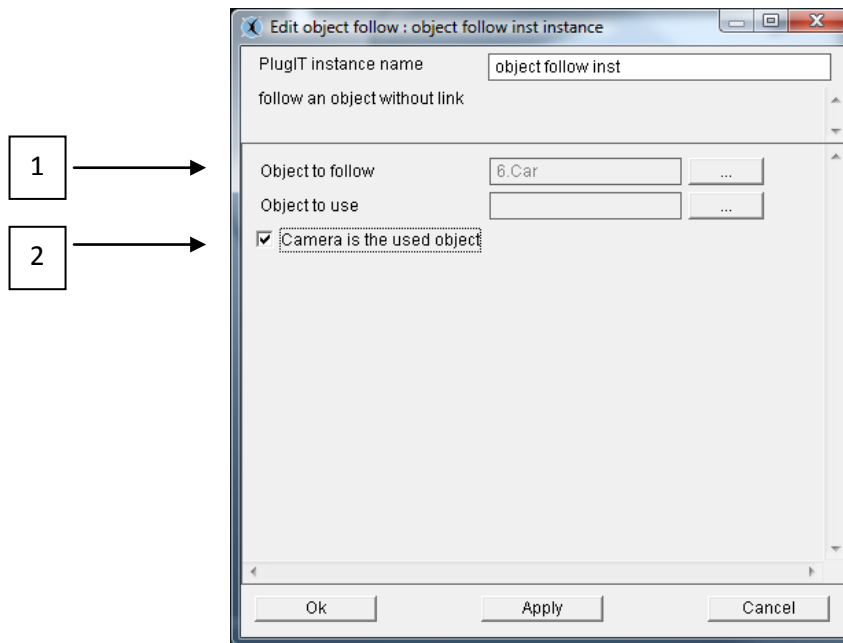
PlugIT Get Camera

Le PlugIT Get Camera permet de récupérer la caméra courante et de la renvoyer en paramètre. Il n'est utile que lorsqu'il est utilisé avec un autre PlugIT, par exemple avec le PlugIT Object Link.



PlugIT objectFollow

Ce plugIT permet à un objet de suivre la trajectoire d'un autre objet sans que ceux-ci ne soit liés par la hiérarchie.

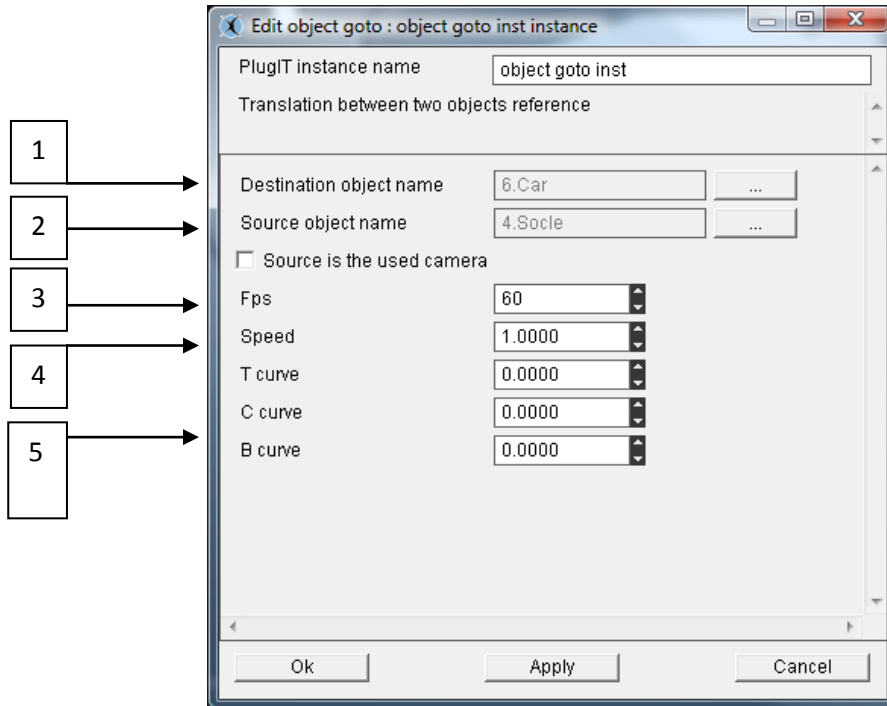


1 °/ L'objet à suivre

2 °/ L'objet suivi, si l'on coche « camera is the used object » alors l'objet qui suit sera la camera courante.

PlugIT objectGoto

Ce plugIT permet automatiquement à un objet d'aller à une position donnée.



1 °/ On renseigne ici l'objet cible qui se déplacera

2 °/ on renseigne ici la position de destination via un objet ou directement en choisissant la position courant de la camera si l'on coche « source is the used camera »

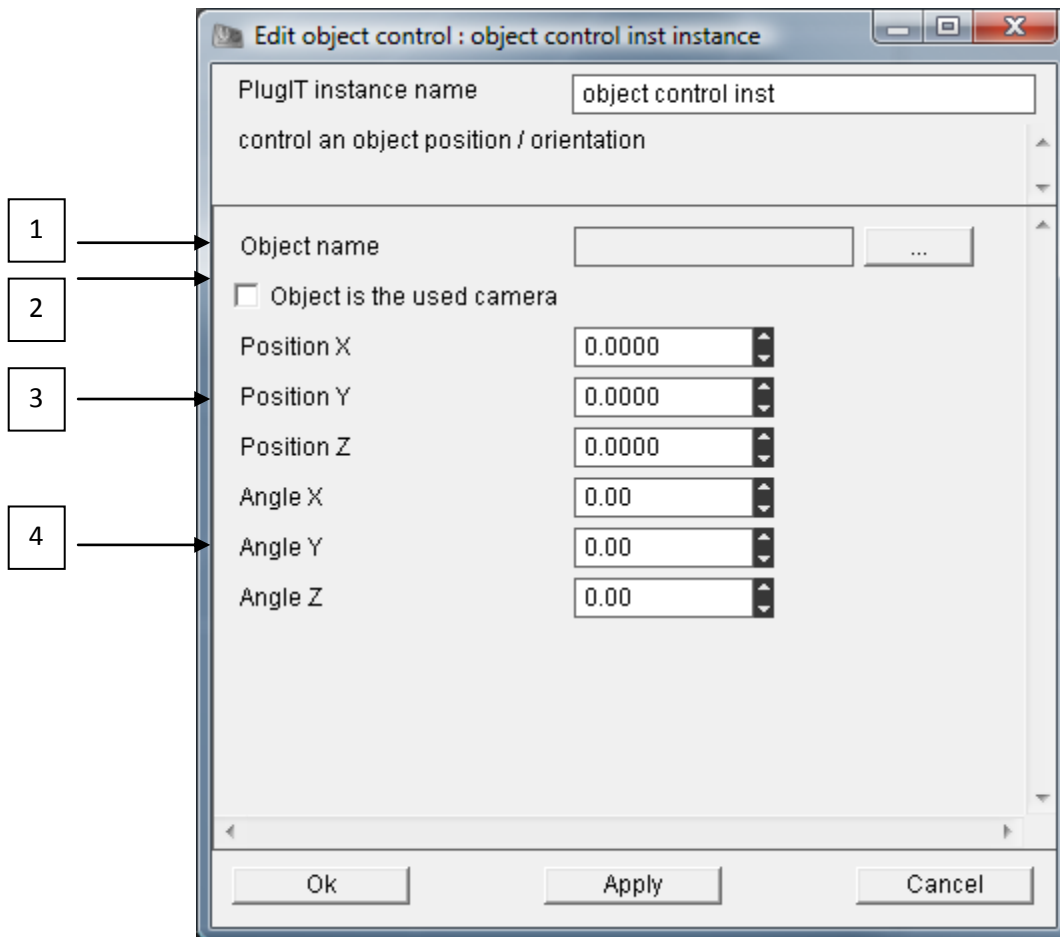
3 °/ Correspond au nombre d'image par seconde utilisé pour l'interpolation linéaire de l'animation

4 °/ Vitesse de déplacement

5 °/ Paramètre TCB pour le calcul des positions intermédiaires par une interpolation de bézier.

PlugIT Object Position

Le plugIT Object position vous permet de contrôler la position et l'orientation d'un objet, cet objet peut également être la caméra courante.



1 °/ On renseigne ici l'objet cible qui sera contrôlé

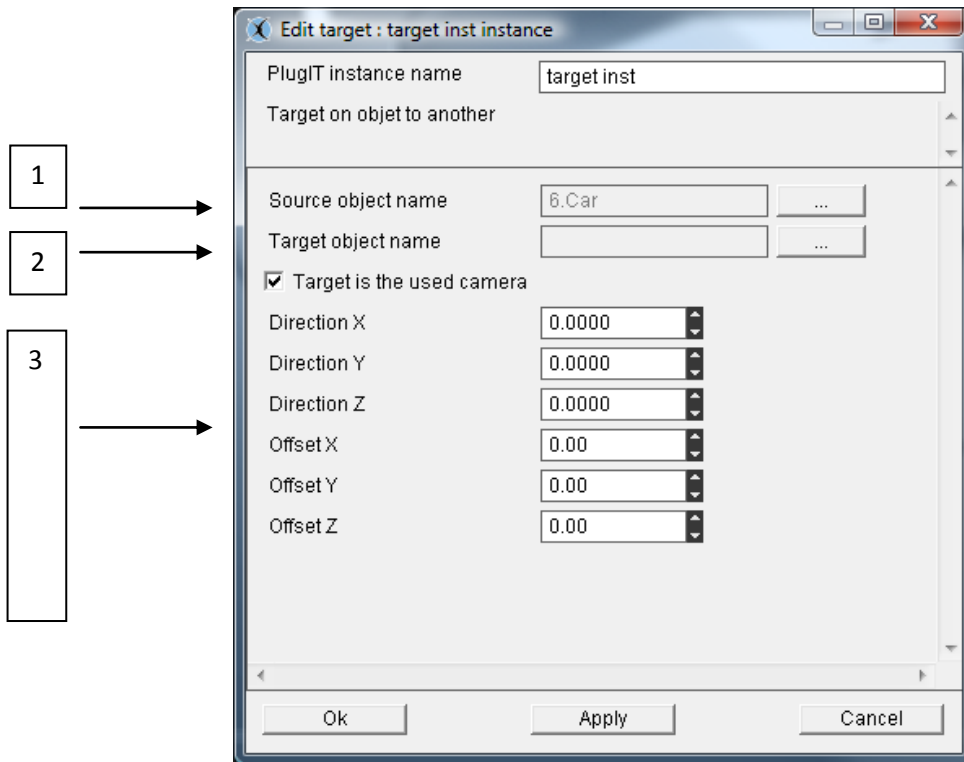
2 °/ on renseigne ici si l'objet à contrôler est la caméra courante

3 °/ On définit ici les positions que l'on souhaite affecter à l'objet

4 °/ On définit ici les angles que l'on souhaite affecter à l'objet (en degrés)

PlugIT target

Ce plugIT permet de fixer l'orientation d'un objet vers un autre.



1 °/ Objet source sur lequel le Target est fixé

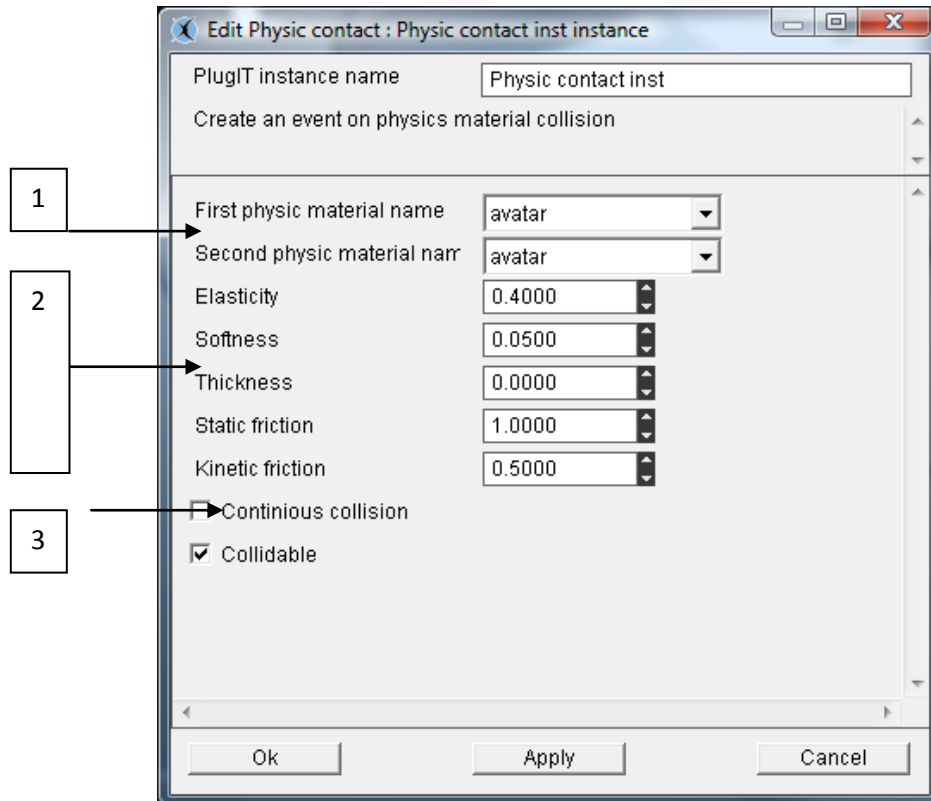
2 °/ Il convient ici de renseigner le nom de l'objet cible ou on peut sélectionner « target is the used camera »

3 °/ On renseigne ici les propriétés de vecteurs pour l'orientation de l'objet source ainsi que les offset qui permettent un décalage par rapport à l'objet cible

Les plugIT « Physics» (cf Utilisation avancées : Moteur Physique)

plugIT PhysicContact

Ce plugIT permet lorsque le moteur physique est activé de paramétrer le contact entre deux matériaux physique (cf : Annexe moteur Physique Newton)



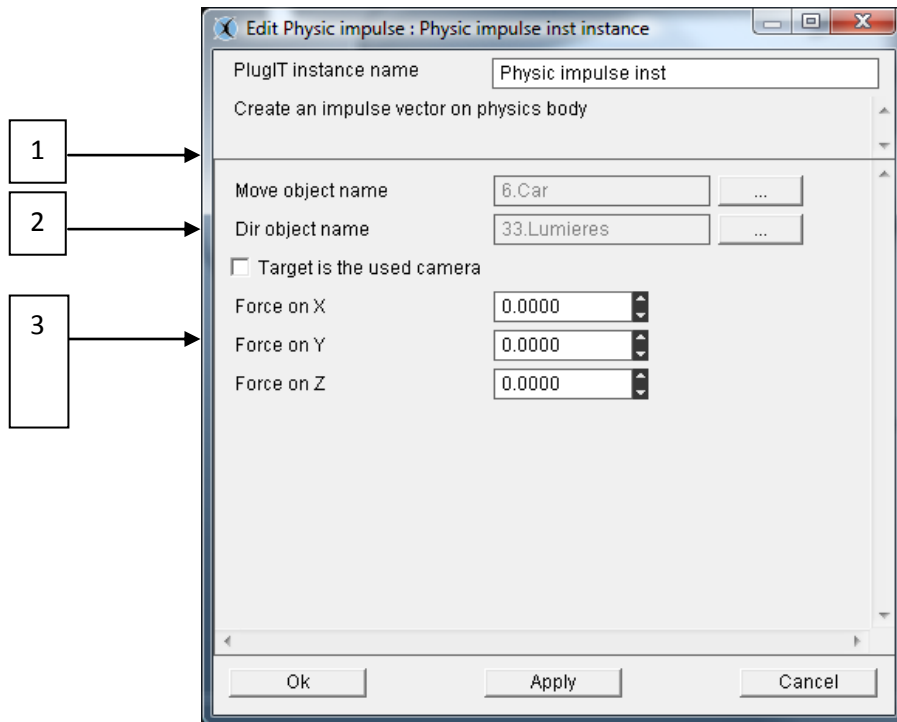
1 °/ On renseigne ici le contact entre deux matériaux par ex : entre le material par défaut appliqué au body et le material correspondant à l'avatar (déplacement de caméra)

2 °/ On paramètre ainsi les réactions physiques lorsque ces deux matériaux se collisionnent : (elasticity, softness, thickness, static friction ,kinetic friction)

3 °/ Options à cocher pour une prise en compte de la collision continue (plus précis mais plus gourmand) et « collidable » pour spécifier que ces matériaux peuvent se collisionner.

plugIT PhysicImpulse

Ce plugIT permet simplement d'ajouter une impulsion (cf : 24. Annexe moteur Physique) sur un body.



1 °/ On renseigne ici le body de l'objet source sur lequel l'impulsion sera appliquée

2 °/ On renseigne ici la direction par l'objet ou l'utilisation de la direction de la camera courante

3 °/ On peut encore paramétrer une impulsion dont la force sera définie par un vecteur Force.

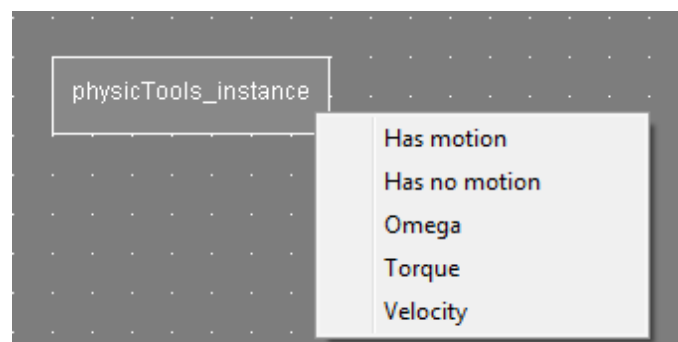
plugIT physicTools

Ce plugIT complet permet d'avoir accès à une toolbox complète afin de paramétrer les réactions physiques sur un objet.



On renseigne le nom de l'objet

Puis :

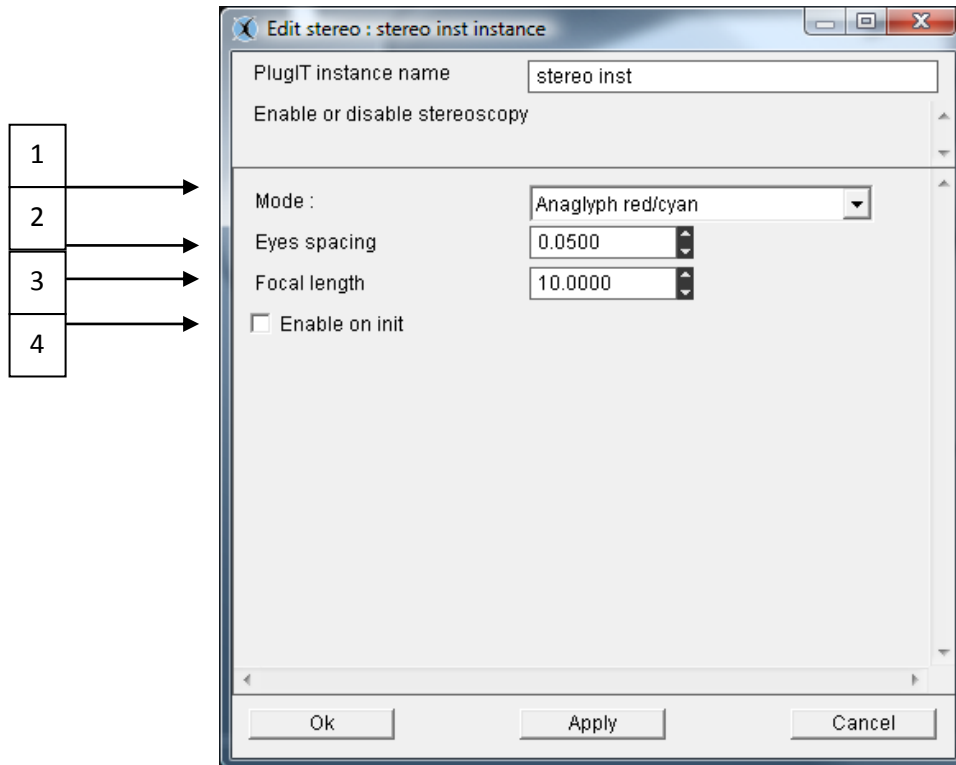


On peut paramétrer le body de l'objet de manière à lui donner des valeurs d'Omega, de Torque et de vélocité (cf : 24. Annexe moteur physique Newton). Ainsi, que récupérer son activité « Has motion » ou sa non activité « Has no motion ».

Les plugIT « Rendering »

plugIT Stereo

Ce plugIT permet d'agir sur le mode de rendu de l'application afin de choisir un mode de stéréoscopie pour la vision en relief.



1°/ Sélection du type de mode stéréo :

Anaglyph red/cyan
 Anaglyph yellow/blue
 Interlaced Horizontal
 Interlaced Vertical
 Interlaced with checkerboard

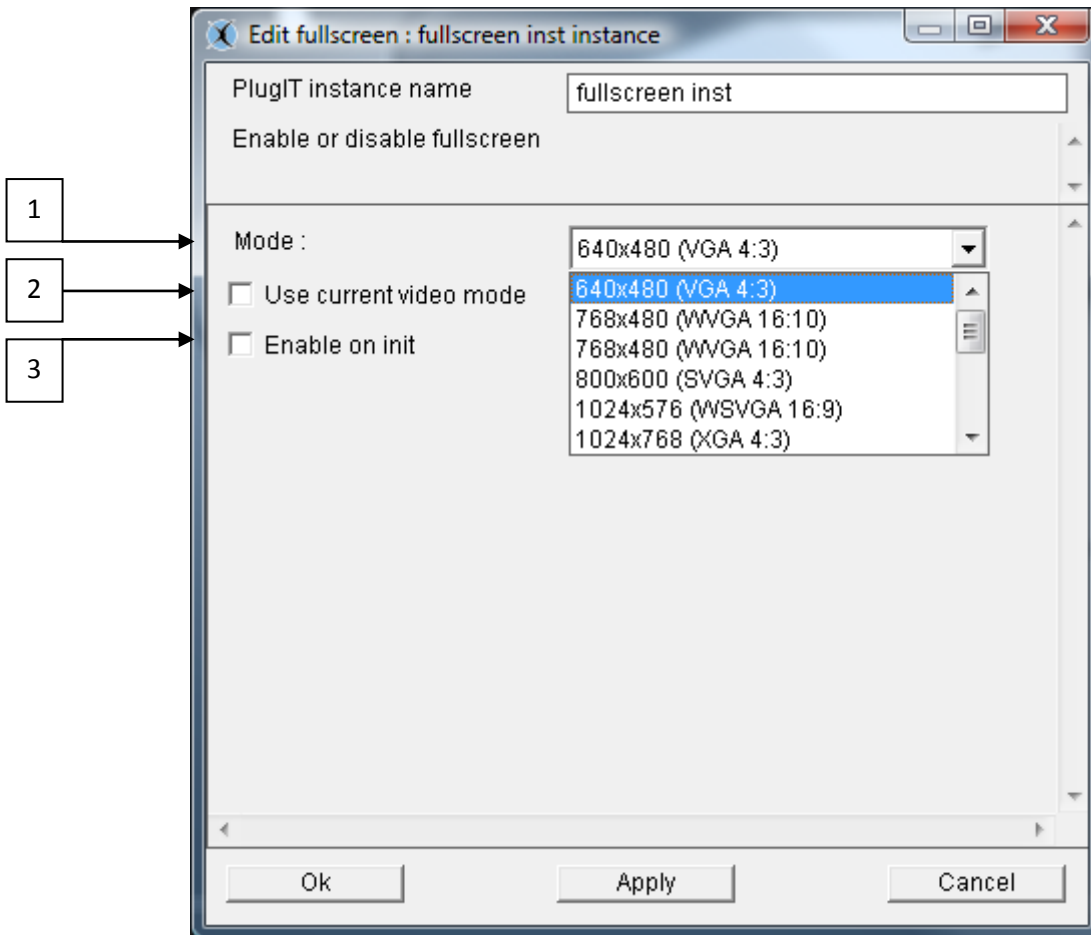
2°/ Définition de la distance entre les yeux

3°/ Définition de la focale

4°/ Activation du mode stéréo au lancement de l'application

PlugIT fullscreen

Ce plugIT permet le lancement automatique du rendu fullscreen (plein écran) au démarrage de l'application.



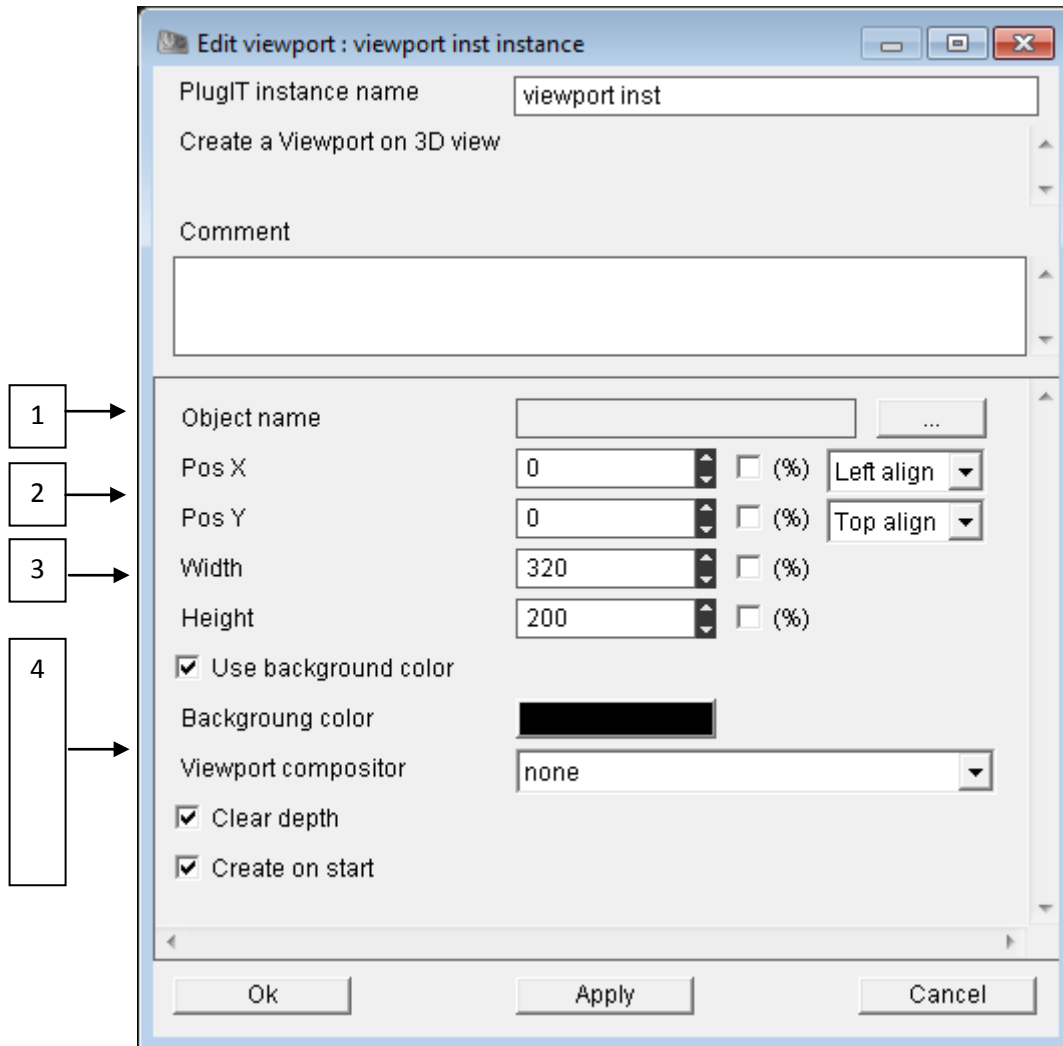
1°/ Sélection d'un mode particulier de Fullscreen pour l'application

2°/ Sélection du mode vidéo courant (sans taille prédéfinie)

3°/ Activation du mode fullscreen au démarrage de l'application

PlugIT Viewport

Le PlugIT viewport permet de créer un viewport, il s'applique sur une caméra.



1°/ Permet de choisir quelle camera sera la caméra courante du viewport

2°/ Position sur la vue 3D du viewport crée

3°/ Taille du viewport

4°/ « use background color » : Permet d'utiliser une couleur de fond sur le viewport si oui une colormap permet de choisir cette couleur

« clear depth » : permet d'utiliser ou non le buffer de profondeur

« clear on start » : Créer le viewport au lancement

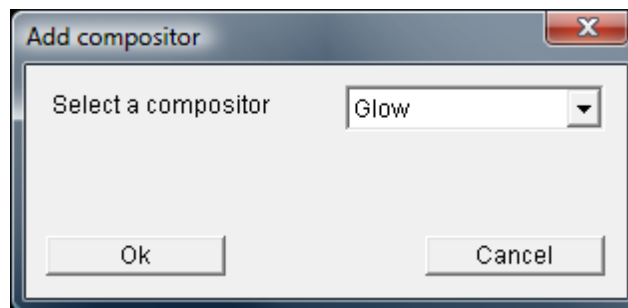
PlugIT Compositor

Le PlugIT Compositor permet d'appliquer un compositor à la scène pour en modifier le rendu.

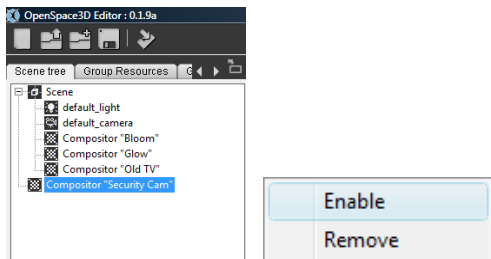
Ajout d'un compositor :

Effectuez d'abord un clic droit sur l'arbre de Scene puis « Add Resources » parcourez votre PartitionLocalUsr pour sélectionner toutes les ressources du compositor : program - script - textures

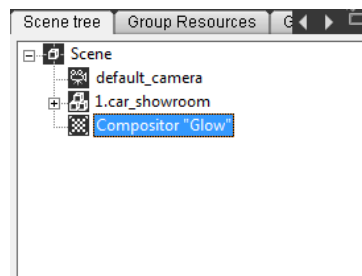
Vous pouvez désormais faire « Add compositor » qui vous proposera le compositor dont vous venez d'ajouter les ressources



Validez : Votre compositor Glow a été ajouté à la scène et activez le en effectuant un clic droit :

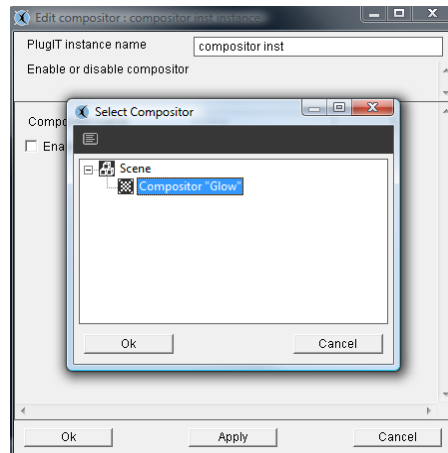


(Ne l'activez pas si vous souhaitez activer le compositor avec un PlugIT).

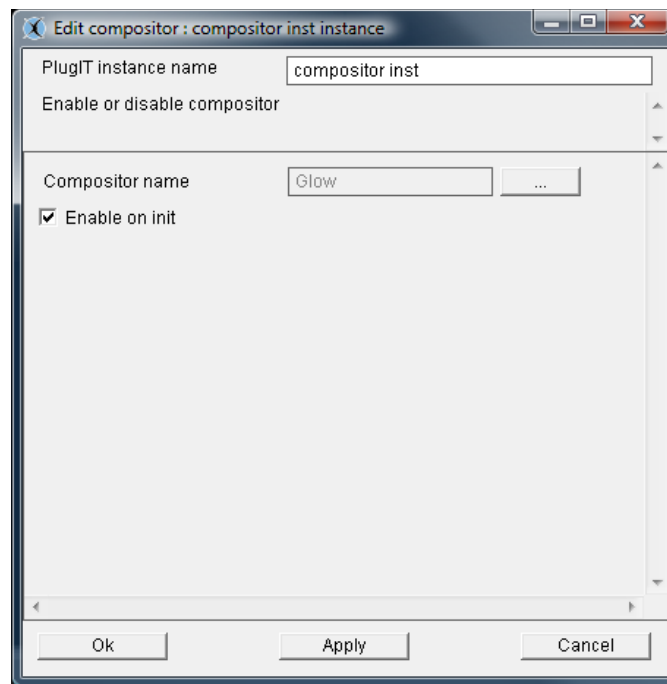


Dans le groupe Scene, allez ajouter le rendering PlugIT : Compositor

Et allez sélectionner le compositor



Cochez la case « Enable on Init » (ne la cochez pas si vous souhaitez activer le compositor avec un autre PlugIT).



Debug et Log

Lors du développement d'une application sous OpenSpace3D, il peut s'avérer très utile de pouvoir connaître pas à pas l'avancement de nos fonctionnalités.

Ainsi, ceci peut aider à déterminer un dysfonctionnement au niveau de l'application.

Pour cela, il existe des outils sous OpenSpace3D se situant en bas à droite de l'éditeur et correspondant aux 2 icônes :

Fenêtre de log :

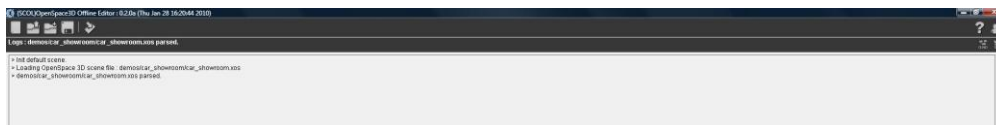


Activation du debug de liens :

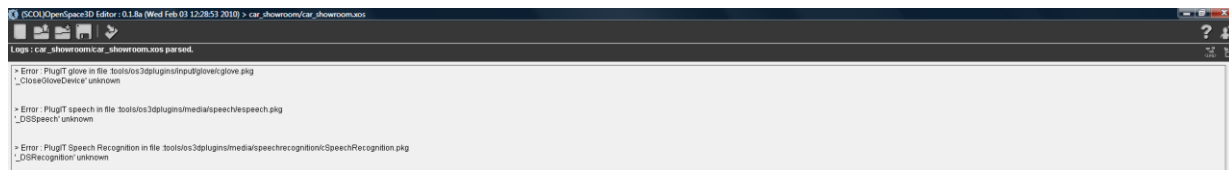


La fenêtre de Log

Cette fenêtre affiche tous les événements relatifs à votre développement d'application (chargement d'un objet, enregistrements et export de votre scène...)



Bien que chargée dans l'éditeur OpenSpace3D, il se peut que certains des PlugITs que vous avez développés n'aient pas été initialisés correctement, dans ce cas, la fenêtre de log vous en informe :



Vous pouvez recharger un PlugIT et donc le réinitialiser dynamiquement en utilisant Clic droit + Refresh dans la zone d'édition de PlugITs

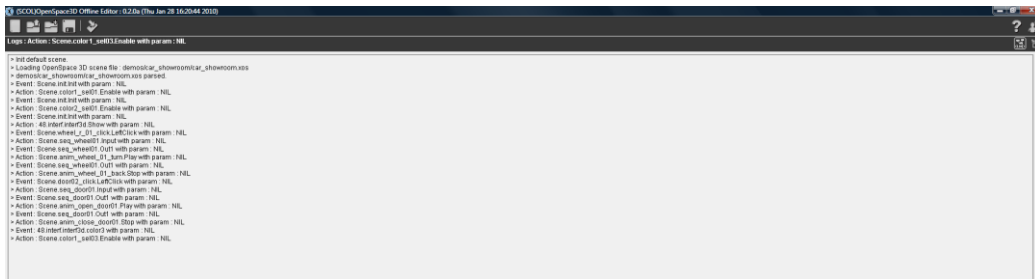
L'activation du debug pour le suivi des liens

En cochant cette option, cela vous permet de suivre tous les actions/événements au niveau de votre applicatif.

Ces informations permettront de suivre en temps réel le développement de votre application.

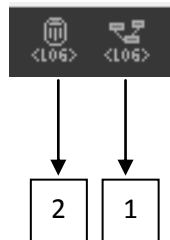
Attention, ce mode « debug » peut ralentir l'exécution des plugIT mais est très pratique pour vous permettre de déterminer la cause d'un dysfonctionnement lors de votre création de scène interactive.

Ces informations s'afficheront dans la fenêtre de Log :



1°/ activer/desactiver les logs

2°/ vider la fenêtre de log





Utilisation avancée et Définitions

Ressources Graphiques (3D)

En 3D une Ressource graphique est un élément contenant des données permettant le rendu d'une scène ou d'un objet.

Ces ressources peuvent être :

Mesh : C'est le format du modèle 3D écrit sous la forme de polygones ou points qui définissent son enveloppe visuelle

Matériau : C'est l'ensemble des propriétés concernant l'aspect visuel d'un objet (réaction à la lumière, couleur, apparence)

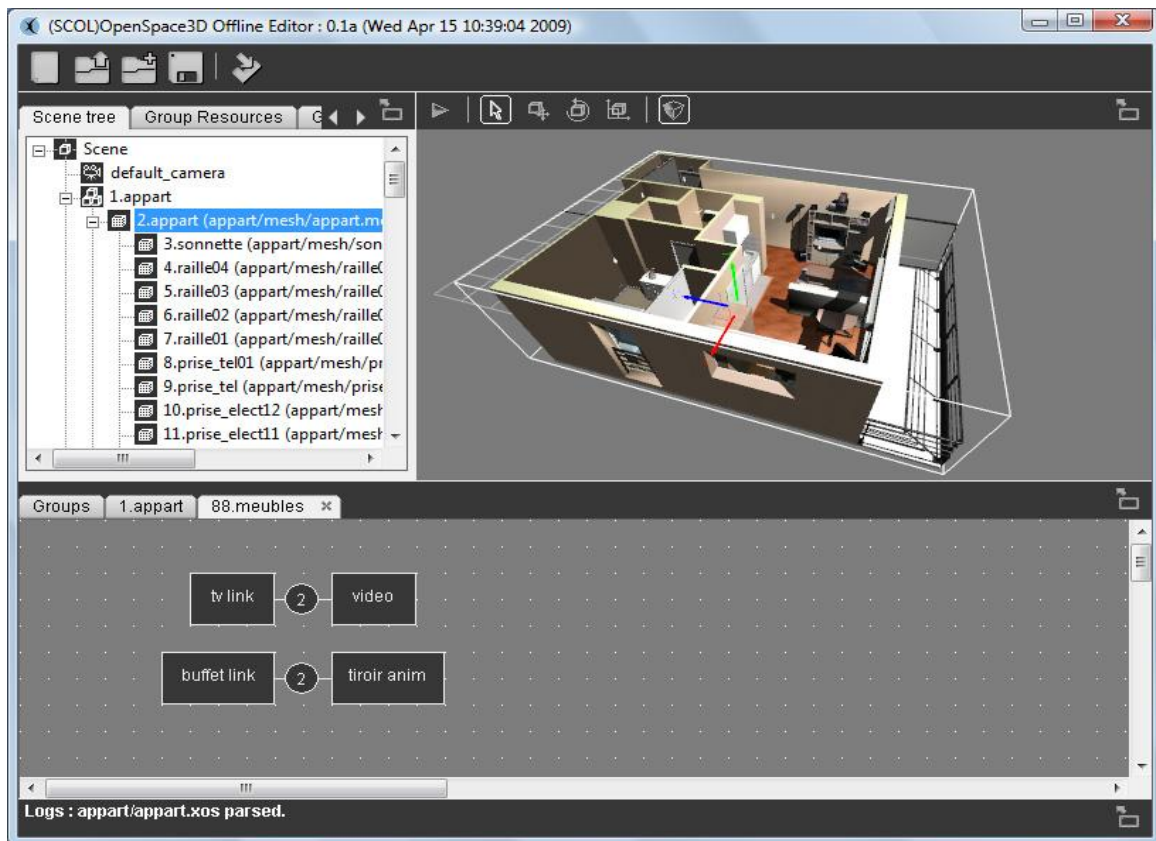
Le photoréalisme d'un objet 3D est permis grâce à l'utilisation de texture

Texture : C'est un ensemble de pixels 2D que l'on va appliquer sur une surface ou un volume 3D. En simplifiant, on peut l'assimiler à un papier peint très plastique et déformable que l'on applique en 3D en spécifiant la transformation géométrique que subit chaque pixel du papier pour s'appliquer sur l'élément 3D. Le pixel ainsi manipulé en 3D est appelé texel.

Une texture peut être définie de façon analytique par un algorithme (texture procédurale) ou être constituée d'un tableau de pixels (une image bitmap par exemple)

- Les groupes de ressources

Dans le moteur 3D SCOL, un groupe de ressources correspond à un ensemble de ressources graphiques permettant l'intégration dans une scène d'un groupe d'objets.

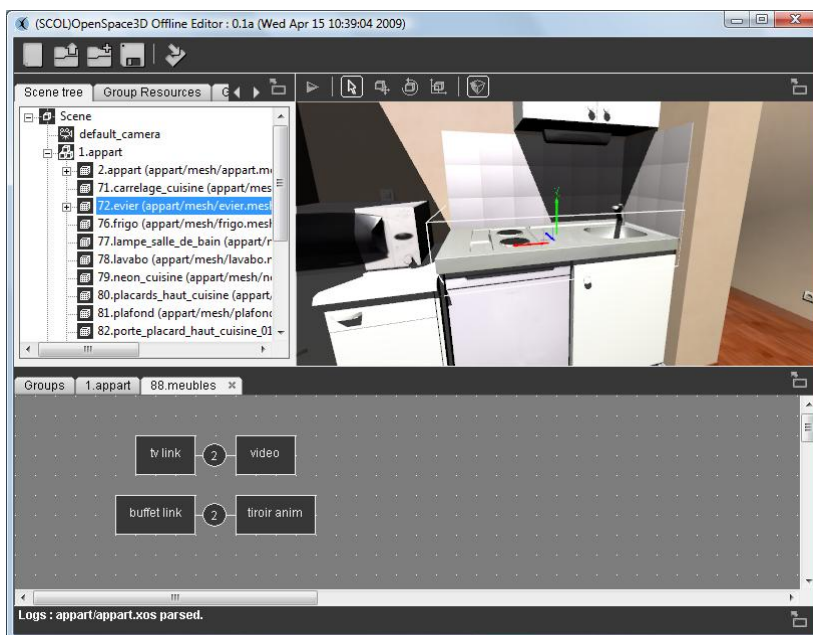
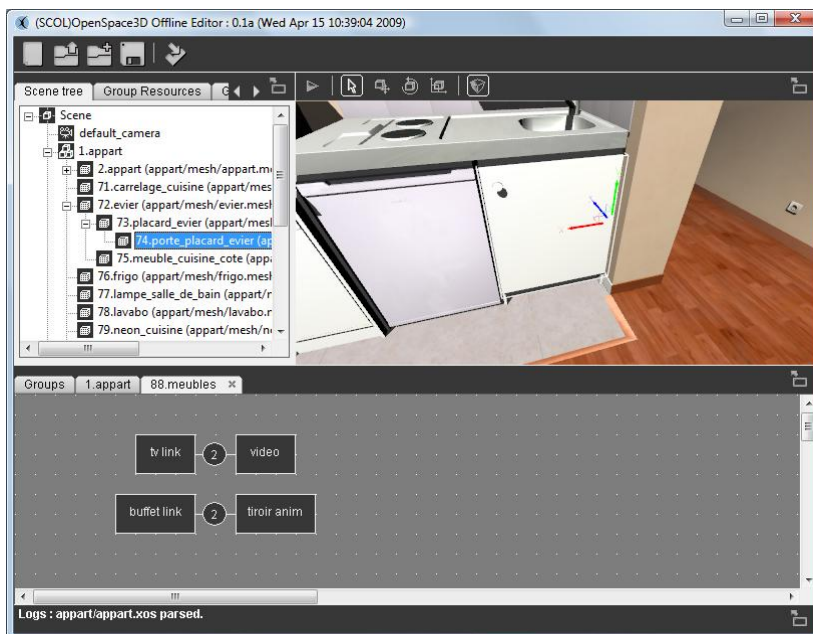


Ici l'1.appart est un groupe de ressources contenant toutes les ressources graphiques permettant l'affichage de l'appartement

- Groupes de mesh

Ensemble de mesh constituant un objet

Exemple :



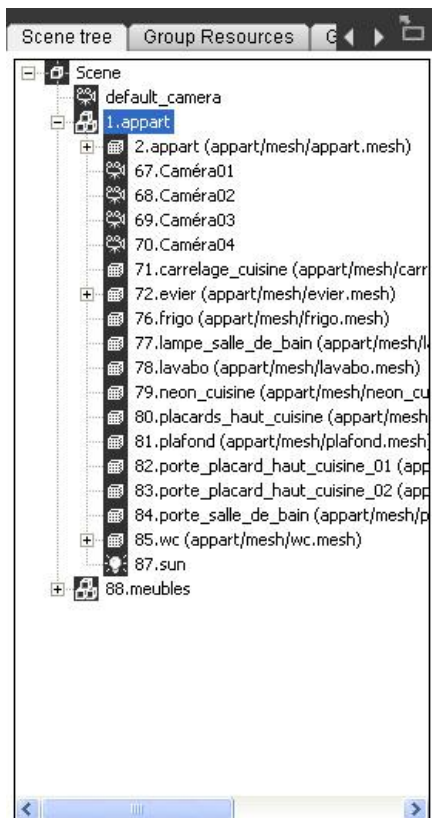
Sur ces images, 72.évier est constitué des mesh : 73.porte_placard_evier , 74.porte_placard_evier et 75.meuble_cuisine_cote.

- Arbre de scène :

Un arbre de scène est une structure de données couramment utilisés des applications modernes et les jeux d'ordinateur. La scène graphique est une structure qui organise la logique et la représentation spatiale de la scène graphique.

Un arbre de scène est une collection de nœuds d'un graphe ou d'arbre de structure. Un nœud a de nombreux enfants, mais un seul parent.

Si une transformation est appliquée au parent alors elle se répercute sur l'ensemble de la hiérarchie. Exemple :



- Nœud de Scène

Un nœud de scène est un élément de la hiérarchie d'un arbre de scene il peut-être un groupe, un objet, une lumière, une caméra ou encore la scène elle-même est le nœud principal de l'arbre.

- Un dummy

C'est un nœud de scène vierge c'est-à-dire un repère dans la 3D sur lequel il est parfois pratique de placer d'autres nœuds afin de contrôler en dynamique le graphe de scène

Group Resources

L'interface de cet outil est la suivante :



Ici est sélectionné le groupe :

 1.appart Dans le Scene Tree

Ce groupe est la représentation d'un ensemble de ressources graphiques qui le compose

Dans l'onglet « Group Resources » on retrouvera ainsi l'ensemble de ses ressources qui pourront ensuite être manipulées.

Group Meshes

 2.appart (appart/mesh/appart.mesh)



Ici est sélectionné le groupe :

 1.appart Dans le Scene Tree

Ce groupe est la représentation d'un ensemble de ressources graphiques qui le composent

Dans l'onglet « Group Resources » on retrouvera ainsi l'ensemble de ses ressources qui pourront ensuite être manipulées.

Resources Directories



Comme expliqué précédemment, le moteur 3D doit connaître tous les chemins relatifs à la partition SCOL pour pouvoir charger les différentes ressources nécessaires à l'affichage dans la scène des différents objets 3D composant le groupe.

Importer une ressources Ogre (shaders, texture, material, mesh)

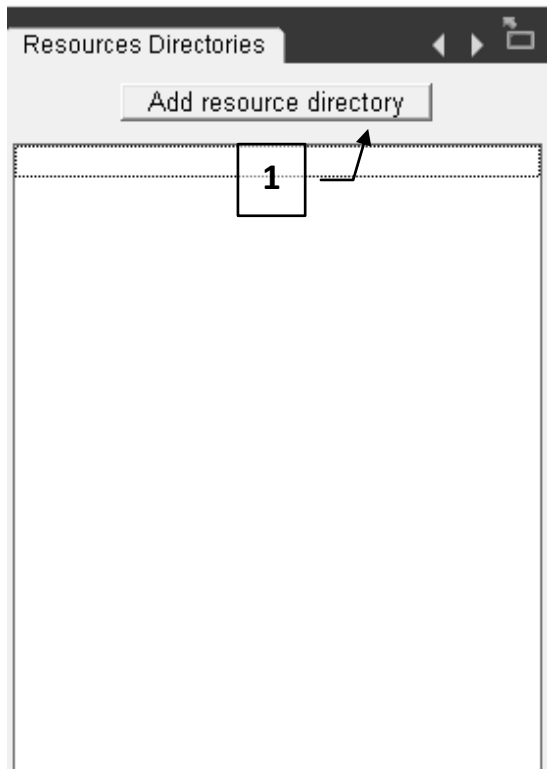
Il peut arriver dans la construction de la scène 3D de vouloir importer directement un objet sans passer par un XML de scène Ogre.

Ainsi, pour cela il faut savoir que Ogre Max permet d'exporter des fichiers .mesh et des ressources de matériaux (.material)

Pour exporter ces données dans la scène 3D ou dans un groupe on utilise le menu par clic

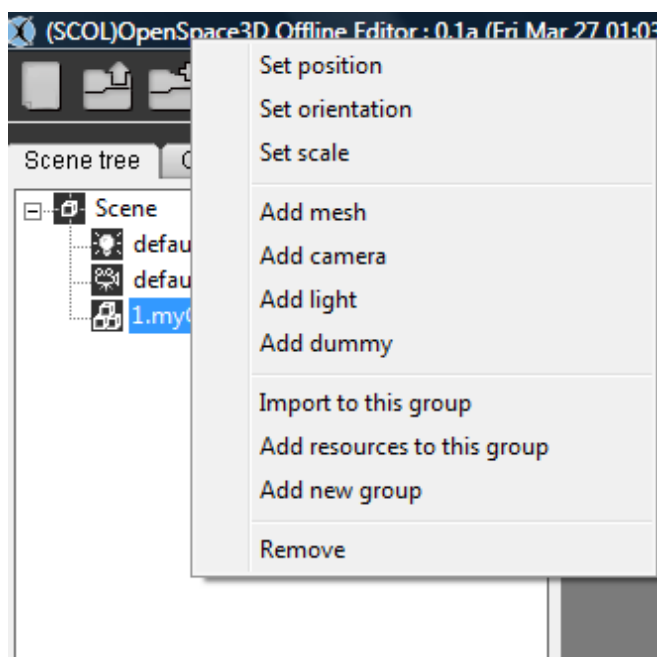
droit dans le « l'arbre de scène » sur la scène ou le group dans lequel on souhaite ajouter l'objet. La procédure d'ajout d'une ressource comporte plusieurs étapes nécessaires et qui doivent être suivies pour le bon déroulement de l'import de l'objet

1 / Le moteur 3D Ogre doit connaître ou se trouve les ressources qu'il devra aller charger

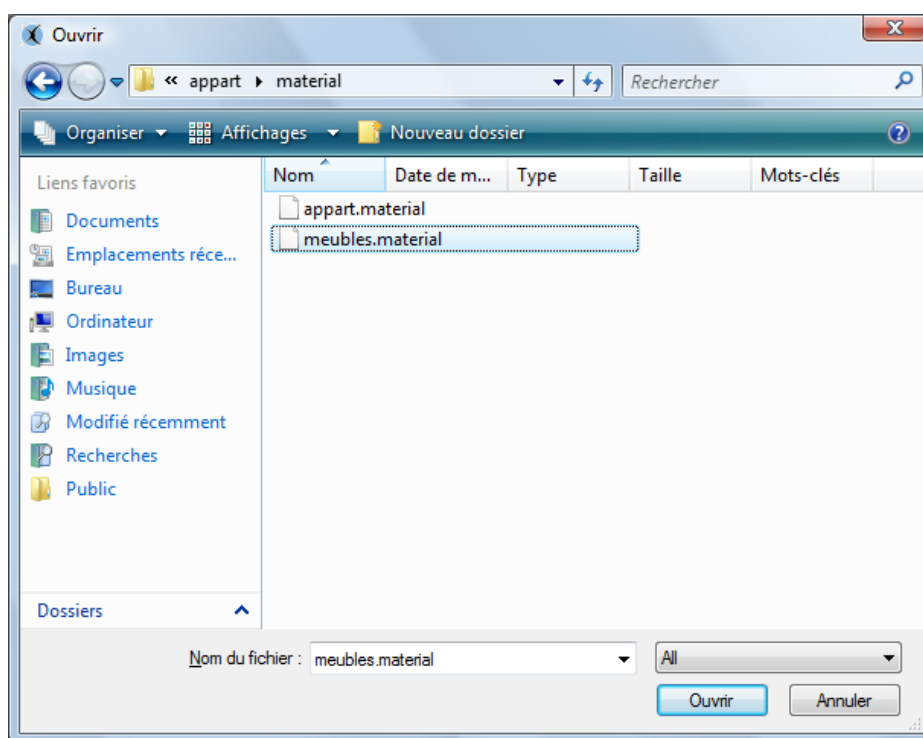


Ainsi, dans l'onglet, Resources Directories il est nécessaire d'ajouter TOUS les chemins (relatifs à la partition SCOL) ou se trouveront les différentes ressources graphiques (.program, .material, textures...)

2 / L'étape suivante consiste à ajouter les ressources graphiques (.program et .material)

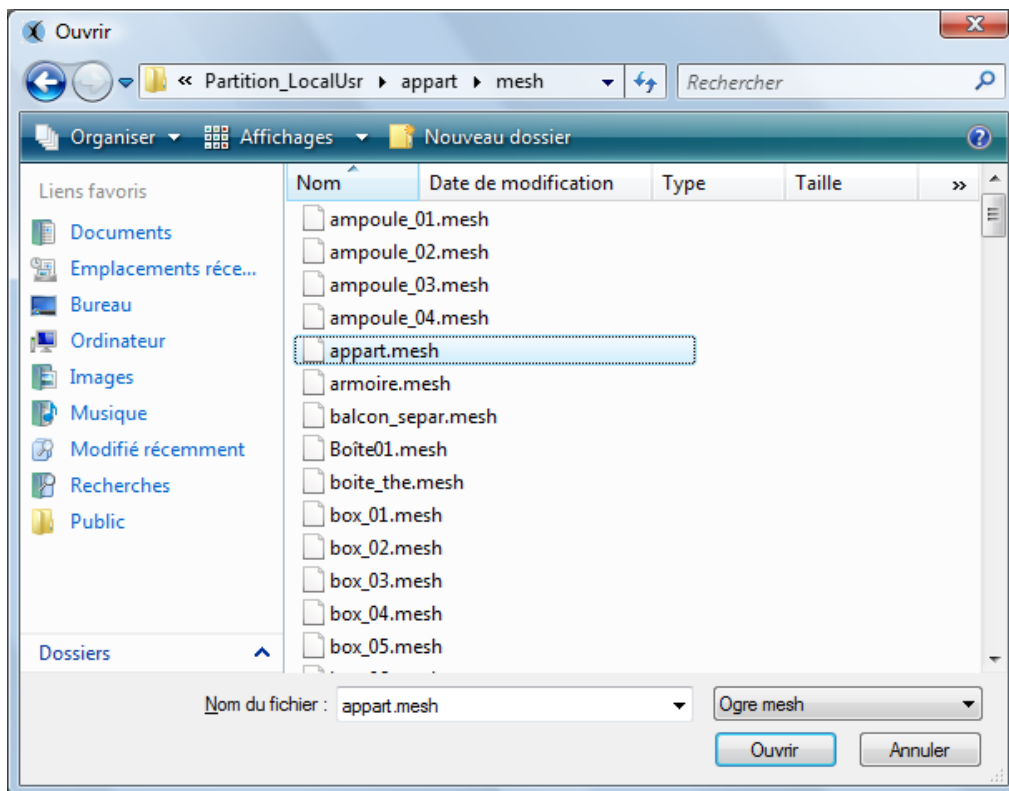
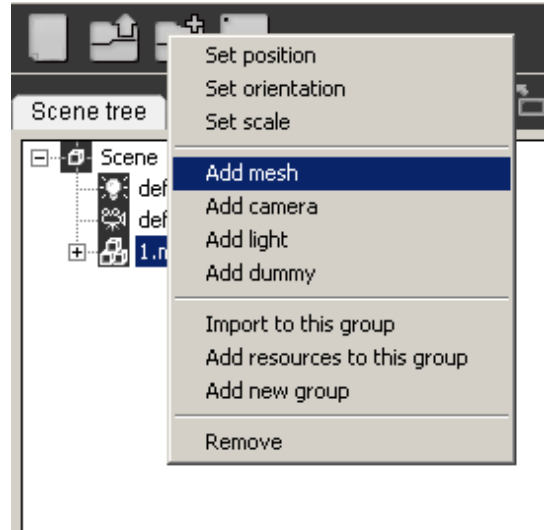


Cet outil ouvre la fenêtre de dialogue :



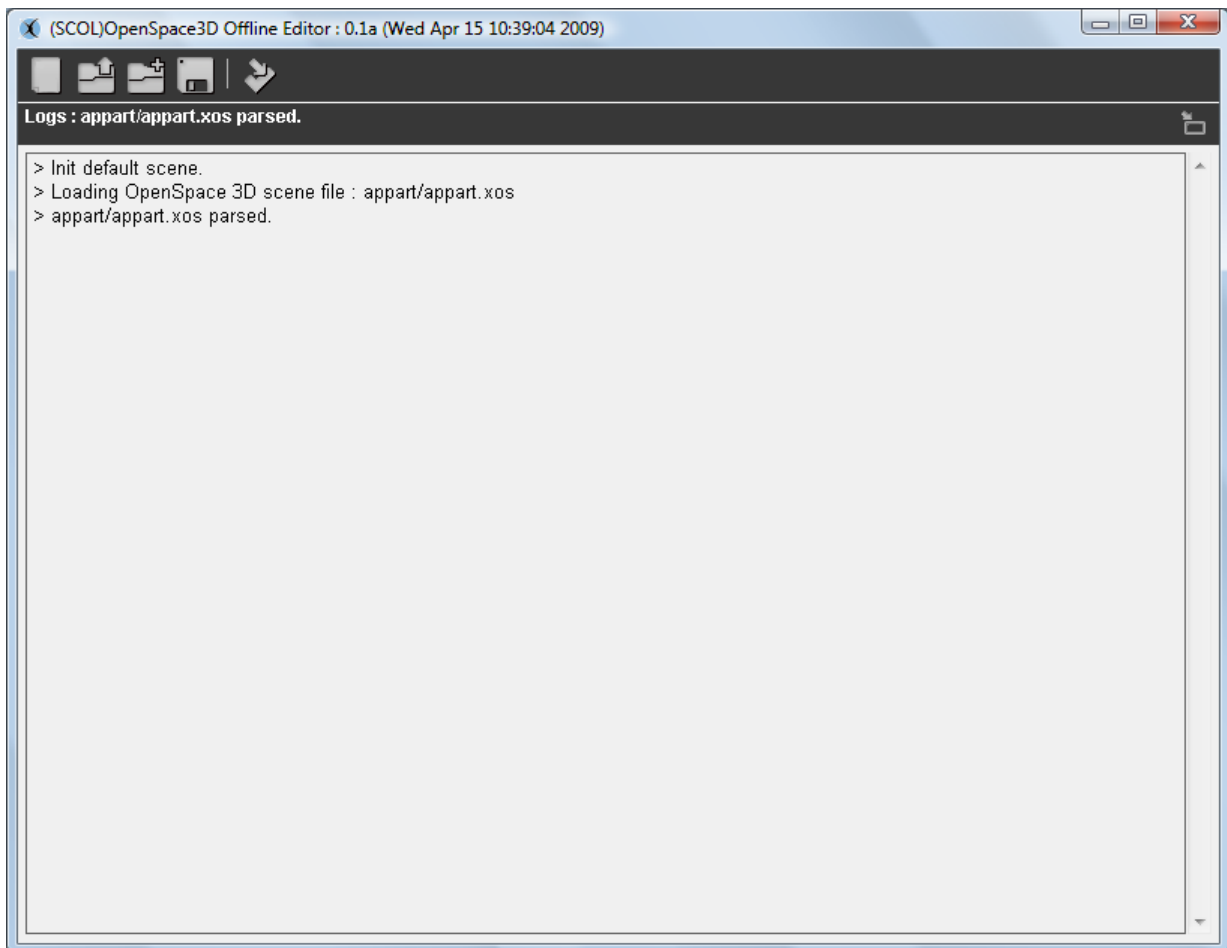
C'est ici que nous devons aller charger les différentes ressources (.material, .program)

3 / La dernière étape consiste à ajouter tout simplement le fichier .mesh correspondant à l'objet que l'on souhaite charger dans la scène ou le groupe



Les ressources graphiques ayant été au préalable chargées avec les étapes 1 et 2 l'objet 3D s'affichera au centre de la scène avec toutes ses données material, program et textures.

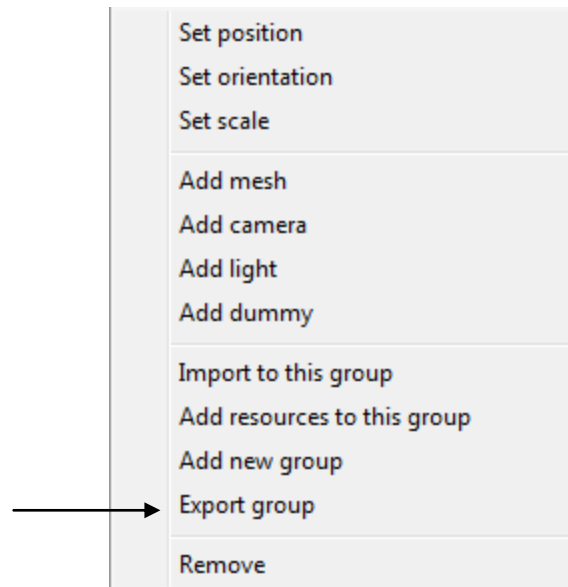
N.B : Il est possible en affichant l'onglet log de voir si aucun message d'erreur n'a été émis



N.B 2 : Les onglets « Group ressources » et « mesh ressources » de la zone de hiérarchie et d'information 3D permettent de vérifier si nos ressources ont bien été chargées dans le groupe souhaité.

Exporter un groupe

Cette fonctionnalité s'avère importante lorsque l'on souhaite développer un projet conséquent faisant intervenir plusieurs personnes. Ainsi, il est possible d'exporter un group c'est-à-dire l'ensemble des objets 3D et des ressources le constituant mais également l'ensemble des plugIT et fonctionnalités de ce groupe.



Ainsi, à partir d'une autre scène on pourra importer le groupe directement via l'ajout du .xos correspondant à ce groupe. (cf. Import to Scene)

Définitions et utilisation avancée des paramètres

- Lumière

En 3D, une lumière sert à illuminer la scène.

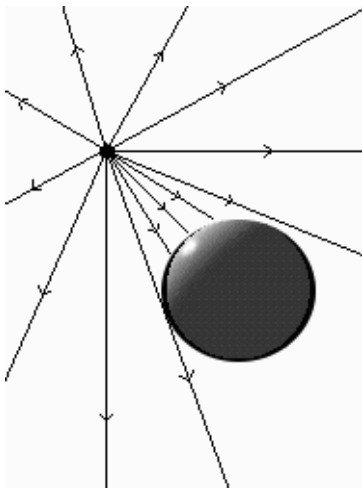
Une lumière ambiante et unique est définie au niveau de la scène (c'est une intensité globale lumineuse)

Ensuite d'autres type de lumières existent et auront un comportement différent sur le rendu de la scène :

- Les lumières omnidirectionnelles ou ponctuelles :

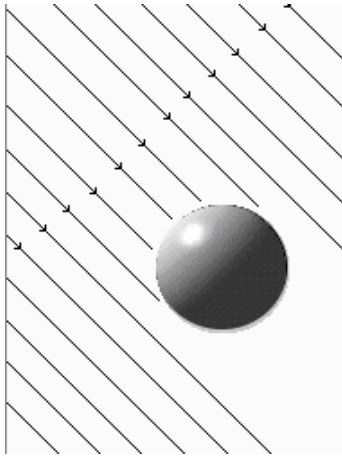
Ces lumières sont représentables par un point qui permet à l'intensité lumineuse de diffuser dans toutes les directions.

Leur représentation dans le monde physique peut-être l'exemple de l'ampoule.



- Les Lumières directionnelles :

Ces lumières sont représentables par un vecteur définissant leur direction, ainsi une lumière directionnelle est souvent assimilée au soleil car elle correspond à une source lumineuse située à l'infini dont l'intensité nous parvient avec une direction donnée.



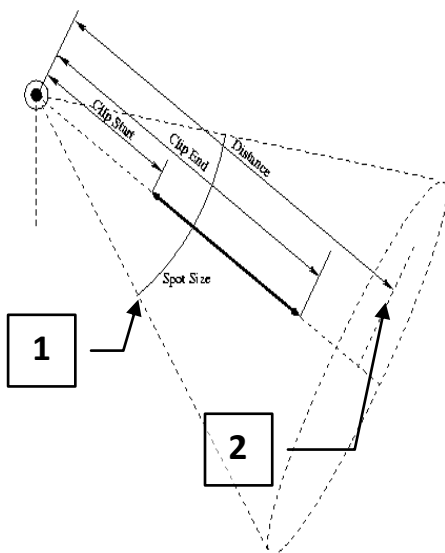
- Les Lumières Spot :

Comme leur nom l'indique ces lumières peuvent être assimilées à des ampoules « Spot ».

Ainsi plusieurs paramètres les caractérisent :

1 Leur ouverture ou spot size qui correspond à un angle de diffusion (le type de lumière omnidirectionnelle est un cas particulier des spots avec une ouverture de 360°)

2 Leur far clip ou Range est la distance d'éclairément de la lumière. Cette valeur permet de définir la distance à partir de laquelle la lumière n'éclaire plus.



L'atténuation

L'atténuation des lumières en 3D est la manière dont l'intensité lumineuse va décroître en fonction de la distance.

Cette donnée est définie suivant la formule suivante :

$$attenuation = \frac{1}{k_c + k_l \times d + k_q \times d^2}$$

Avec d : distance à la lumière

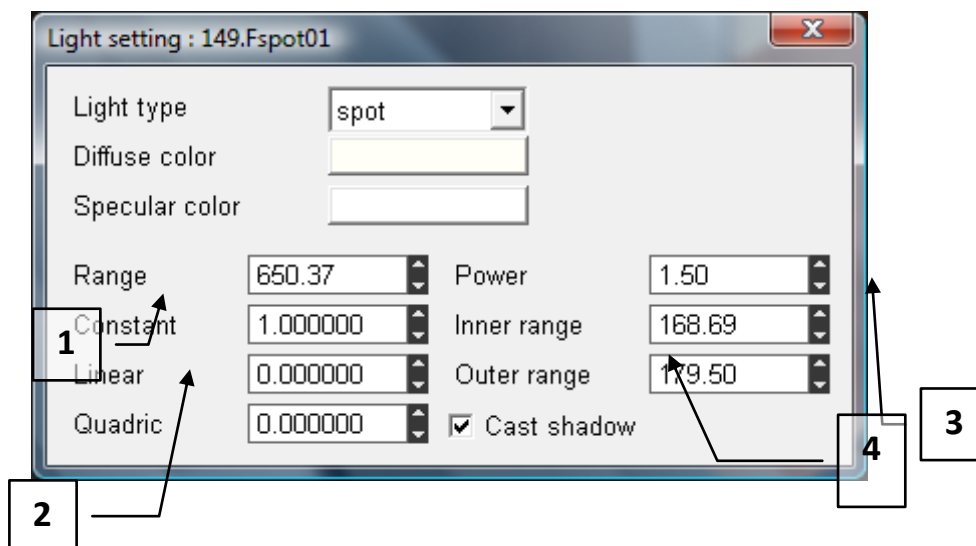
kc : Coefficient atténuation constante

kl : Coefficient atténuation linéaire

kq : Coefficient Atténuation quadratique

Par conséquent, suivant la variation de la distance les coefficients seront plus ou moins influents (sauf pour kc qui est une atténuation constante qui définit par conséquent la puissance de l'intensité lumineuse)

Lien avec l'Edit Setting des lumières dans l'OS3DEditor :



1 : Distance jusqu'à laquelle la lumière éclaire

2 : Ces sont les coefficients d'atténuations de la lumière

3 : Cette variable n'est active que dans le cas particulier ou un HDR est activer sur la scène

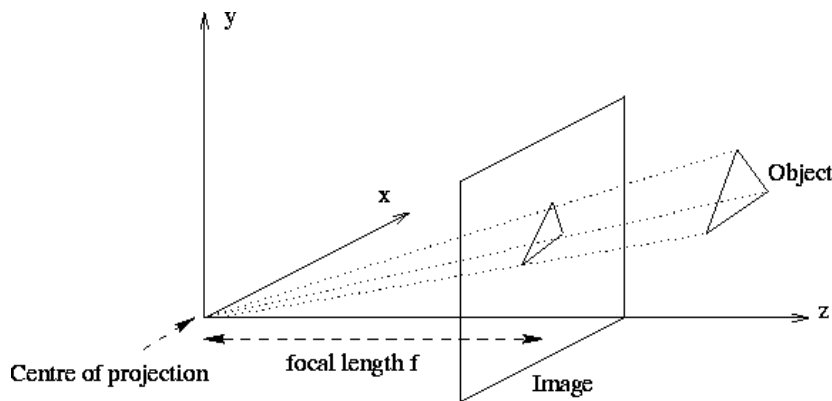
4: Inner Range et Outer Range définissent l'ouverture angulaire des spots

- Les Cameras

En 3D, une camera est une représentation virtuelle d'une caméra réelle avec des paramètres identiques à une caméra physique.

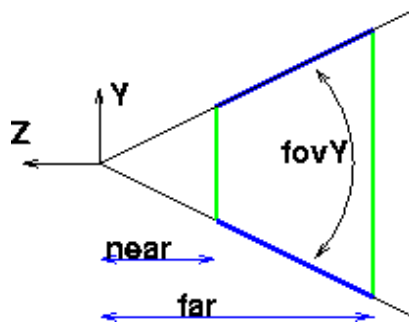
Ainsi, une caméra 3D est l'œil de l'utilisateur permettant l'observation et la projection du monde 3D sur un plan 2D représentant l'image observée.

La représentation usuelle s'appelle le « pin hole camera model »



Focal length : La distance focale d'un objet est définie par la mesure de sa puissance de convergence, elle est liée au système optique car elle définit la puissance de convergence ou de divergence de la lentille du modèle de caméra.

Ainsi, la variation de la distance focale provoque une déformation de l'environnement 3D.



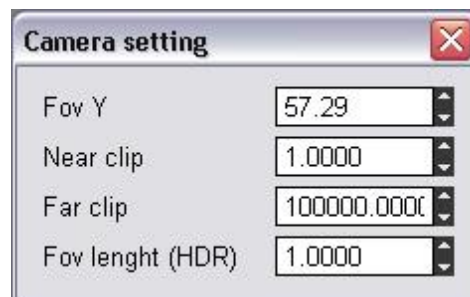
Near Clip and Far clip : Ces deux paramètres correspondent à la zone dans laquelle la camera fera le rendu (Ils définissent le frustrum)

Si un Objet, se trouve au-delà de la distance far clip alors il ne sera pas rendu sur l'image.

De même, si un objet se trouve avant la distance near clip elle ne sera pas rendu sur l'image finale.

Fov Y : Pour Field Of View, qui définit l'angle de vue de la caméra comme indiqué sur le schéma du dessus.

Lien avec l'Edit Setting des caméras dans l'OS3DEditor :



Dans l'edit setting des caméras, on retrouve les différents paramètres permettant de fixer le modèle de la caméra utilisée.

- Les Ombres

Les ombres et l'ombrage représentent un domaine particulièrement actif dans la 3D. L'ombrage est au cœur du réalisme d'une scène 3D. Les ombres résultent d'une interaction entre la lumière et les objets.

Suivant l'environnement et le type de lumière on utilisera l'une ou l'autre des techniques de projection d'ombres.

Les principales méthodes de projection des ombres sont :

Stencil Shadow Method et Texture Shadow Method

- Stencil Shadow Method :

Dans le fonctionnement global des stencils Shadow volumes, trois acteurs interviennent:

La lumière

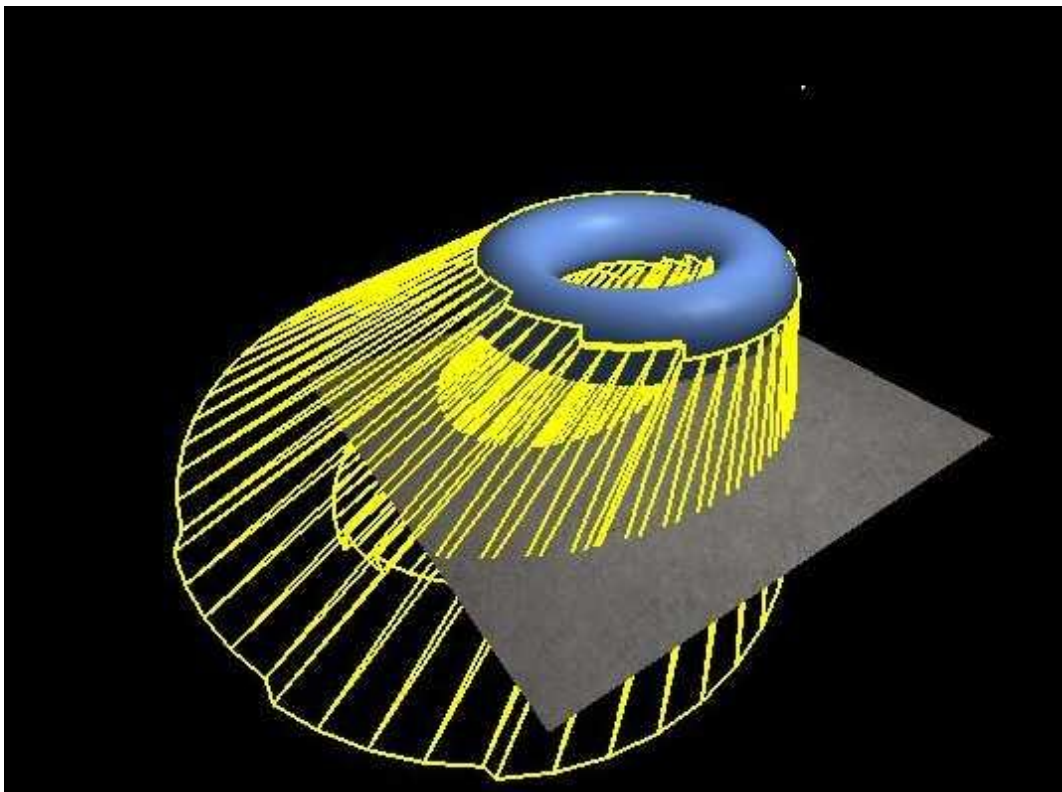
Le Shadow-caster

Le Shadow-receiver

La lumière est en général une source de lumière de type positionnelle ou OMNI.

Le Shadow-caster est le lanceur d'ombre: c'est l'objet qui va bloquer la lumière et créer l'ombre proprement dite. Le Shadow-receiver est un objet qui "reçoit" l'ombre, ou plus précisément c'est l'objet (tout ou une partie seulement) qui est privé de lumière.

Les ombres sont dites volumiques tout simplement parce que le Shadow-caster va créer un volume d'ombre dans lequel les objets s'y trouvant ne recevront pas de lumière.



- Texture Shadow Méthode /

```
textureModulative  
textureAdditive  
textureAdditiveIntegrated  
textureModulativeIntegrated
```

Le principe est assez simple et naturel. En effet qu'est-ce qui fait qu'il y a une ombre sur un objet ?

La réponse est évidente il y a un autre objet (ou lui-même) qui empêche la lumière d'atteindre la partie ombrée. Autrement dit, il y a une ombre sur une zone s'il y a un autre objet plus proche de la lumière et qui se trouve sur le chemin du rayon qui aurait dû illuminer notre zone.

Le principe est donc simple et très logique. Au pixel shader on va regarder s'il y a un objet qui se trouve sur le chemin du rayon de lumière et si c'est cas alors la zone est ombrée.

L'algorithme de base va donc se faire en deux étapes.

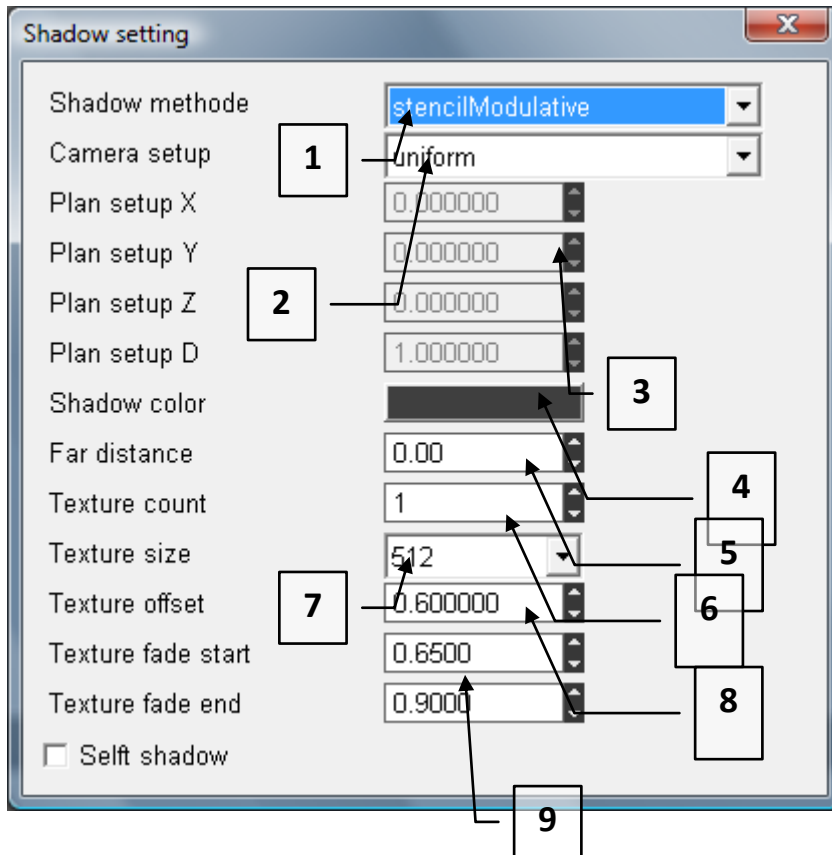
La première consiste à rendre la scène dans une texture (render target) en prenant pour caméra la lumière. Si on a plusieurs lumières, il faut le faire pour chacune d'entre elles. Lors de se rendu nous ne stockerons en sortie que ce dont nous avons besoin à savoir la profondeur.

Dans le cas d'une lumière omnidirectionnelle, tout ceci se complique, il faut passer par un cube map pour avoir la profondeur de tout ce qui est éclairé par la lumière.

La deuxième étape consiste à afficher la scène normalement et de s'occuper de l'ombre au pixel shader.

Le pixel shader a besoin des coordonnées du point dans le repère de la lumière. On a alors tout ce qu'il faut pour savoir si le pixel en cours est ombré ou non. En effet si la profondeur se trouvant dans la texture de profondeur aux coordonnées du pixel en cours est plus petite que celle (dans le repère de la lumière) du pixel en cours alors le pixel est ombré.

Lien avec l'Edit Setting des ombres sur les scènes dans l'OS3DEditor :



1/ Permet de Choisir le type de technique de la méthode d'ombres (stencil ou texture)

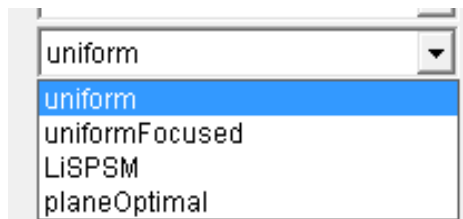
Additif et modulatif permettent de choisir la manière dont les ombres vont se mélanger avec le reste du rendu 3D

En additif les pixels seront ajoutés

En modulatif les pixels seront multipliés

2/ Option pour le cas des textures Shadow, permettant une optimisation du calcul des ombres ainsi qu'une plus grande efficacité du calcul d'ombre

Dans le cas où le choix est fait avec une optimisation en « Plan Optimal » alors il est nécessaire de fixer les paramètres du plan d'optimisation (3/)



4/ Permet tout simplement de définir la couleur des ombres projetées

5/ Permet de définir une distance à partir de laquelle les objets ne projeteront plus d'ombres (distance Camera/Objet) cela permet d'optimiser les scènes en ne faisant pas de calcul de projection au delà d'une distance donnée.

6/ Permet de définir le nombre de texture Shadow qui seront utilisées

On définit souvent une texture par lumière dans la scène.

Si on a 3 lumières dans la scène alors on fixe le texture count à 3

Cependant, il est déconseillé pour l'optimisation de la scène de fixer un texture count trop important (toutes les lumières ne sont pas obligées de projeter des ombres...)

7/ Définit la résolution de la texture d'ombres donc la résolution de la qualité des ombres.

8/ Définit un offset sur le rendu (un offset est un décalage de la texture d'ombre)

9/ ces paramètres vont définir l'atténuation de l'ombre sur les bords à partir du fade Start jusqu'au fade End ou la projection s'arrêtera.

N.B : Suivant le type de lumière de la scène la qualité des ombres sera plus ou mieux bien rendu. En effet par exemple, les stencils Shadow fonctionnent très bien avec des lumières de types directionnelles alors que les textures shadow vont bien réagir sur des lumières de types omnidirectionnelles ou spot.

N.B 2 : Il est impossible de définir plusieurs méthodes d'ombres pour une scène



Les SkyBoxes et les environnements de scène

Les fichiers de scène chargés dans openSpace 3d, exportés via Ogre Max sont des fichiers XML comportant les balises de définition des objets, des groupes d'objets mais également des variables de type environnement.

Ces environnements définissent par exemple le type de méthode d'ombres dans la scène ou encore les paramètres de Fog, la couleur de background...

Tout ce qui fait référence à la scène directement.

Ainsi, les skies ou les ciels sont également de ces variables.

Il existe dans Ogre 3 types de Skies :

- Les SkyBoxes : Pour ce type de Sky, le ciel est représenté sous la forme d'une boîte englobant la scène
- Les SkyDomes : Pour ce type de Sky, le ciel est représenté par un dôme
- Les SkyPlanes : Pour ce type de ciel, le ciel est représenté par un plan situé au dessus de la scène 3d.



Le moteur Physique

- Le moteur Physique Newton :

Le moteur physique utilisé par OpenSpace3D est le moteur Newton.



Une rapide description de ce moteur est disponible en français sur wikipedia :

http://fr.wikipedia.org/wiki/Newton_Game_Dynamics

- Une intégration partielle

L'intégration du moteur physique Newton dans OpenSpace3D n'a été que partiellement réalisée. Des fonctionnalités plus avancées arriveront dans une prochaine version d'OpenSpace3D.

Cependant, actuellement OpenSpace3D gère les bases fondamentales permettant de faire de la physique réaliste sur les environnements développés.

- Rappel et Définitions :

World : Il correspond au monde physique associé à une scène (il a une taille et des options de bases telles que la constante gravitationnelle)

Body : Enveloppe physique appliquée sur un objet afin d'optimiser le calcul de collision

Shape : Enveloppe Physique optimisée pour les objets complexes

Collision Tree : Enveloppe de collision exacte sur les objets considérés comme statiques (pas de forces) mais sensibles à la collision (ex : Murs d'une maison, sol..)

Architecture model : Niveau de précision du modèle qui sera utilisé par le moteur physique pour le calcul de la simulation

Solver model : Niveau de précision pour le calcul du résultat de l'application de force physique sur un objet

Solver model : Niveau de précision pour le calcul du résultat relatif à la friction entre objets.

FrameRate : Rapidité d'actualisation des phénomènes physiques

Physic material : Matériaux physiques associés à un body. Ex : bois, fer...

Angular Damping : Notion relative au centre de gravité d'un objet.

Contact : Phénomène se produisant lorsque deux objets (associé à des matériaux physiques) se collisionnent entre eux.

N.B : Les valeurs de réactions (élasticity, softness, friction...) à la collision de deux body correspondent à la résultantes du contact entre deux matériaux physiques.

Elasticity : Coefficient d'élasticité entre deux matériaux.

Softness : Coefficient correspondant à l'amortissement du contact.

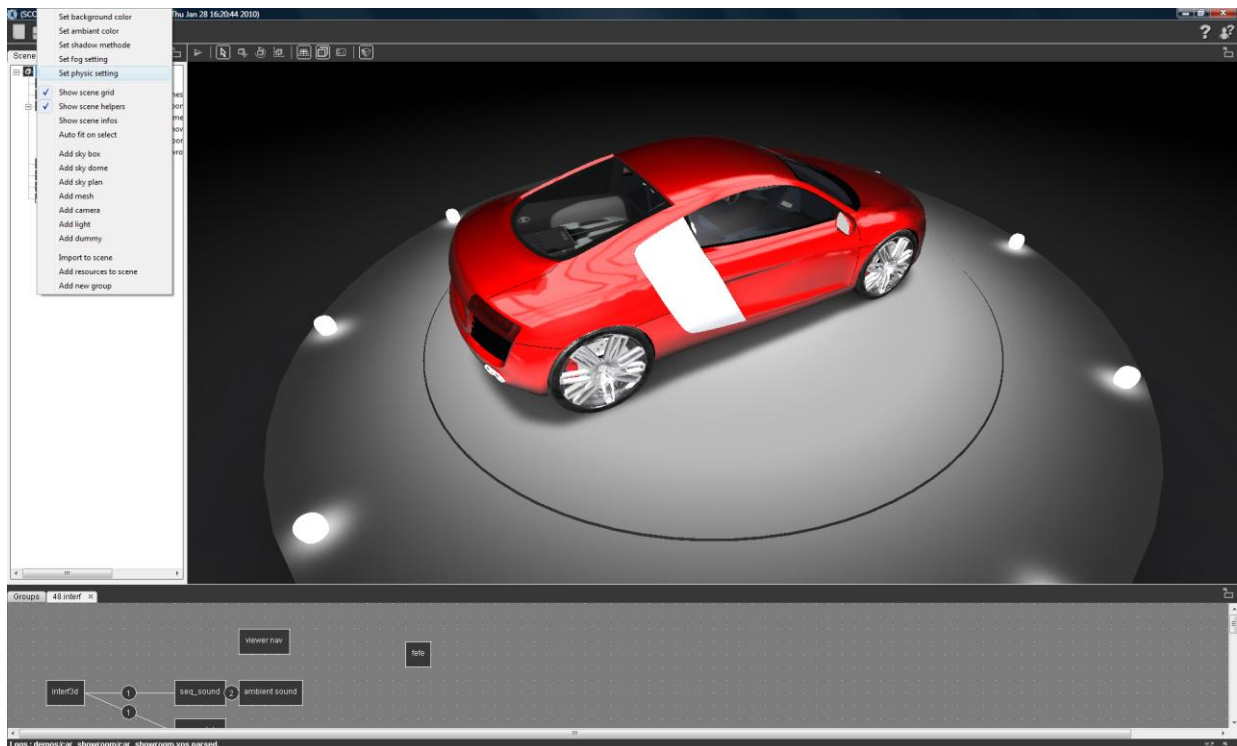
Thickness : Coefficient correspondant à l'épaisseur du contact.

Static friction : Coefficient de friction statique

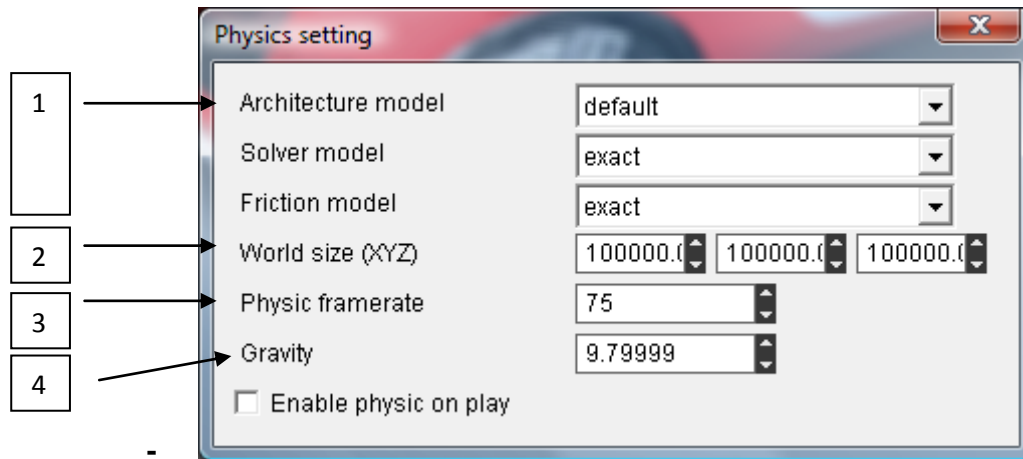
Kinetic friction : Coefficient de friction dynamique c'est-à-dire dépendant de la vitesse des objets au moment du contact.

- Les paramètres physiques dans OpenSpace3D:

Au niveau de la scène :



Par un clic droit sur la scène (dans le scene Tree) on accède aux configurations générales de la physique au niveau de la scène



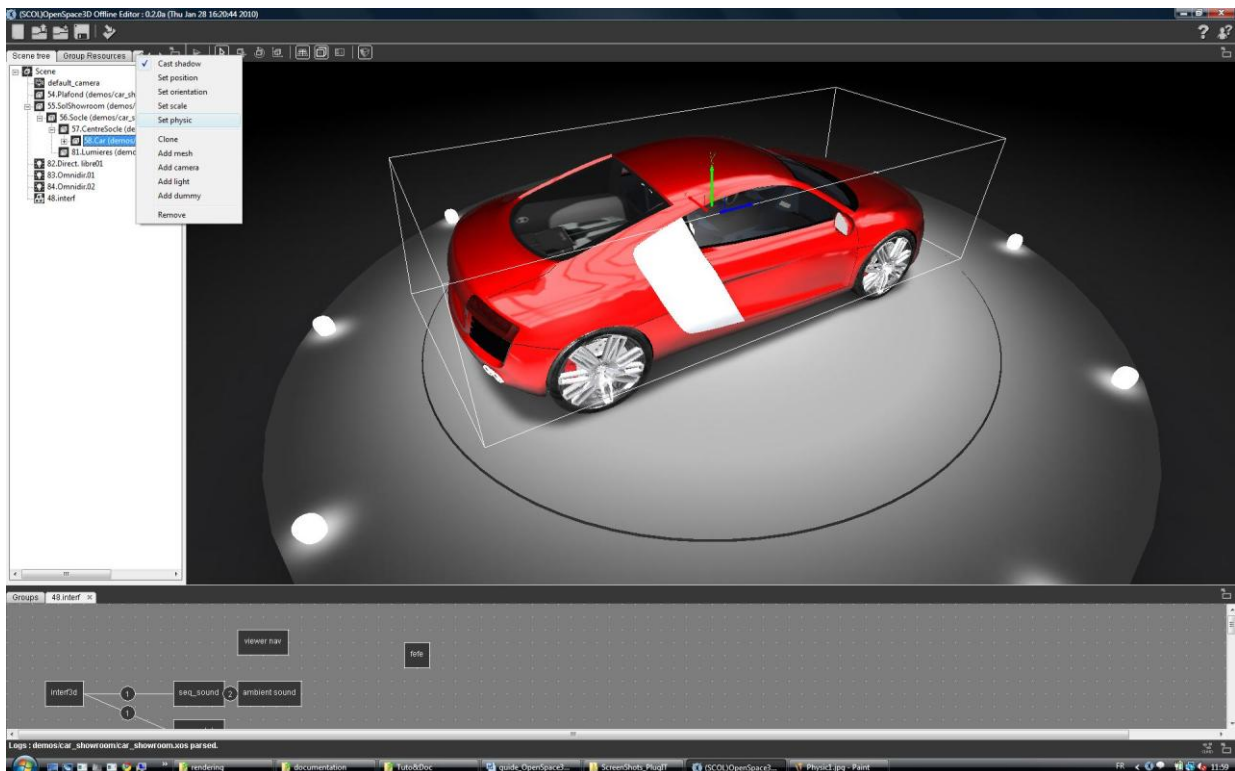
1 °/ Options de calcul du moteur physique (cf. Rappels et Définitions)

2 °/ Taille du monde Physique

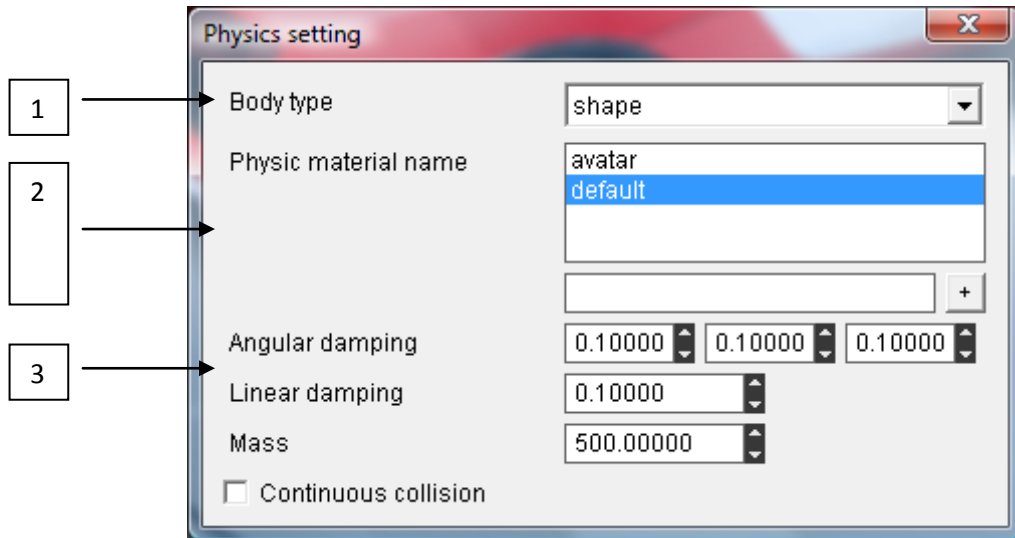
3 °/ Taux de rafraichissement du monde physique

4 °/ Valeur de la constante gravitationnelle (sur Terre : 9.81)

Au niveau des objets :



Par un clic droit un objet de l'arbre de scène : (set Physic) on accède à la configuration physique d'un objet :



1 °/ Type de body appliqué à l'objet

2 °/ Gestion des matériaux physiques : on choisit soit par le menu déroulant des matériaux présents dans la scène ou par un ajout direct par le nom sur quel material physique on applique notre objet

3 °/ Valeurs initiales de la physique sur l'objet : Damping (cf. Rappels et Définitions) et masse de l'objet en kg.

Contact et Support

N'hésitez pas à nous contacter pour toutes informations (problème de fonctionnement, propositions d'ajout, demande de formations) via le formulaire à l'adresse :

<http://www.openspace3d.com/support/>

De plus, le forum ou les développeurs d'OpenSpace3D pourront vous répondre très rapidement :

Le forum de la technologie SCOL : <http://www.scolring.org/>