

Virtual Universe Pro



Irai

Table des matières

Introduction	7
Installation	8
Configuration matérielle requise	8
Installation du logiciel	8
Enregistrement des licences	8
Enregistrer une licence	8
Enregistrer un code de sécurité (player illimité)	11
Installation en réseau	12
Vue d'ensemble	13
Menus et Fenêtres	13
Navigation et interactions	17
Etapes de construction d'un simulateur	19
Composition d'un simulateur	20
Types de données CAO 3D importables	24
Liste des connecteurs Automate disponibles	25
Fonctionnement interne de VIRTUAL UNIVERSE PRO	26
Construction d'un simulateur	28
Régler les propriétés générales	28
Propriétés d'affichage	28
Propriétés d'éclairage	30
Propriétés de visualisation	31
Propriétés de navigation	33
Options de simulation	35
Importer et alléger les modèles CAO 3D	38
Importer des modèles CAO 3D	38
Alléger les modèles CAO 3D	49
Concevoir des ressources et systèmes 3D intelligents	52
Modifier la position et les dimensions d'un sprite 3D	53
Ajouter des formes 3D primitives	55
Ajouter des comportements aux sprites 3D	57
Définir des profils de mouvement avec le Motion Assistant	67
Utiliser une bibliothèque de ressources 3D intelligentes	83
Importer une ressource 3D depuis la bibliothèque	84

Exporter une ressource 3D vers la bibliothèque.....	86
Connecter rapidement des ressources 3D avec l'option « Magnétique ».....	88
Connecter une simulation à un logiciel/contrôleur externe	91
Définir la liste des entrées/sorties d'une simulation.....	91
Définir la connexion avec le logiciel/contrôleur externe.....	94
Lier les entrées/sorties de la simulation aux variables du logiciel externe.....	95
Tester et mettre au point une simulation	97
Lancement de la simulation.....	97
Diagnostic de la simulation.....	97
Tester la simulation en forçant ses entrées/sorties	100
Mesurer et optimiser les performances d'une simulation.....	102
Mesurer les performances graphiques	103
Optimiser les performances graphiques.....	104
Mesurer les performances du moteur physique	107
Optimiser les performances du moteur physique	108
Performances du dialogue avec le logiciel/contrôleur externe	109
Générer des émulateurs 3D indépendants (players)	111
Players à durée limitée.....	112
Players à durée illimitée	112
Propriétés détaillées d'une simulation.....	114
Propriétés de l'Univers.....	114
Propriétés détaillées de l'Univers	114
Propriétés du Monde	117
Propriétés détaillées du Monde.....	117
Fonctionnalités au niveau Monde.....	120
Propriétés des Lumières.....	124
Propriétés détaillées des Lumières	124
Fonctionnalités au niveau Lumière.....	126
Propriétés des Caméras.....	127
Propriétés générales des Caméras.....	127
Fonctionnalités au niveau Caméra.....	129
Propriétés des Sprites	130
Propriétés détaillées des sprites	130
Fonctionnalités au niveau Sprite	134
Propriétés des Comportements	140

Propriétés détaillées des comportements	140
Types de comportements.....	142
Fonctionnalités au niveau Comportement	159
Propriétés des IHMs	160
Propriétés détaillés des IHM	161
Création ou modification d'un IHM	161
Propriétés des Contrôleurs	167
Fonctions de programmation.....	167
Propriétés détaillées des contrôleurs	169
Programmation d'un contrôleur	169
Eléments communs.....	172
Langage Ladder	173
Langage Fbd/Sfc.....	180
Mode RUN	188
Propriétés détaillées des surfaces.....	189
Vue haute qualité	201
Utilisation	201
Navigation	202
Configuration	202
Position initiale de la caméra.....	202
Qualité	202
Matériaux	203
Réalité virtuelle	211
Installation	211
Utilisation	211
Configuration	212
Contrôleurs Htc Vive.....	213
Contrôleur XBOX.....	216
Serveur de simulation	218
Principe	218
Configuration	218
Lancement du client PC	219
Lancement du client Web	220
Navigations.....	220

Serveur distant	221
Simulation de schémas	222
Ajout d'un folio de simulation	224
Edition du contenu d'un folio de simulation.....	225
Ajout d'un objet sur un folio de simulation	226
Tracé d'un lien sur un folio de simulation.....	228
Génération automatisée d'un schéma	228
Définition d'interfaces pour les objets intelligents	238
Généralités	238
Utilisation	239
Connexions externes	245
Connexion à un automate M340 ou au simulateur du logiciel Unity Pro de Schneider Electric.....	245
Connexion à un automate m238 de Schneider Electric.....	246
Connexion à un automate Siemens S7-1200, S7-1500, S7-300 ou S7-400.....	250
Connexion à un automate Rockwell Compact Logix, Control Logix ou à un émulateur Softlogix.....	251
Connexion à un serveur OPC.....	252
Connexion à l'émulateur PLCSIM de Siemens.....	253
Cas de PLCSIM 5.4 SP<5.....	253
Cas de PLCSIM 5.4 SP>=5.....	254
Connexion à l'émulateur CX-Simulator d'Omron	255
Connexion à AUTOMGEN.....	256
Connexion à CoDeSys	257
Connexion universelle	259
Exemple, utilisation de la connexion universelle avec l'émulateur automate du logiciel WinSPS-S7 de la société MHJ-Software.....	261
Exemple, utilisation de la connexion universelle avec le simulateur automate du logiciel SoMachine de Schneider Electric.....	265
Connexion à Universal Gateway	267
Syntaxe des liens	267
Accès direct aux variables externes	268

Introduction

VIRTUAL UNIVERSE PRO est un logiciel innovant de modélisation et de simulation 3D, permettant de construire rapidement des simulateurs 3D interactifs de systèmes automatisés (ou « machines virtuelles »), en exploitant les modèles 3D issus des logiciels de CAO. Les concepteurs d'équipements industriels et de machines automatisées peuvent ainsi expérimenter leurs systèmes dans un environnement virtuel 3D réaliste et émuler leur comportement en temps réel. En connectant les simulateurs 3D à des contrôleurs externes de type API (Automate Programmable Industriel) ou à des contrôleurs virtuels embarqués, VIRTUAL UNIVERSE PRO permet de reproduire les conditions réelles de fonctionnement d'un produit ou d'une machine, dans un environnement virtuel.

Installation

Configuration matérielle requise

VIRTUAL UNIVERSE PRO fonctionne sur les systèmes d'exploitation Windows suivants : Windows XP, Windows Vista, Windows 7, 8 ou 10.

VIRTUAL UNIVERSE PRO est disponible en version 32 bits pouvant fonctionner sous les version 32 et 64 bits de Windows ou en version 64 bits pouvant fonctionner sous les version 64 bits de Windows. La version 32 bits ne peut utiliser que 2Go de mémoire au maximum. Pour tirer le meilleur parti de ses performances (notamment en simulation), il est vivement recommandé d'utiliser VIRTUAL UNIVERSE PRO sur un ordinateur récent, doté d'une carte graphique et d'un processeur performants.

L'utilisation d'un casque de réalité virtuelle nécessite des performances élevées pour le PC. Pour la compatibilité et les PC recommandés, se référer au site Internet de chaque constructeur.

Installation du logiciel

Pour installer VIRTUAL UNIVERSE PRO, lancez l'exécution du package d'installation.

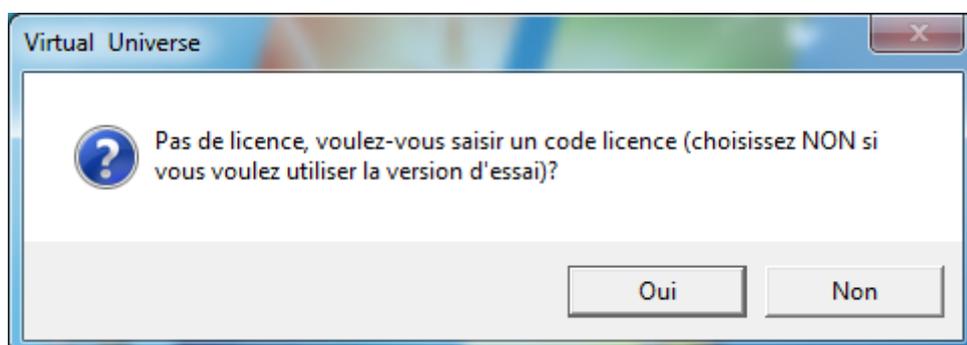
Enregistrement des licences

Enregistrer une licence

VIRTUAL UNIVERSE PRO fonctionne en version d'évaluation (limitée à 40 jours d'essai) tant que vous n'avez pas enregistré la licence.

Pour enregistrer la licence, lancez VIRTUAL UNIVERSE PRO en mode administrateur en faisant un clic droit sur le raccourci de lancement et en choisissant "Exécuter en tant qu'administrateur".

Au premier lancement de VIRTUAL UNIVERSE PRO, si aucune licence n'a encore été enregistrée sur le PC, la fenêtre suivante est affichée :



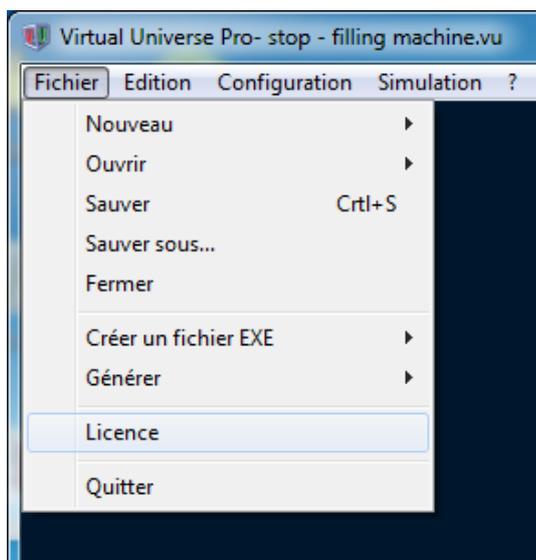
Cette opération nécessite que le PC soit connecté à Internet.

Cliquez sur "Oui" pour entrer le code licence qui vous a été fourni (code commençant par "C." ou "R."). Utilisez le code commençant par "R." si vous souhaitez installer définitivement la licence sur le PC. Le code commençant par "C." permet de se connecter à la licence sans la transférer sur le PC. Dans ce dernier cas, la licence restera utilisable depuis un autre PC mais nécessitera de rester connecté à Internet à chaque fois que vous voudrez utiliser le logiciel.

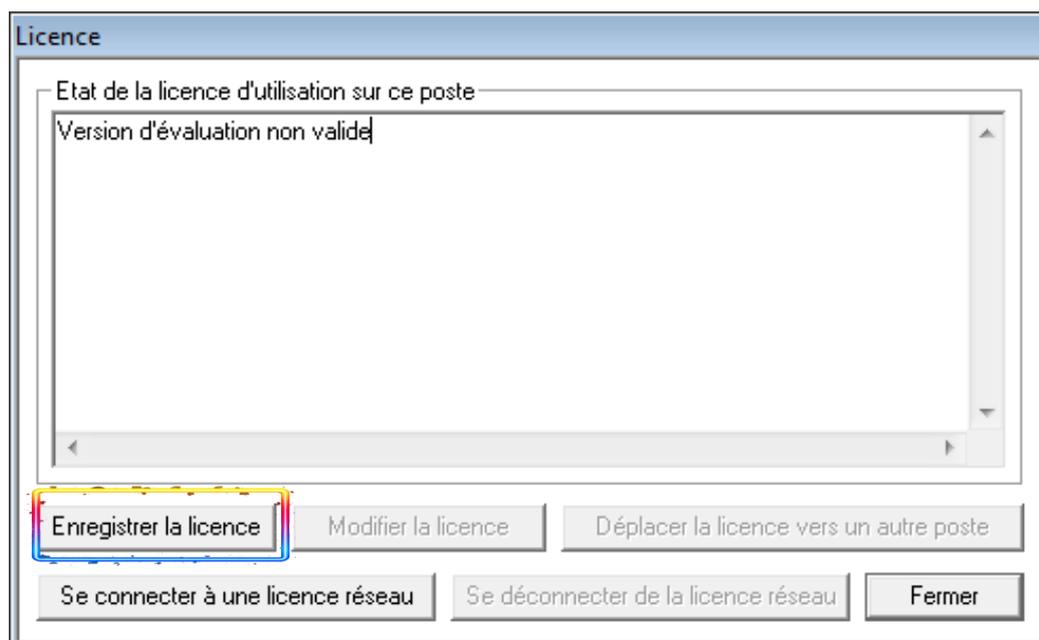
La boîte de dialogue suivante vous permet d'entrer le code licence:



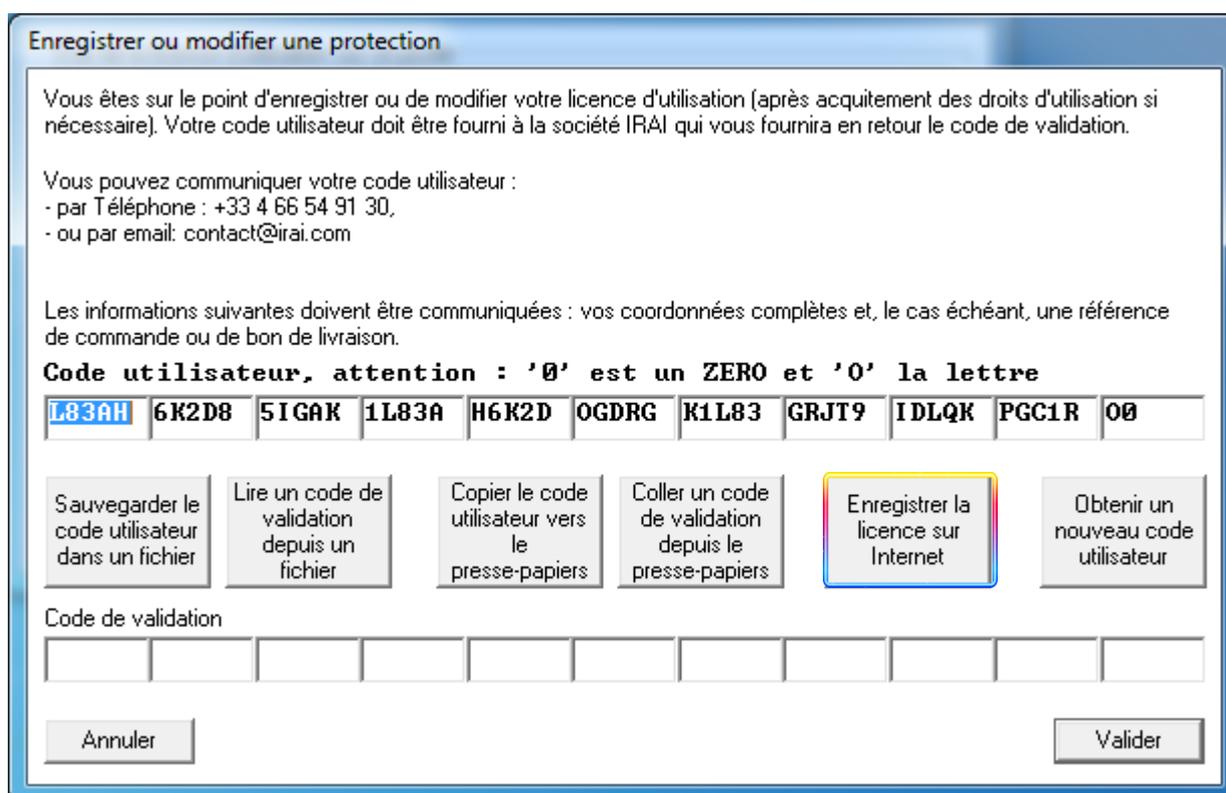
Si vous souhaitez accéder à l'enregistrement de la licence une fois le logiciel lancé, sélectionnez « Licence » dans le menu Fichier :



Une fenêtre License s'ouvre. Cliquez sur le bouton « Enregistrer la licence » :



Cliquez sur le bouton « Enregistrer la licence sur Internet » et saisissez votre code licence.

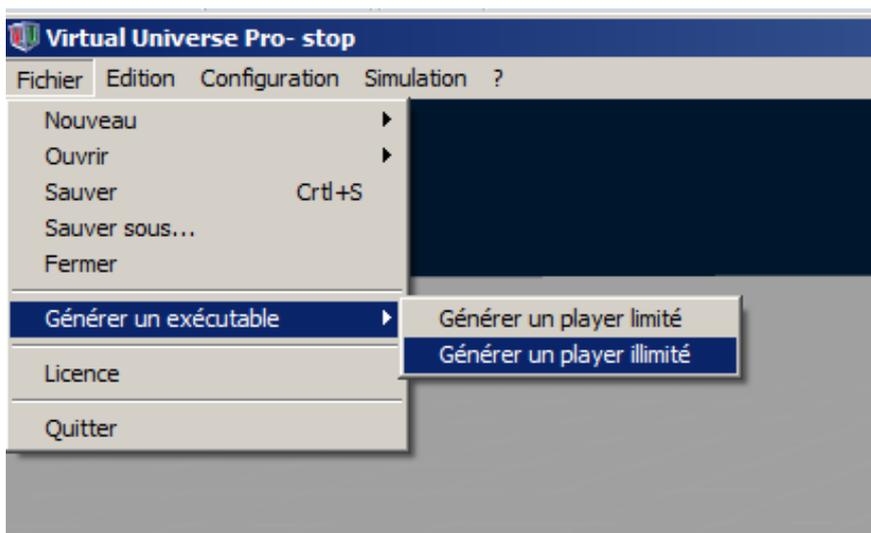


Enregistrer un code de sécurité (player illimité)

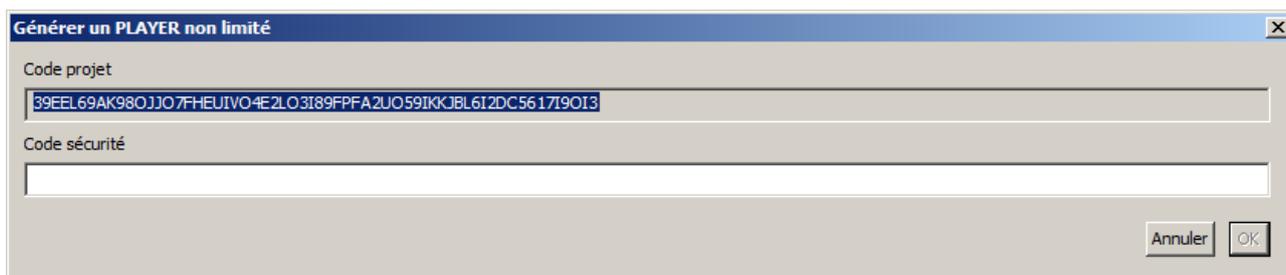
VIRTUAL UNIVERSE PRO permet de générer des simulateurs 3D indépendants (fichiers exécutables nommés « players ») ayant une durée d'utilisation illimitée.

La génération des players à durée illimitée est protégée par un code de sécurité. Chaque player généré possède un code de sécurité unique.

Pour enregistrer un code de sécurité, sélectionnez « Générer un exécutable/Générer un player illimité » dans le menu Fichier:



Une fois le nom du player renseigné, une fenêtre s'ouvre, affichant un code projet.



Copiez et collez ce code projet dans un mail et envoyez ce mail à l'adresse suivante: contact@irai.com

Vous recevrez un code de sécurité unique à enregistrer dans le champ « Code sécurité » vous permettant de générer le player à durée illimitée. A réception du code de sécurité, vous disposez de 30 jours pour utiliser ce code et générer le player. Une fois le player généré, la fonctionnalité de génération sera à nouveau bloquée, et nécessitera l'enregistrement d'un nouveau code de sécurité.

Il est vivement recommandé d'utiliser les players à durée limitée pour faire vos tests (test de la communication avec le logiciel externe,..) avant de générer le player à durée illimitée, considéré comme définitif.

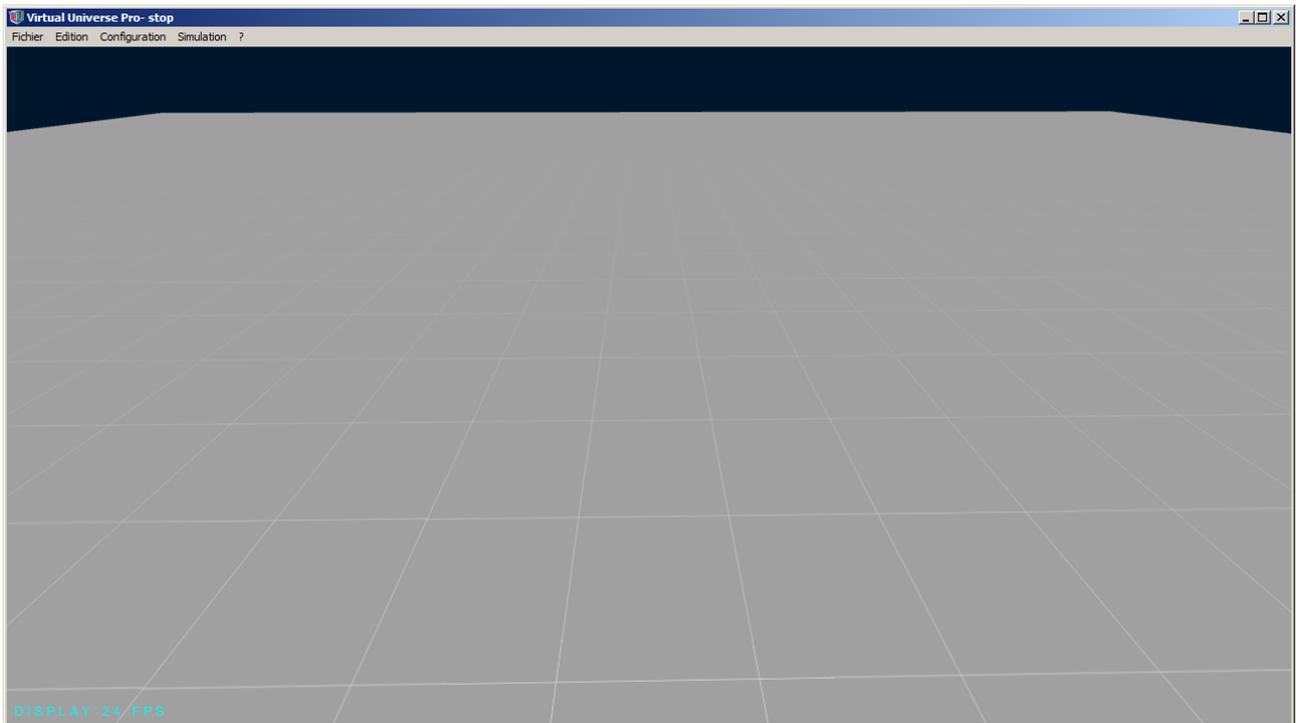
Installation en réseau

VIRTUAL UNIVERSE PRO peut être installé sur un serveur de fichiers, et les licences être gérées par un gestionnaire de licences réseau. Merci de vous rapprocher de notre support technique pour procéder à cette installation : contact@irai.com

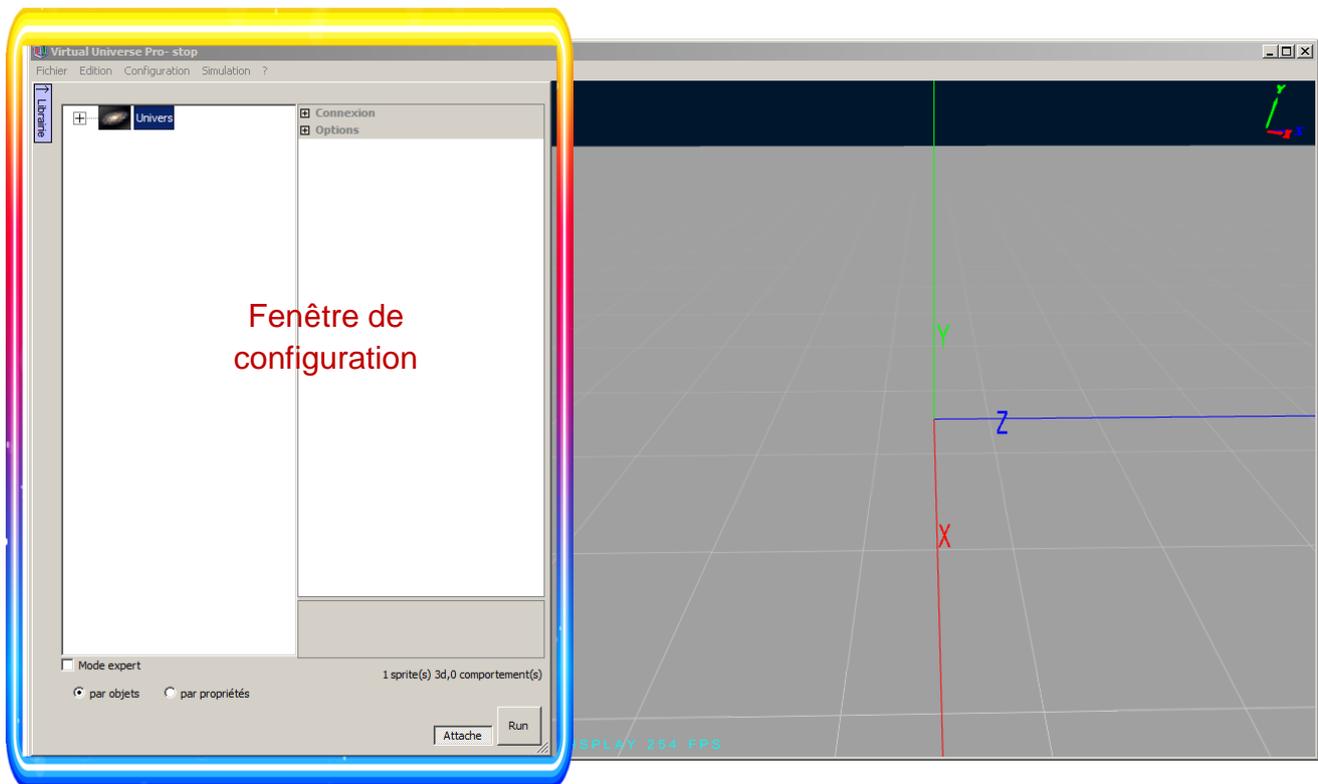
Vue d'ensemble

Menus et Fenêtres

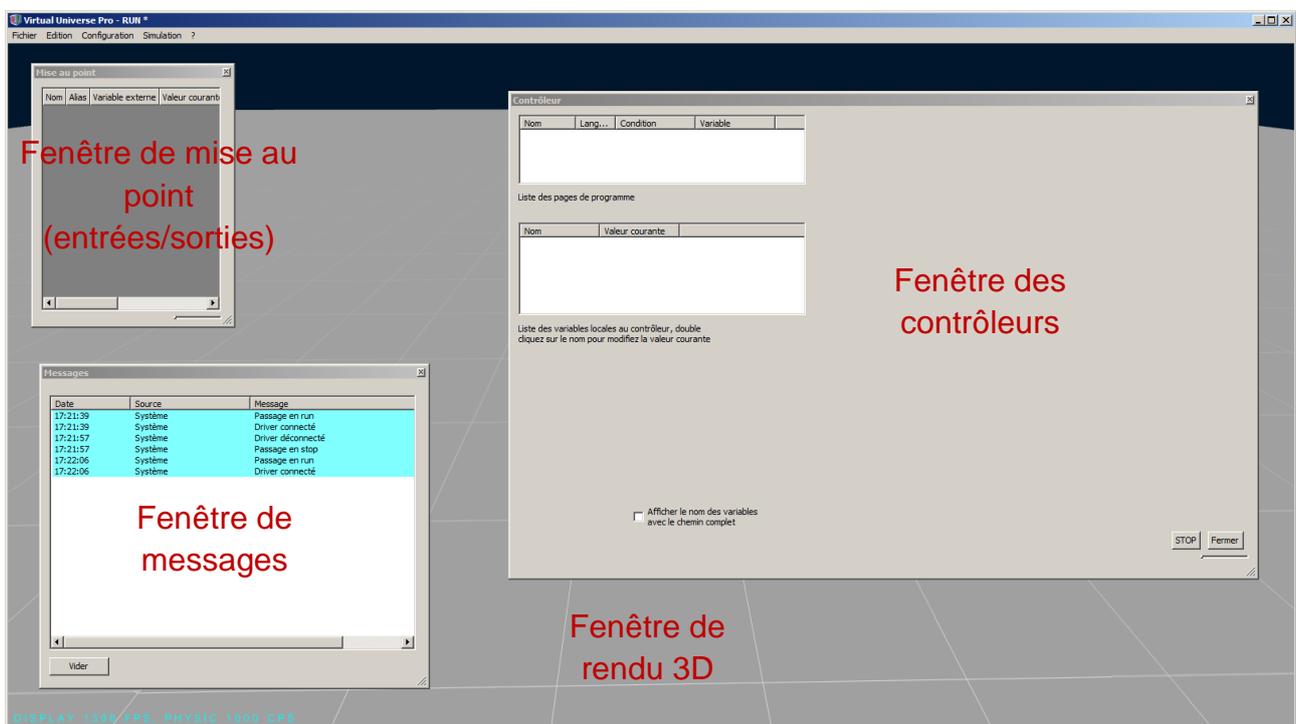
Au lancement de VIRTUAL UNIVERSE PRO, un projet vierge est ouvert dans la fenêtre principale du rendu 3D.



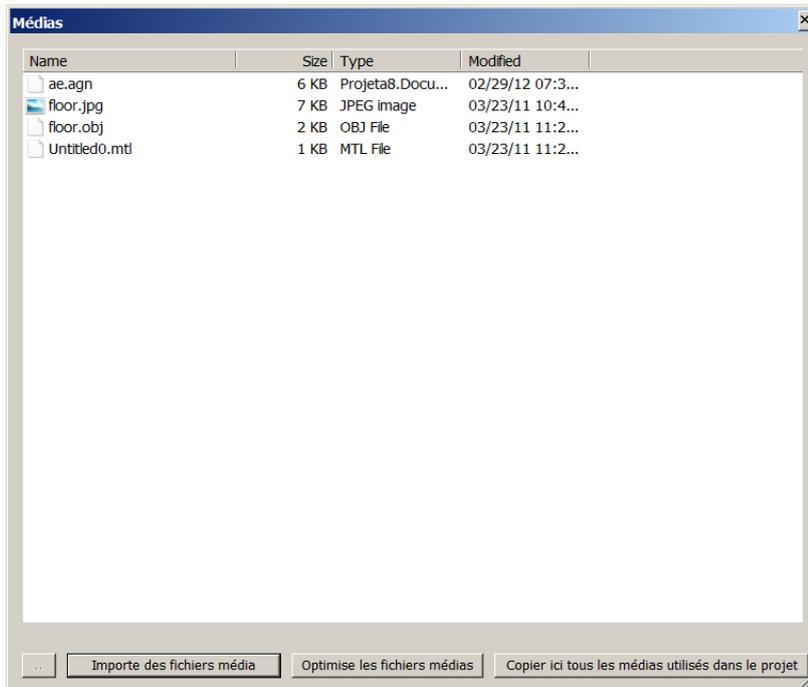
- Le menu **Configuration** permet d'accéder à la fenêtre de Configuration et aux outils de construction du projet.



- Le menu **Simulation** permet d'accéder au lancement de la simulation et aux outils de debug de la simulation.



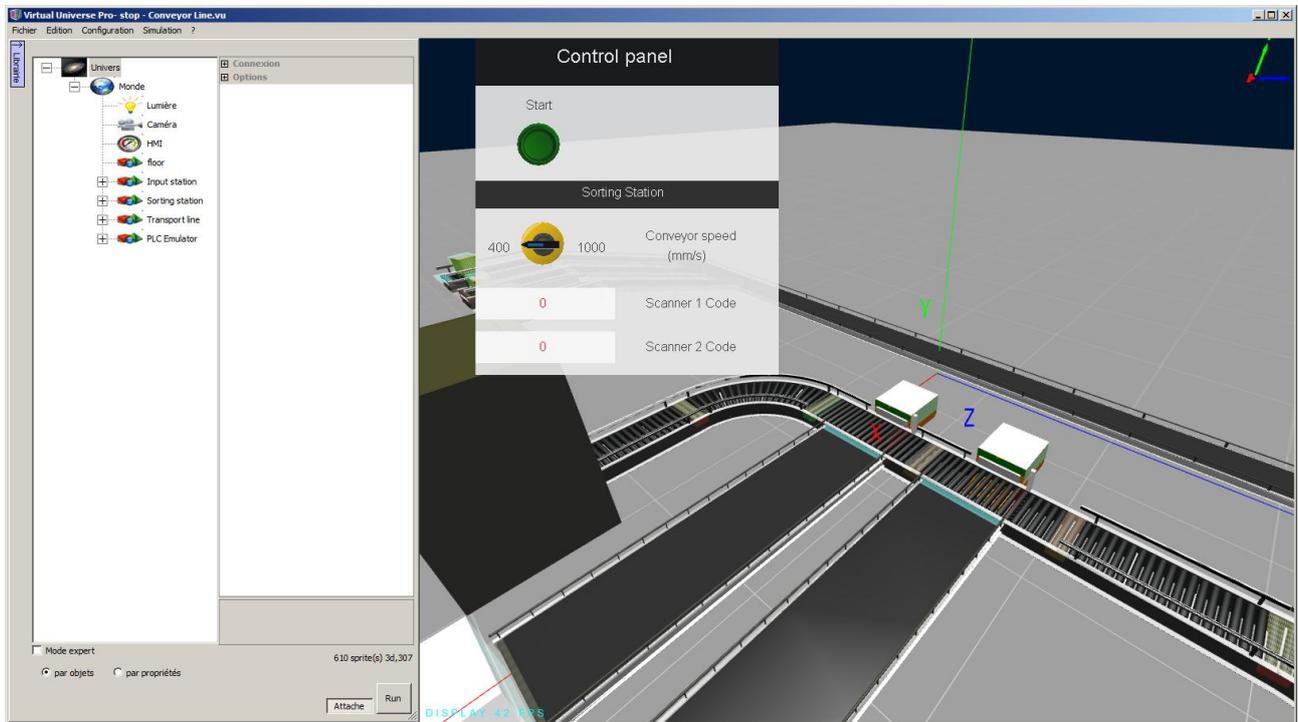
- Le menu **Medias** permet d'accéder au gestionnaire des médias présentant la liste de tous les fichiers médias (fichiers 3D, fichiers bitmaps) qui sont utilisés dans le projet de simulation. Tous ces fichiers sont sauvegardés automatiquement dans le fichier du projet.



- Le menu ? permet d'accéder à la documentation et aux informations du logiciel.
- Le menu **Fichier** permet de créer un nouveau projet, d'ouvrir un projet existant, de sauvegarder et fermer un projet. Ce menu donne également accès à la génération de simulateurs indépendants (fichiers exécutables appelés « players », à période d'utilisation limitée ou illimitée) ainsi qu'à l'enregistrement des licences VIRTUAL UNIVERSE PRO.
- Le menu **Edition** permet d'annuler ou de rétablir les modifications réalisées dans le projet courant.

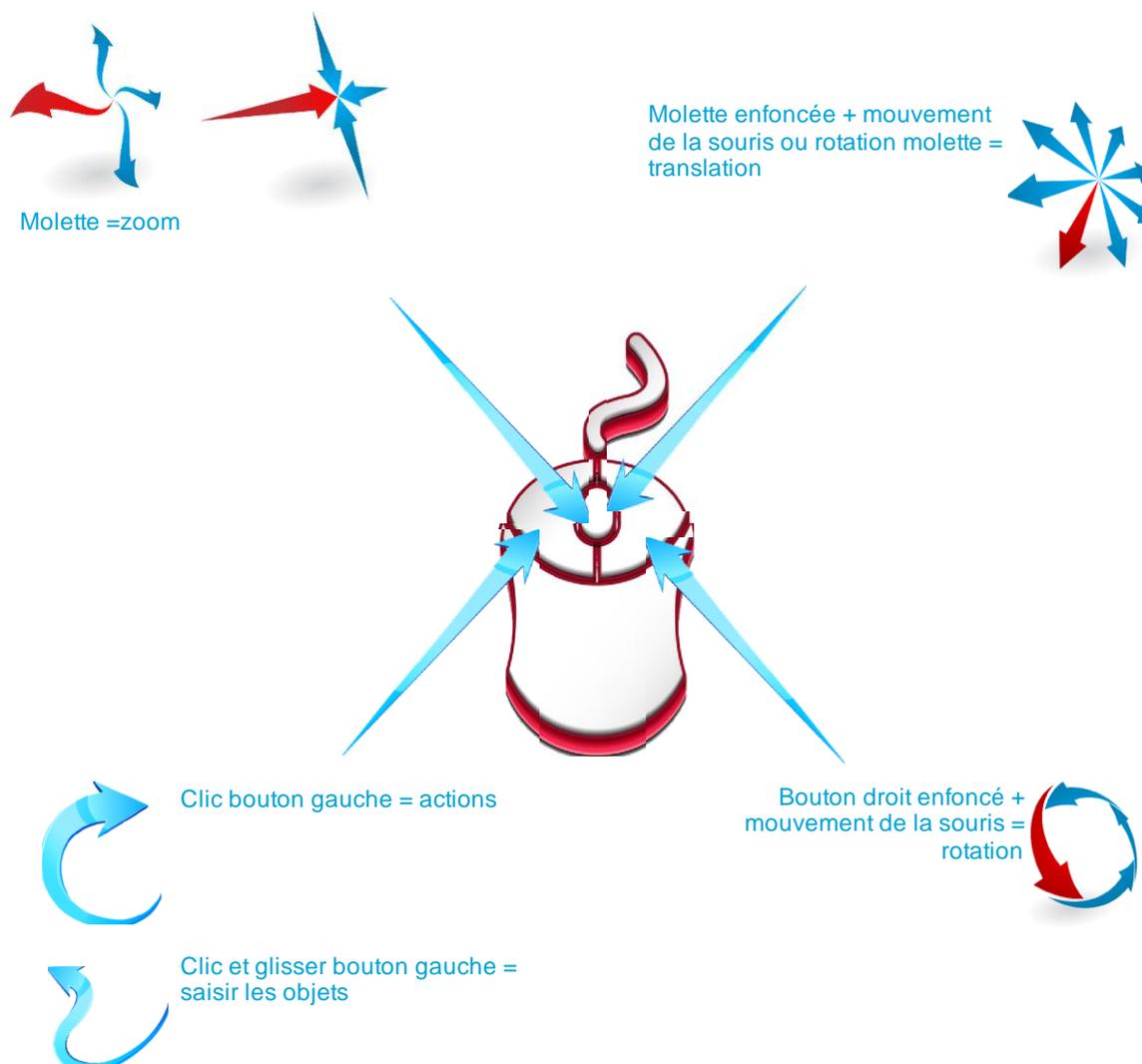
Exemple

Pour vous familiariser aux différents menus et fenêtres de VIRTUAL UNIVERSE PRO, des exemples de simulateurs sont livrés avec le produit et disponible via le menu Fichier/Ouvrir/Ouvrir un exemple.

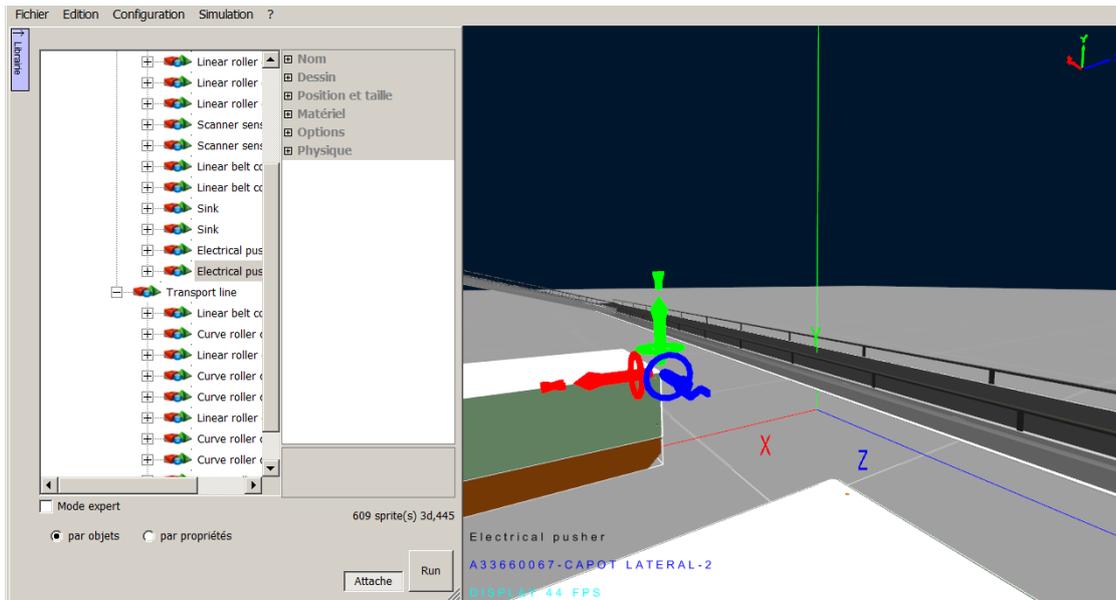


Navigation et interactions

Par défaut, la navigation dans le monde 3D et l'interaction avec les objets 3D durant la simulation s'opèrent par l'intermédiaire de la souris.

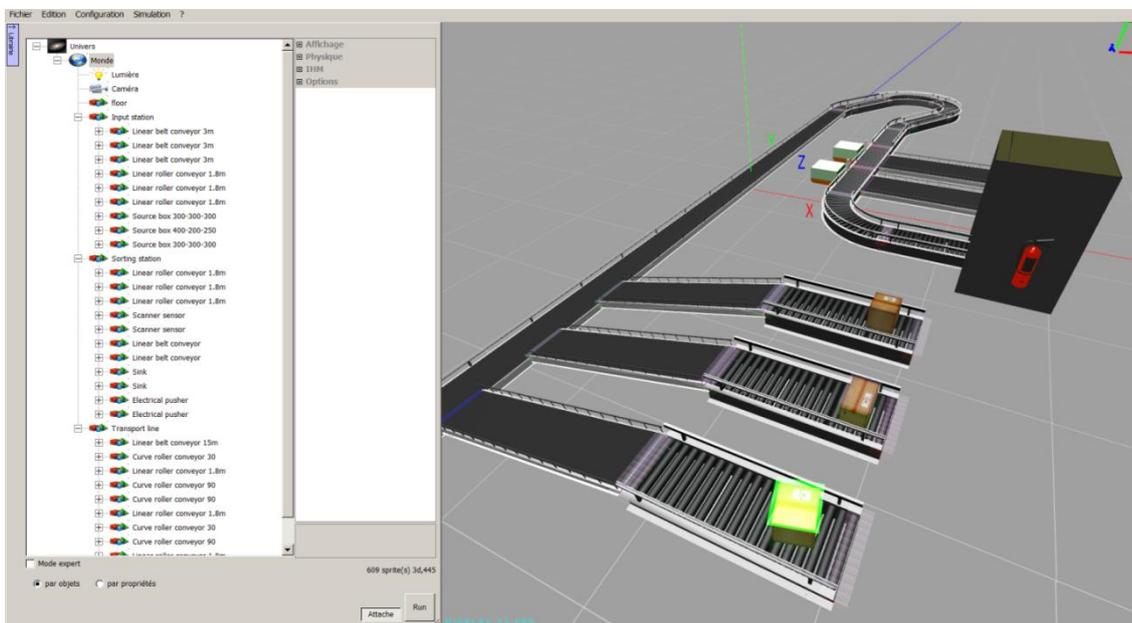


Hors simulation et lorsque la fenêtre de configuration est ouverte, un repérage des axes (X,Y,Z) est affiché dans la fenêtre de rendu 3D au centre du monde, ainsi que sur le sprite 3D sélectionné dans l'arborescence.



Exemple

Pour vous familiariser avec la navigation et les interactions dans VIRTUAL UNIVERSE PRO, des exemples de simulateurs sont livrés avec le produit et disponible via le menu Fichier/Ouvrir/Ouvrir un exemple.



Etapes de construction d'un simulateur

VIRTUAL UNIVERSE PRO permet de construire des simulateurs interactifs de systèmes automatisés (ou «machines virtuelles»), composés d'une partie opérative virtuelle pilotée par un ou plusieurs contrôleurs (internes ou externes aux simulateurs VIRTUAL UNIVERSE PRO).

- Les dessins et modèles 3D importés depuis les outils de CAO permettent de créer dans VIRTUAL UNIVERSE PRO les ressources 3D (composés d'objets 3D appelés « sprites ») et assemblages de ressources 3D qui vont constituer les éléments de la partie opérative contrôlée.
- L'ajout de comportements aux sprites donne une véritable intelligence aux ressources 3D et permet de modéliser le fonctionnement de ressources, telles que les actionneurs et récepteurs de la partie opérative. Les comportements sont, soit des comportements prédéfinis disponibles dans VIRTUAL UNIVERSE PRO, soit des comportements plus élaborés de type script créés par l'utilisateur avec l'éditeur de script intégré à VIRTUAL UNIVERSE PRO.
- Les ressources créées dans VIRTUAL UNIVERSE PRO (ressources 3D intelligentes ou simples comportements sans représentation 3D) peuvent être capitalisées dans une bibliothèque intégrée à VIRTUAL UNIVERSE PRO et être réutilisées pour construire rapidement de futurs projets de simulateurs.
- Il est possible de modéliser un ou plusieurs contrôleurs internes dans le simulateur (contrôleur de mouvement, séquence de contrôle) et d'ajouter un pupitre de contrôle 2D modélisant une interface homme-machine.
- Le simulateur ainsi construit peut être raccordé aux entrées/sorties d'un contrôleur externe (de type automate programmable industriel).
- A tout moment durant sa construction, les performances du simulateur (qualité du rendu 3D, performances du moteur physique,..) peuvent être mesurées et si nécessaire, être améliorées.
- Il est enfin possible de générer des simulateurs indépendants (fichiers exécutables appelés « players ») limités à la simulation d'un projet et ne nécessitant aucune installation logicielle.

Composition d'un simulateur

A tout moment, la fenêtre de **Configuration** permet d'accéder à la construction et configuration d'un projet de simulation. Les éléments constituant le projet de simulation sont représentés sous forme d'une arborescence.



- Le niveau **Univers**  Univers permet d'accéder aux propriétés générales d'utilisation du simulateur :
 - Utilisation du simulateur avec ou sans connexion à un logiciel/contrôleur externe
 - Démarrage automatique de la simulation à l'ouverture du projet
 - Mode de navigation dans le rendu 3D
 - Réglage des droits d'accès aux propriétés de la simulation
 - ...etc.

Un réglage par défaut de ces propriétés est proposé. Ces propriétés sont modifiables par l'utilisateur. Pour plus de détails sur les propriétés de l'Univers, [voir Propriétés de l'Univers](#).

- Le niveau **Monde**  Monde permet d'accéder aux propriétés générales d'affichage de la simulation :
 - Réglage de la couleur du fond
 - Réglage de la lumière ambiante
 - Ajout d'une image pour le ciel
 - Réglage des unités
 - ...etc.

Un réglage par défaut de ces propriétés est proposé. Ces propriétés sont modifiables par l'utilisateur. Pour plus de détails sur les propriétés du Monde, voir [Propriétés du Monde](#).

- Le niveau **Lumière**  Lumière permet d'accéder aux propriétés d'éclairage de la simulation.

Par défaut, la simulation possède déjà une lumière pré-réglée. Il est possible de définir plusieurs lumières dans la simulation. Pour plus de détails sur les propriétés des Lumières, voir [Propriétés des lumières](#).

- Le niveau **Caméra**  Caméra permet de définir les options de visualisation de la simulation.

Par défaut, la simulation possède déjà une caméra pré-réglée. Il est possible de définir plusieurs caméras dans la simulation afin de créer différents points de vue. Pour plus de détails sur les propriétés des Caméras, voir [Propriétés des caméras](#).

- Le niveau **Sprite**  Sprite permet d'accéder aux propriétés d'un Sprite.

Les Sprites sont les objets 3D constitutifs d'un projet. Un Sprite est le plus souvent associé à un fichier image représentant une forme 3D avec ses dimensions et son emplacement dans le monde 3D. Un sprite peut également ne posséder aucune image et ne servir qu'à structurer la donnée 3D. Les Sprites sont structurés selon une arborescence parent/enfant.

Il est possible d'ajouter, de copier, de coller, de déplacer, de supprimer manuellement des sprites afin de construire des assemblages et ressources en 3D.

L'importation de données 3D issues des logiciels de CAO permet de recréer automatiquement dans VIRTUAL UNIVERSE PRO des assemblages et des ressources 3D, composés d'un ensemble structuré de sprites.

Ces assemblages de sprites et ressources 3D créés dans VIRTUAL UNIVERSE PRO peuvent être capitalisés dans une bibliothèque intégrée à VIRTUAL UNIVERSE PRO et être réutilisés pour construire rapidement de futurs projets d'émulateurs 3D.

Pour plus de détails sur les propriétés des Sprites, voir [Propriétés des Sprites](#).

- Le niveau **Comportement**  **Comportement** permet d'accéder aux propriétés d'un Comportement.

Les comportements sont au cœur de la simulation dans VIRTUAL UNIVERSE PRO. Ils représentent l'intelligence apportée aux Sprites en simulation. Les comportements permettent de transformer des Sprites 3D inertes en de véritables ressources 3D de simulation, intelligentes, communicantes et pouvant être mobiles en cours de simulation.

Il existe différents types de comportements prédéfinis disponibles dans VIRTUAL UNIVERSE PRO selon le type de ressource à modéliser (actionneur, récepteur, contrôleur). Les comportements peuvent également être des entrées/sorties pour les Sprites, leur permettant de dialoguer entre eux et avec tout contrôleur externe à la simulation.

En outre, un éditeur de scripts permet de décrire (dans un langage Basic) des comportements plus élaborés à partir des comportements prédéfinis afin de modéliser une véritable logique comportementale pour la ressource.

Les comportements, tout comme les sprites, peuvent être sauvegardés et capitalisés dans une bibliothèque intégrée à VIRTUAL UNIVERSE PRO et être réutilisés pour construire rapidement de futurs projets de simulation.

Il est possible d'ajouter, de copier, de coller, de déplacer, et de supprimer manuellement des comportements.

Pour plus de détails sur les propriétés des Comportements, voir [Propriétés des Comportements](#).

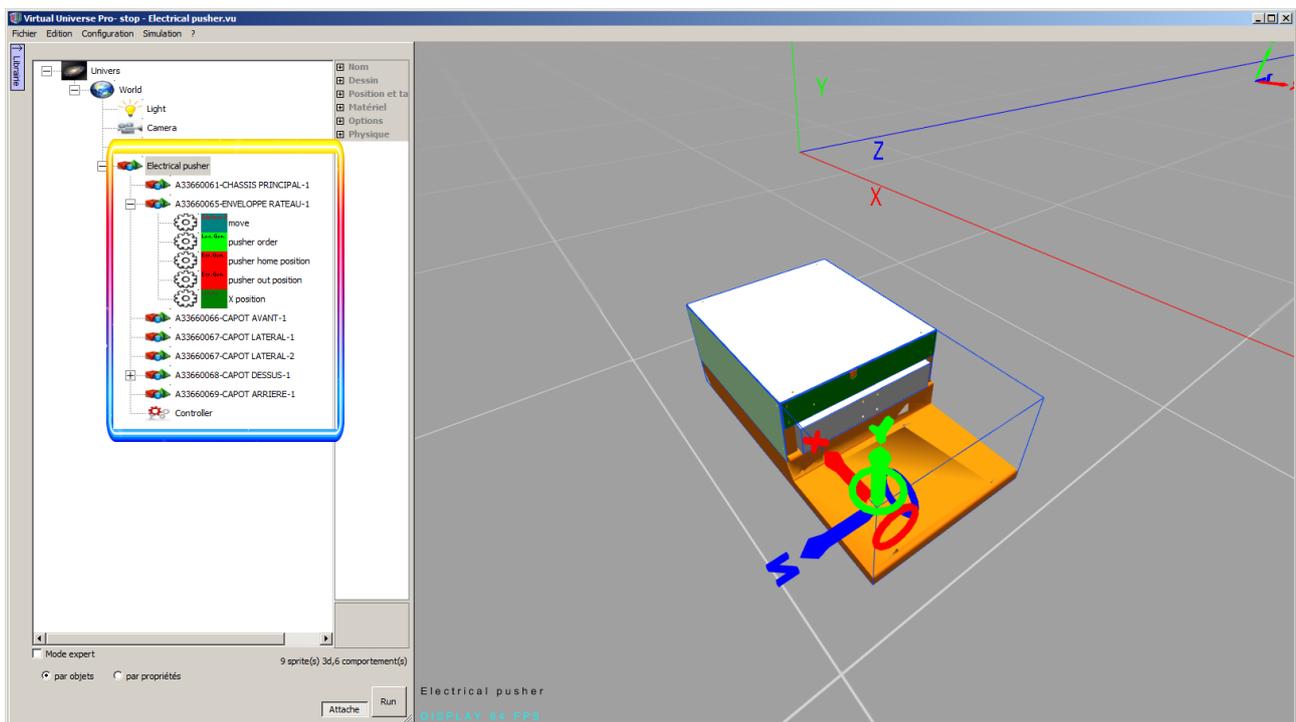
- Le niveau **IHM**  **IHM** permet d'accéder aux propriétés d'un IHM. Les IHMs permettent de créer des pupitres qui seront affichés sur la fenêtre de rendu et qui seront composés d'éléments tels que des boutons poussoirs, voyants, potentiomètres, etc.
Pour plus de détails, sur les propriétés des IHMs voir [Propriétés des IHMs](#).
- Le niveau **Contrôleur**  **Contrôleur** permet d'accéder aux propriétés d'un contrôleur. Les contrôleurs peuvent être enfants du monde ou d'un sprite. Ils permettent de créer des programmes pour piloter un système complet ou un sous-

ensemble. Pour plus de détails, sur les propriétés des Contrôleurs voir [Propriétés des Contrôleurs](#).

- Le niveau **Surface**  **Surface** permet de définir des surfaces à plaquer sur les Sprites 3d.

Exemple

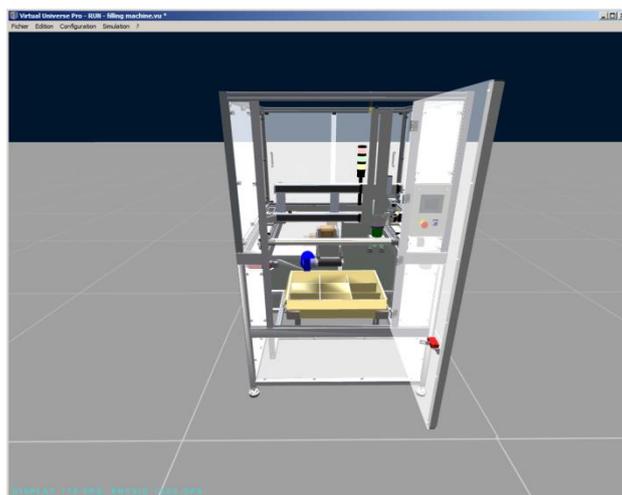
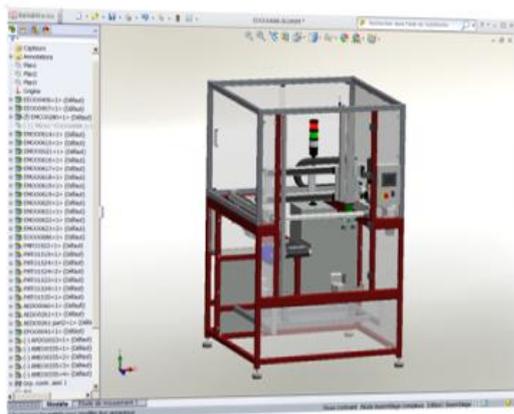
Pour bien comprendre le concept des sprites et des comportements dans VIRTUAL UNIVERSE PRO, nous vous suggérons de regarder l'exemple de la ressource « Electrical Pusher » utilisée dans l'exemple de simulation livré avec le produit (menu Fichier/Ouvrir/Ouvrir un exemple).



Types de données CAO 3D importables

VIRTUAL UNIVERSE PRO permet de réutiliser les modèles 3D issus des logiciels de CAO (Conception Assistée par Ordinateur) en les important directement depuis le logiciel de CAO, ou indirectement par l'intermédiaire d'un fichier d'échange.

Logiciel de CAO



Voici les types de données CAO qu'il est possible d'importer et de réutiliser aujourd'hui dans VIRTUAL UNIVERSE PRO :

Logiciel de CAO	Fichiers d'échange (import indirect)
DS SolidWorks	3DXml
Autodesk Inventor	
DS Catia	3DXml
DS Delmia	3DXml
Siemens SolidEdge	

Liste des connecteurs Automate disponibles

Les émulateurs 3D de partie opérative créés avec VIRTUAL UNIVERSE PRO peuvent être raccordés à des API (Automates Programmables Industriels) ou autres cibles, afin de créer de véritables systèmes automatisés virtuels.

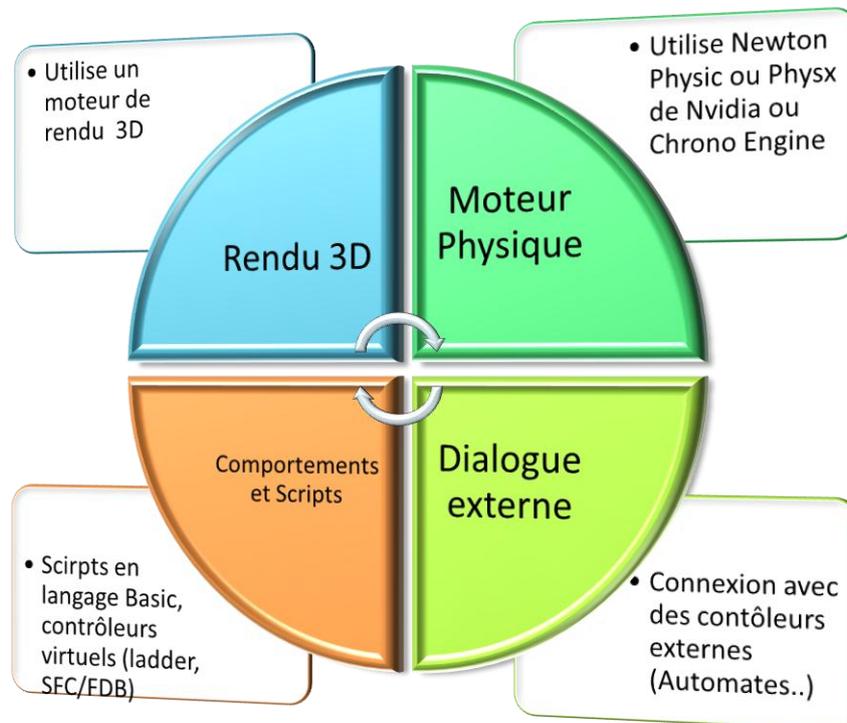
L'automate connecté à VIRTUAL UNIVERSE PRO pilote directement le système virtuel simulé dans VIRTUAL UNIVERSE PRO, par l'intermédiaire d'un dialogue établi en permanence durant la simulation entre les entrées-sorties du programme automate et celles du système virtuel.

VIRTUAL UNIVERSE PRO est compatible avec certaines marques d'Automates Industriels, et propose pour cela plusieurs types de connecteurs (protocoles de communication). En voici la liste :

Communication avec APIs
OPC
SLMP
MODBUS-TCP
MITSUBISHI MELSEC
ROCKWELL
ETHERNET IP
SIEMENS S7 IP
SIEMENS S7 PPI
SIEMENS S7 MPI
SIEMENS S5
BECKHOFF
SCHNEIDER ELECTRIC TWIDO
SCHNEIDER ELECTRIC SOMACHINE API COMPATIBLES
AUTOMGEN API COMPATIBLES
SCHNEIDER ELECTRIC TWIDO
CODESYS APIs COMPATIBLES (Ex. Schneider M238)
CONNEXION DIRECTE AUX E/S DE L'API AVEC CARTE d'E/S ADVANTECH
Communication avec simulateurs APIs
SIEMENS PLCSIM
SCHNEIDER ELECTRIC SIMULATEUR UNITY
ROCKWELL SOFTLOGIX
UNIVERSAL MEMORY ACCESS
OMRON CX-SIMULATOR
CODESYS SIMULATEUR
SCHNEIDER ELECTRIC SIMULATEUR SOMACHINE
MHJ WINSPS SIMULATEUR
EXECUTEUR PC AUTOMGEN

Fonctionnement interne de VIRTUAL UNIVERSE PRO

Il peut être utile de comprendre quels sont les technologies et mécanismes internes utilisés dans VIRTUAL UNIVERSE PRO pour la simulation.



- Pour gérer l'affichage des éléments 3D et 2D de l'émulateur (en cours de simulation et hors simulation), VIRTUAL UNIVERSE PRO exploite un moteur de rendu 3D. Ce moteur est exécuté en parallèle des autres moteurs, et n'affecte en rien le bon déroulement de la simulation. Par défaut, ce moteur est exécuté au maximum de ses capacités, pour permettre d'optimiser les performances graphiques de la simulation (fréquence de rafraichissement graphique). Il est toutefois possible de fixer une fréquence de rafraichissement maximale à ne dépasser pour ne pas consommer « inutilement » des ressources de l'ordinateur. A tout moment, il est possible de connaître la fréquence de rafraichissement graphique courante de la simulation, grâce à l'indicateur FPS (Frame Per Second) visible en bas à gauche de la fenêtre de rendu de VIRTUAL UNIVERSE PRO.
- Pour la modélisation et la simulation des phénomènes physiques appliqués aux sprites (gravité, friction, collisions, forces..), trois moteurs physiques sont également utilisables dans VIRTUAL UNIVERSE PRO: Newton Physic (moteur par défaut), Physx Nvidia et Chrono Engine. L'utilisation du moteur physique n'est pas systématique ni obligatoire. Seuls les sprites ayant la propriété « Utilise la physique » activée sont soumis au moteur physique. Par défaut, le moteur physique est exécuté au maximum de ses capacités dans VIRTUAL UNIVERSE PRO (dans un thread séparé avec un échantillonnage variable) pour permettre le plus grand réalisme possible dans la

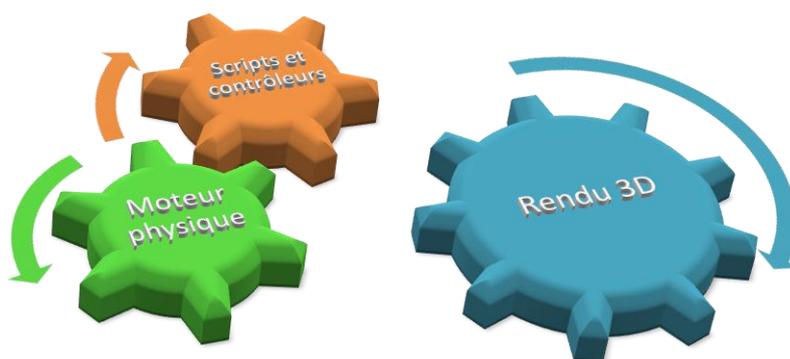
simulation des phénomènes physiques. A tout moment, il est possible de connaître la vitesse de calcul du moteur physique, grâce à l'indicateur CPS situé en bas à gauche de la fenêtre de rendu de VIRTUAL UNIVERSE PRO.

- Pour permettre aux émulateurs 3D VIRTUAL UNIVERSE PRO de communiquer en temps réel avec un contrôleur externe (type Automate Programmable Industriel), un dialogue est également assuré en cours de simulation avec ce contrôleur externe (échange de variables entrées/sorties). Différents protocoles de communication industriels sont disponibles dans VIRTUAL UNIVERSE PRO pour pouvoir se connecter et dialoguer avec différents types et marques de contrôleurs industriels (Siemens, Rockwell, Schneider,..). Par défaut, la vitesse du dialogue entre VIRTUAL UNIVERSE PRO et le contrôleur externe est maximale. Il est toutefois possible d'imposer une période minimale pour l'échange des variables d'entrée/sortie, afin de ne pas consommer « inutilement » les ressources de l'ordinateur.
- L'intelligence de simulation donnée aux objets 3D (sprites) est modélisée par un ensemble de comportements prédéfinis et de comportements personnalisés (scripts écrits en langage basic, contrôleurs virtuels programmés en ladder ou en Sfc/Fdb). Ces comportements et scripts sont tous exécutés en parallèle (dans un thread séparé).

Mode RUN/STOP de la simulation

La simulation dans VIRTUAL UNIVERSE PRO peut être en mode STOP (simulation stoppée et initialisée) ou RUN (simulation en cours).

En mode RUN, le moteur de rendu 3D, le moteur physique et le dialogue avec le logiciel externe sont activés. Les Comportements, les scripts et les contrôleurs virtuels sont actifs.



Les sprites 3D et les caméras possèdent un doublon pour certains paramètres (leurs positions par exemple). Le premier jeu de paramètre correspond aux valeurs initiales, le deuxième jeu aux valeurs courantes. En mode STOP, les valeurs initiales sont recopiées dans les valeurs courantes.

Construction d'un simulateur

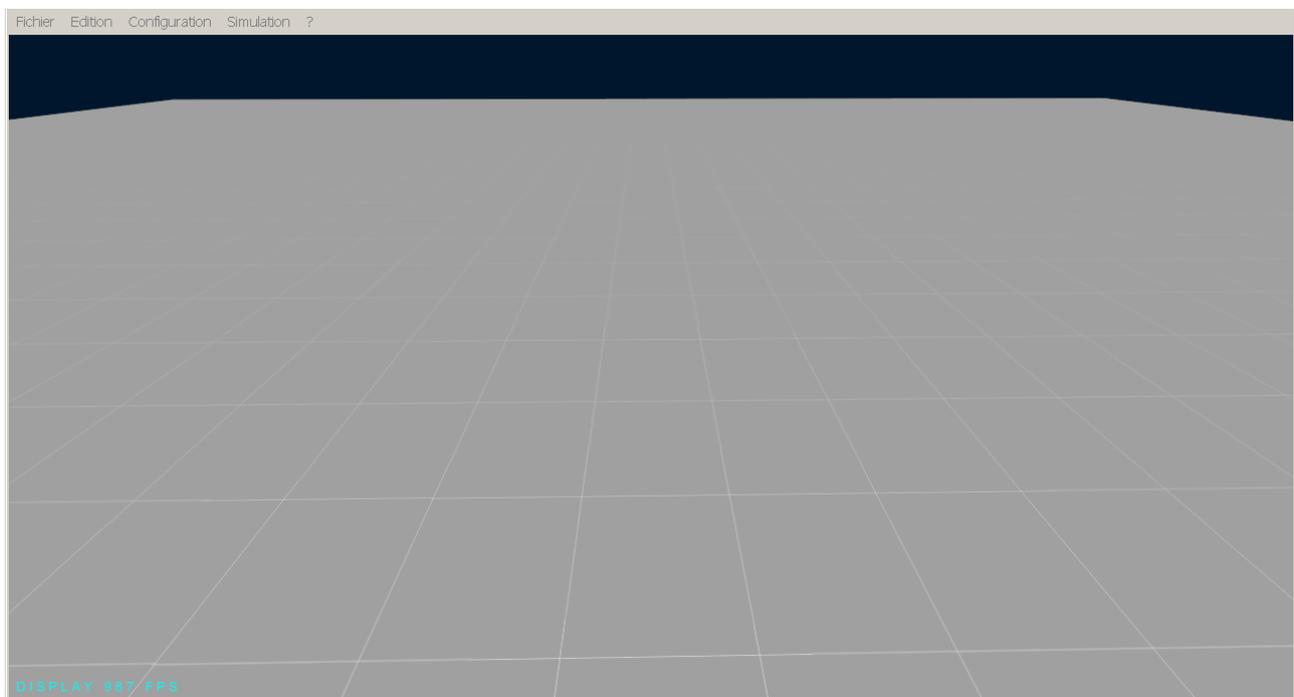
Régler les propriétés générales

Propriétés d'affichage

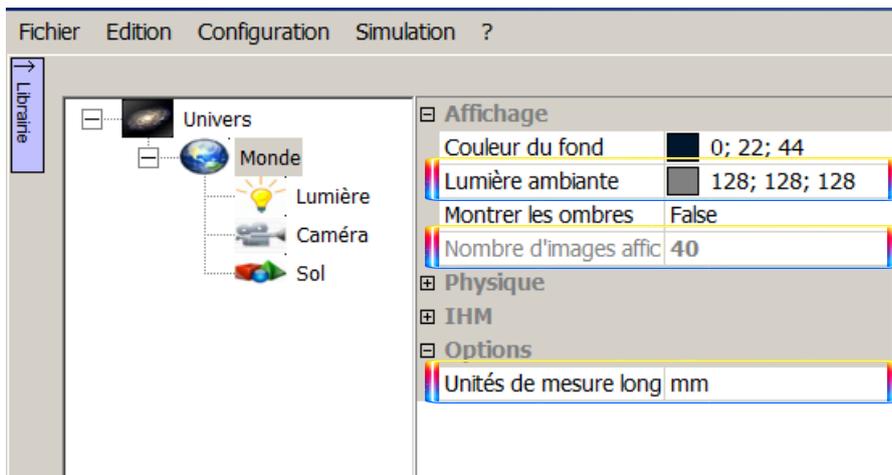
VIRTUAL UNIVERSE PRO s'ouvre automatiquement par défaut sur un modèle de monde pré-réglé en termes d'affichage.

Les unités utilisées par défaut dans ce monde sont :

- **millimètre** (longueur)
- **degré** (angle)

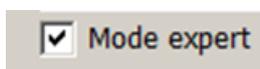
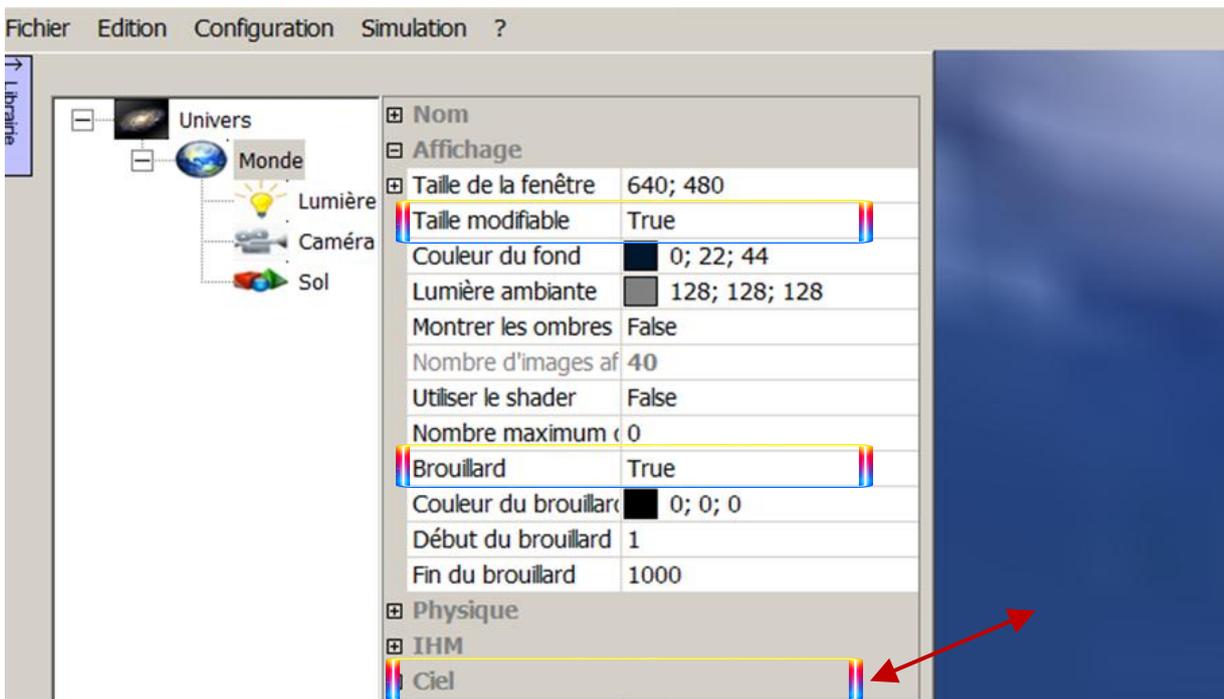


Il est possible de modifier les unités de longueur (mètres ou millimètres), la couleur du fond, ou encore d'ajouter des ombres, en accédant aux propriétés du monde.



En mode Expert, il est possible de régler de nombreux autres paramètres d'affichage, toujours dans les propriétés du monde :

- modifier les dimensions de la fenêtre du simulateur à son ouverture
- ajouter une image pour représenter un ciel au dessus du monde
- imposer au moteur de rendu 3D une fréquence de rafraîchissement graphique maximale à ne dépasser

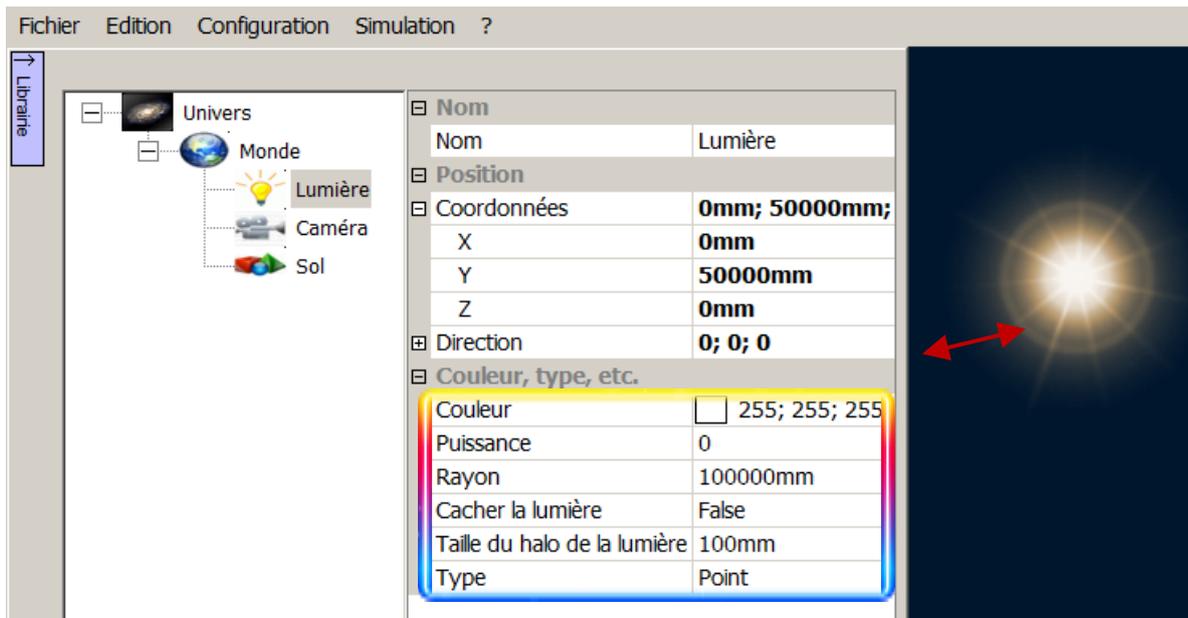


Propriétés d'éclairage

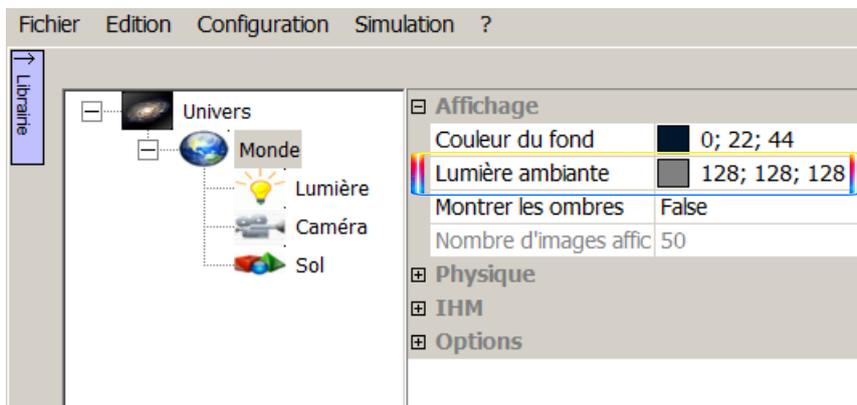
L'éclairage d'une simulation est obtenu par l'ajout et le réglage de lumières.

Le monde 3D par défaut possède déjà une lumière pré-réglée. Cette lumière, de couleur blanche, est située à 50 mètres au dessus du sol et possède un rayon de 100 mètres.

Il est possible de modifier les propriétés de cette lumière, tout comme il est possible d'ajouter d'autres lumières dans la simulation.



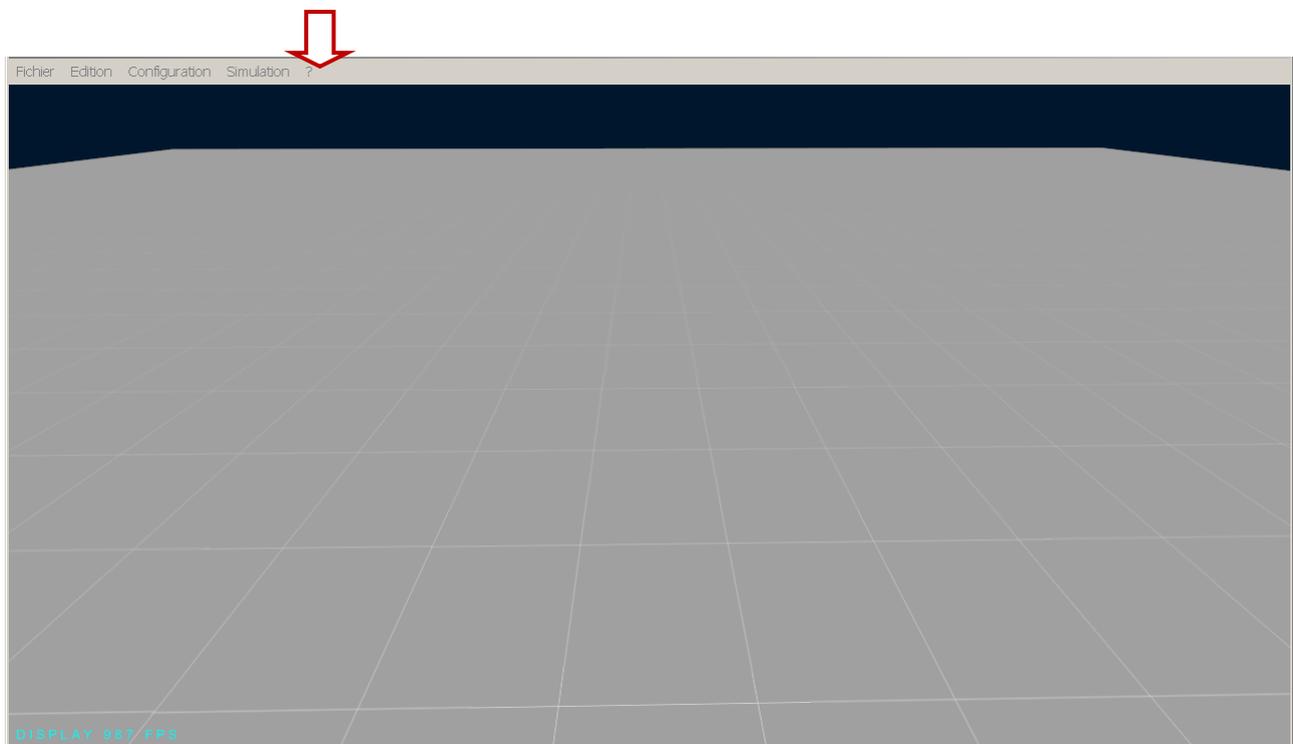
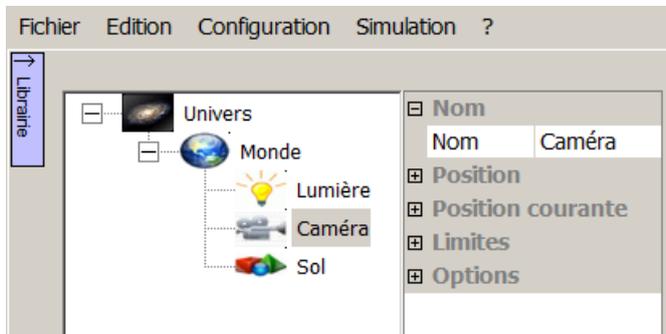
Un autre paramètre permet de régler la lumière ambiante du monde par défaut (sa couleur et son intensité), indépendamment des lumières ajoutées dans la simulation. Ce paramètre se situe dans les propriétés du monde.



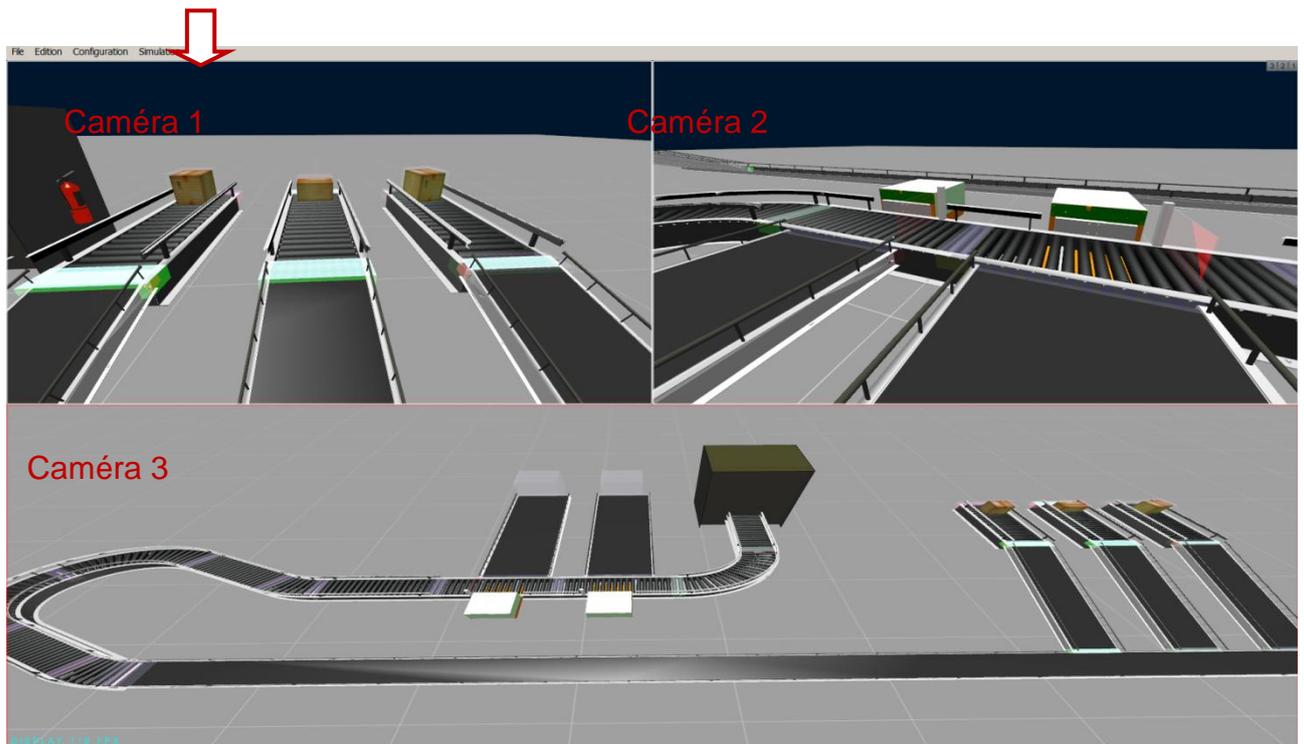
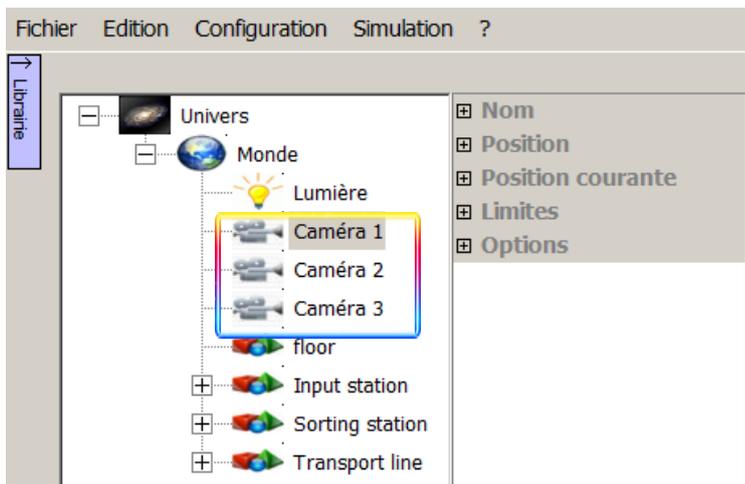
Propriétés de visualisation

La visualisation dans une simulation est gérée par l'usage de caméras (points de vue).

Par défaut, la simulation possède une seule caméra fixe (un seul point de vue), dont l'objectif est fixé vers le centre du monde.



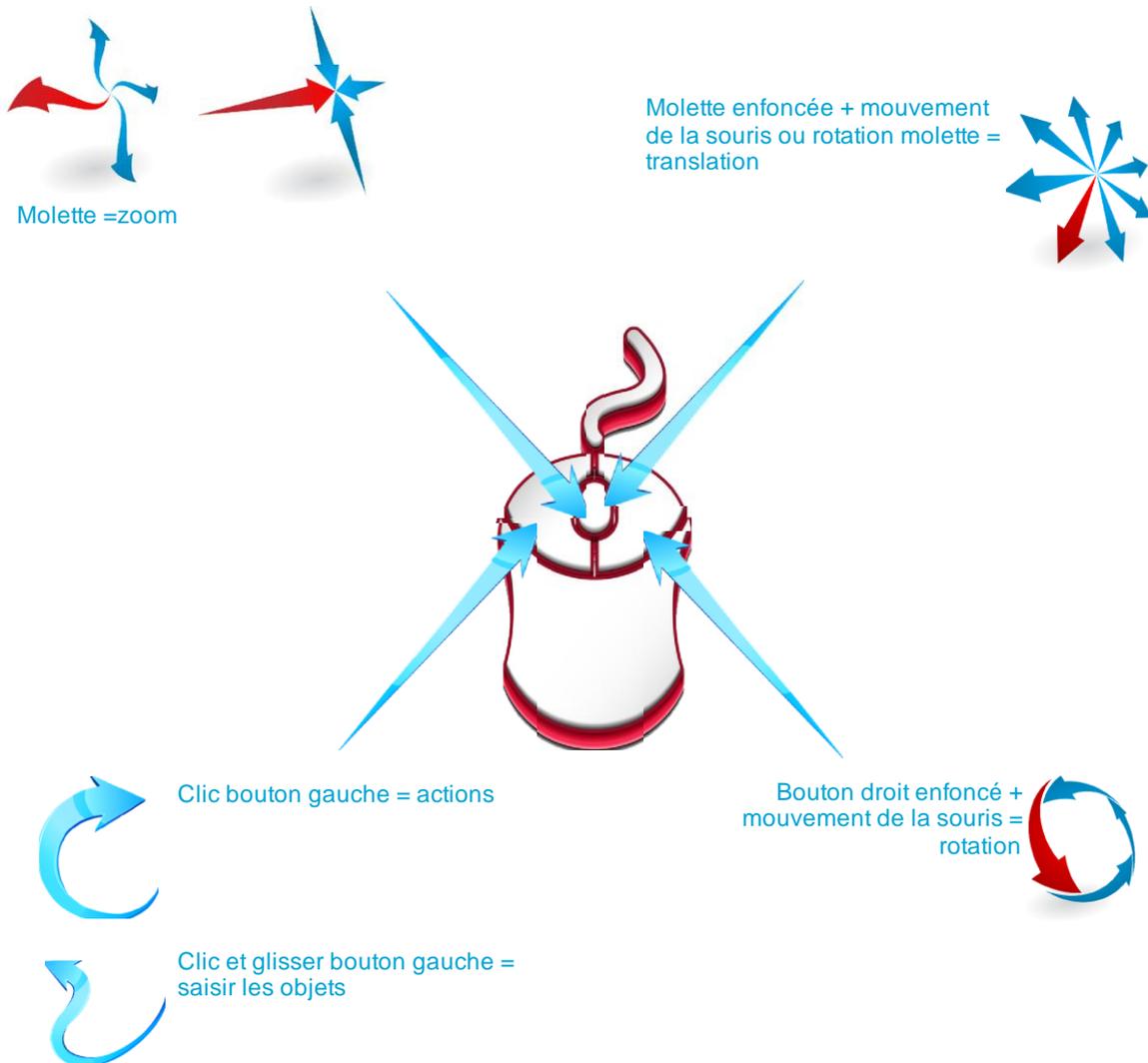
Il est possible d'ajouter d'autres caméras (fixes ou mobiles) et ainsi de créer des points de vue supplémentaires à l'intérieur de la simulation.



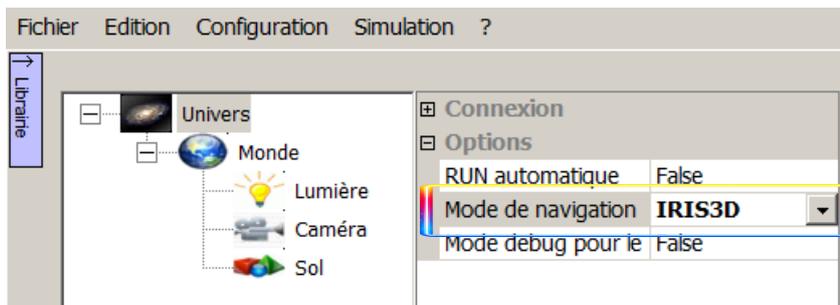
Attention ! Ce mode de visualisation multi-caméras peut réduire fortement les performances graphiques (fréquence de rafraichissement).

Propriétés de navigation

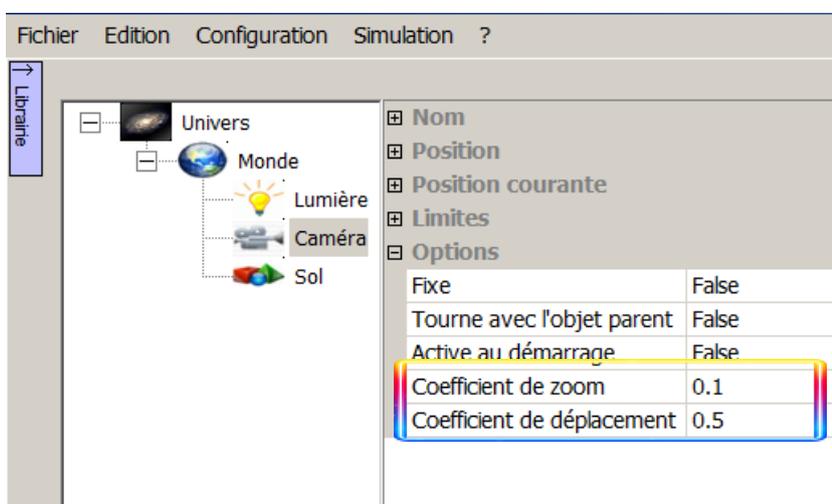
Par défaut, la navigation à l'intérieur d'une simulation se fait en utilisant la souris et ses boutons (mode de navigation VU).



Il est possible d'utiliser un autre mode de navigation (appelé IRIS3D) accessible dans les propriétés de l'univers. Ce mode de navigation utilise des boutons flèches pour se déplacer et zoomer dans le monde 3D. Dans ce mode, la navigation à la souris reste possible.

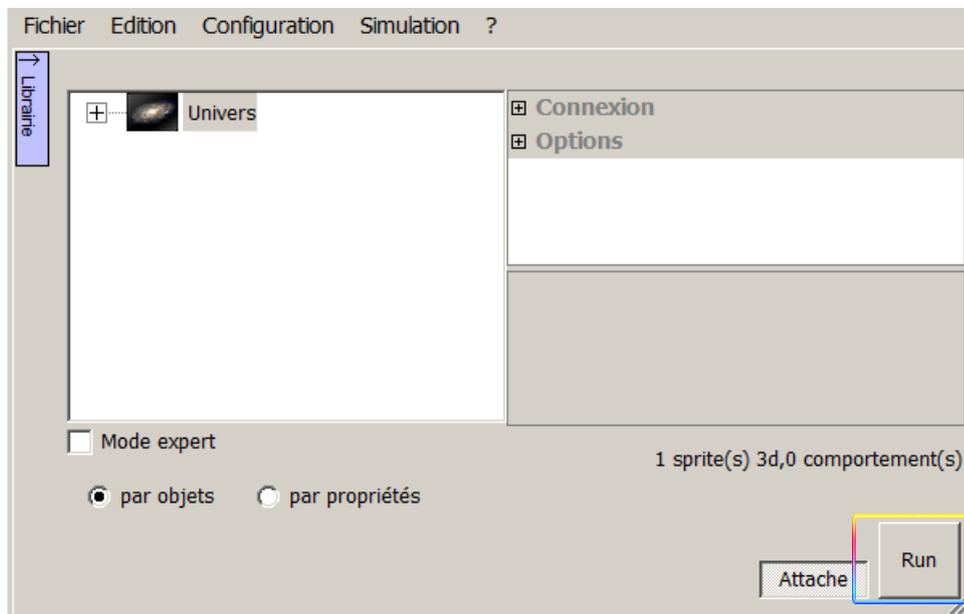


Enfin, il est possible de régler les coefficients de déplacement et de zoom, pour se déplacer ou zoomer plus ou moins vite à l'intérieur du rendu 3D (quelque soit le mode de navigation utilisé, VU ou IRIS3D). Ces coefficients sont propres à la caméra et sont accessibles dans ses propriétés.

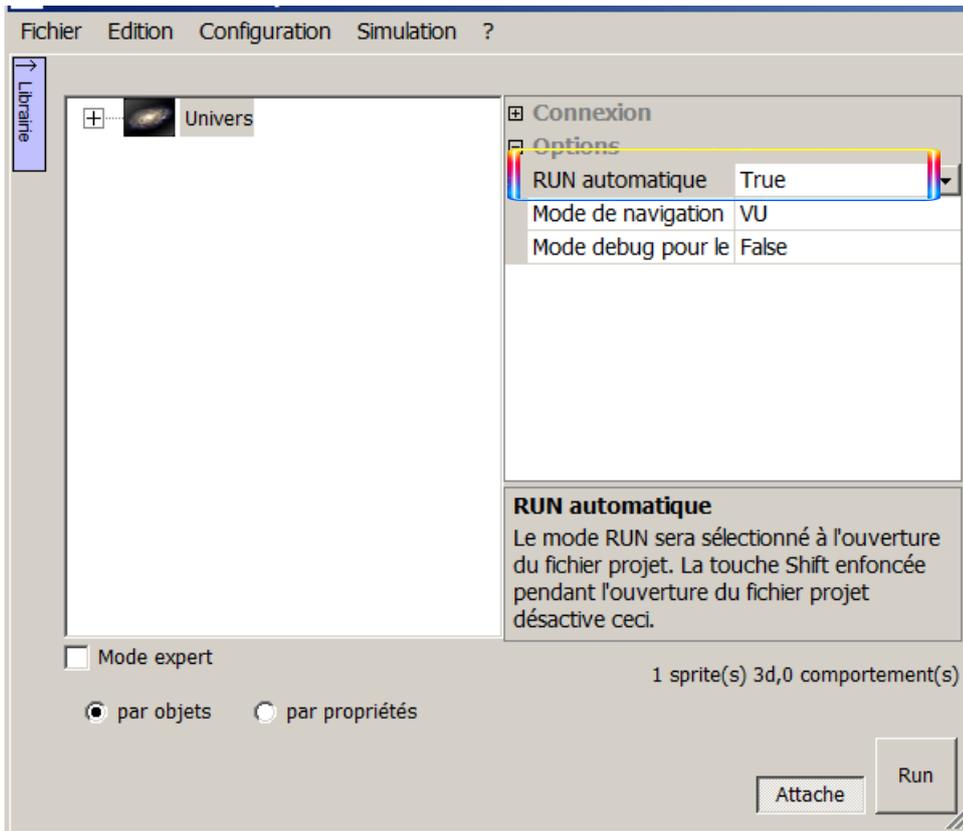


Options de simulation

Par défaut à l'ouverture, la simulation dans VIRTUAL UNIVERSE PRO est en mode STOP. Le lancement de la simulation (passage en mode RUN) s'obtient à tout moment en appuyant sur le bouton « RUN ».



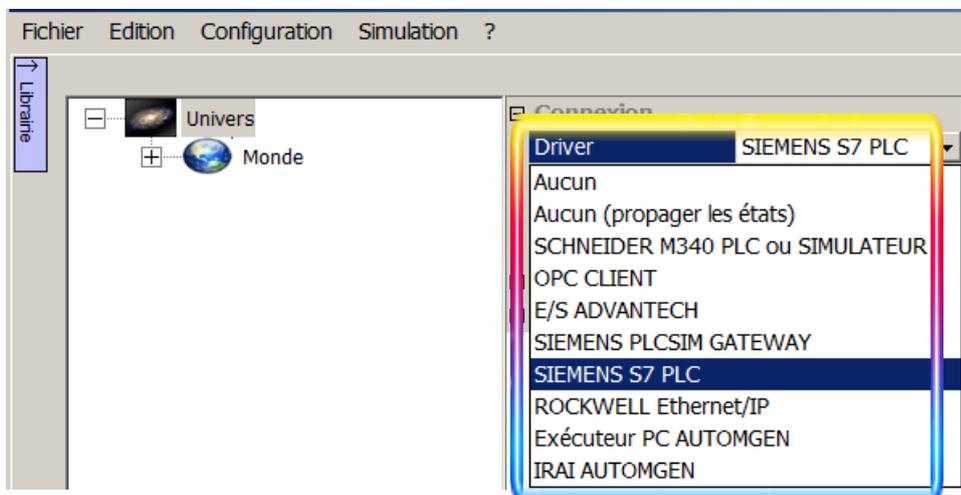
Il est possible de lancer automatiquement la simulation dès l'ouverture du projet VIRTUAL UNIVERSE PRO, en sélectionnant l'option « Run automatique » dans les propriétés de l'Univers.



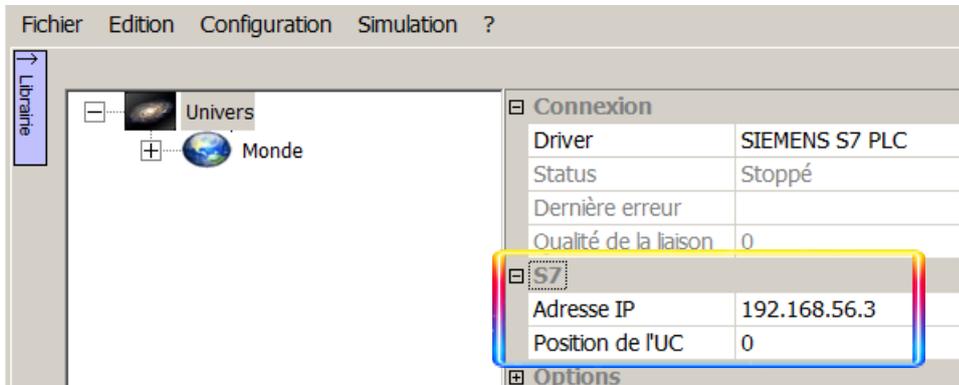
Dans le projet ouvert par défaut, VIRTUAL UNIVERSE PRO n'est connecté à aucun logiciel/contrôleur externe.

Pour connecter la simulation VIRTUAL UNIVERSE PRO à un logiciel/contrôleur externe, il est nécessaire de sélectionner un « Driver » dans les propriétés de l'Univers. Au lancement de la simulation (passage en mode RUN), un dialogue sera établi entre VIRTUAL UNIVERSE PRO et le logiciel/contrôleur externe.

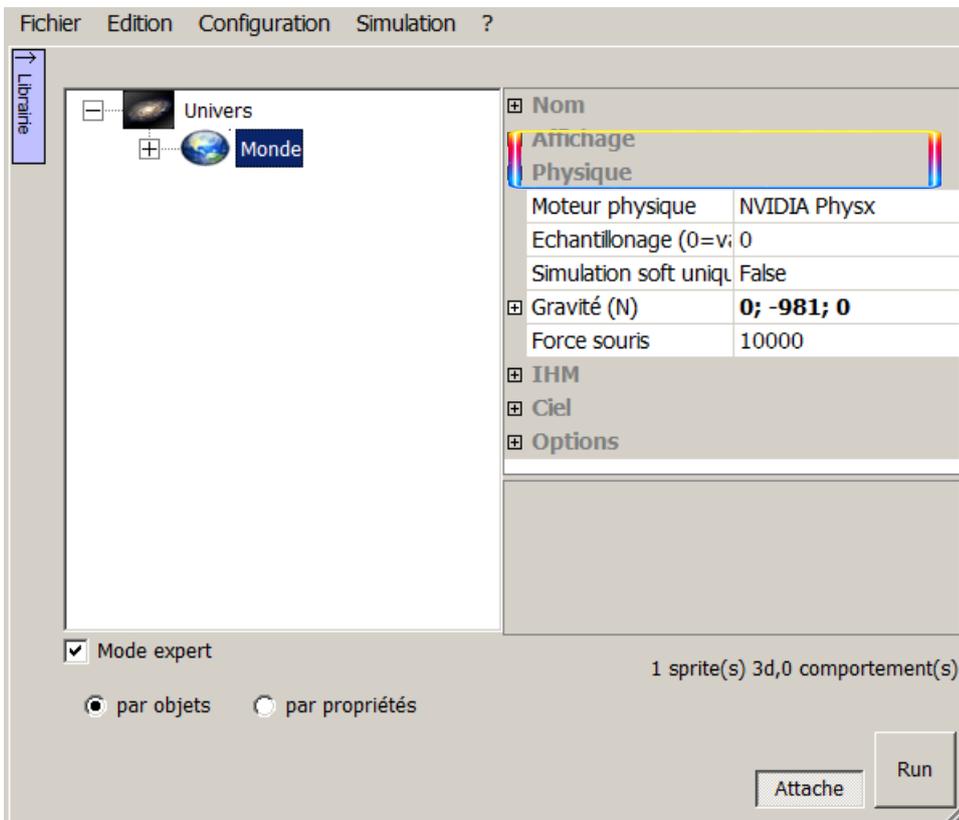
VIRTUAL UNIVERSE PRO est compatible avec les plus grandes marques d'Automate Programmable Industriel et propose plusieurs types de drivers (pour différents protocoles de communication Automate).



Une fois le « Driver » sélectionné, le paramétrage de la connexion (adresse IP de l'automate, nom du serveur OPC, position de l'UC sur le rack...) s'effectue dans l'onglet propre au driver.



En mode Expert, et pour les utilisateurs avertis, il est possible de changer de moteur physique pour la simulation. Dans les propriétés du Monde, il est en effet possible de choisir le moteur Nvidia Physx ou Chrono Engine à la place de Newton Physics (moteur utilisé par défaut).



Importer et alléger les modèles CAO 3D

Importer des modèles CAO 3D

VIRTUAL UNIVERSE PRO permet de réutiliser les modèles 3D issus des logiciels de CAO (Conception Assistée par Ordinateur) en les important directement depuis le logiciel de CAO, ou indirectement par l'intermédiaire d'un fichier d'échange.

Voici les types de données CAO qu'il est possible d'importer et de réutiliser aujourd'hui dans VIRTUAL UNIVERSE PRO :

Logiciel de CAO	Fichiers d'échange (import indirect)
DS SolidWorks	3DXml
Autodesk Inventor	
DS Delmia	3DXml
DS Catia	3DXml

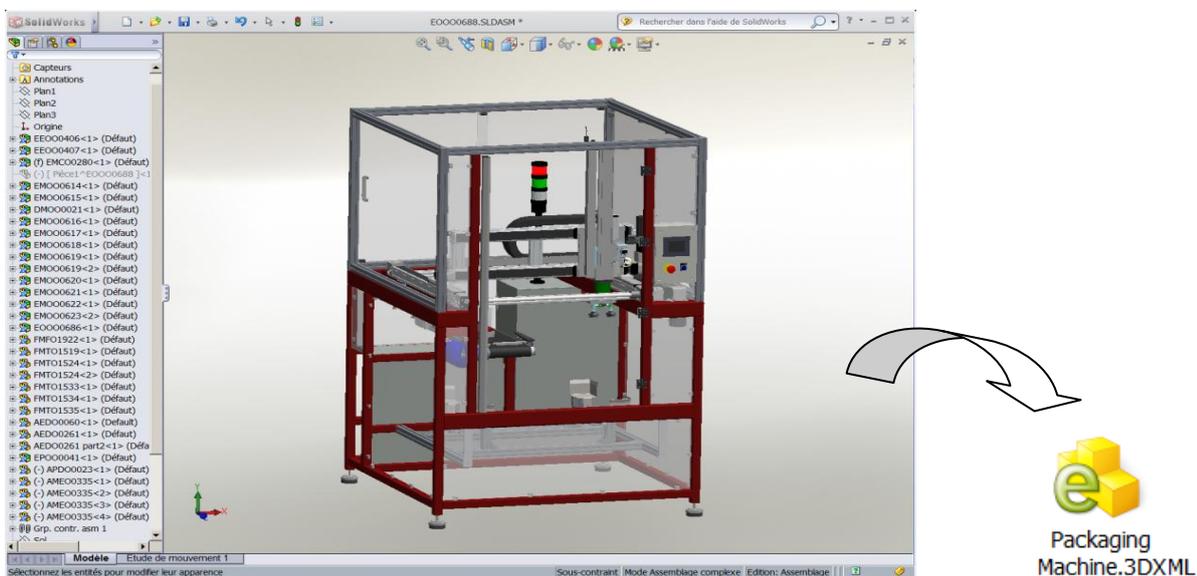
Importer des modèles 3D issus de SolidWorks

Avec cette méthode d'importation, la donnée SolidWorks est d'abord exportée et enregistrée sous forme d'un fichier 3DXML. Ce fichier 3DXML est ensuite importé dans VIRTUAL UNIVERSE PRO.

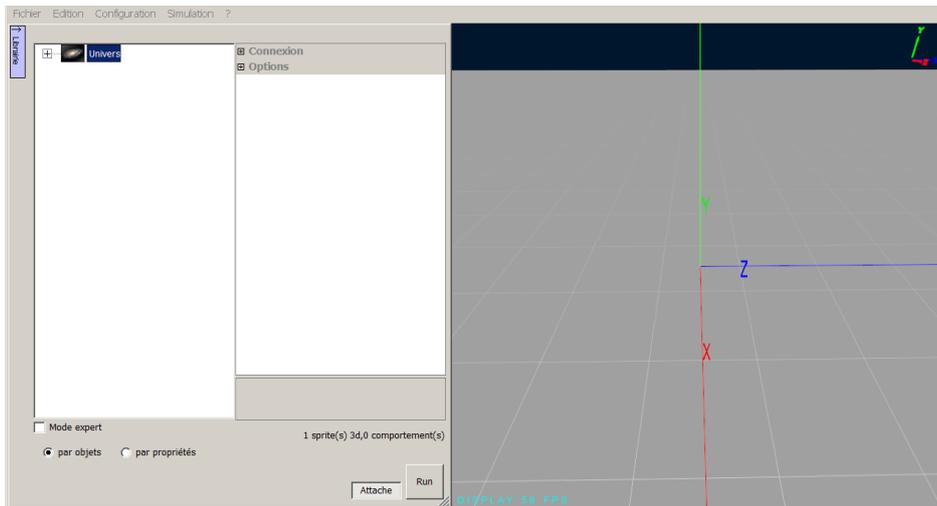


L'intérêt de cette méthode est de ne pas contraindre l'utilisateur de VIRTUAL UNIVERSE PRO à disposer du logiciel SolidWorks sur son ordinateur.

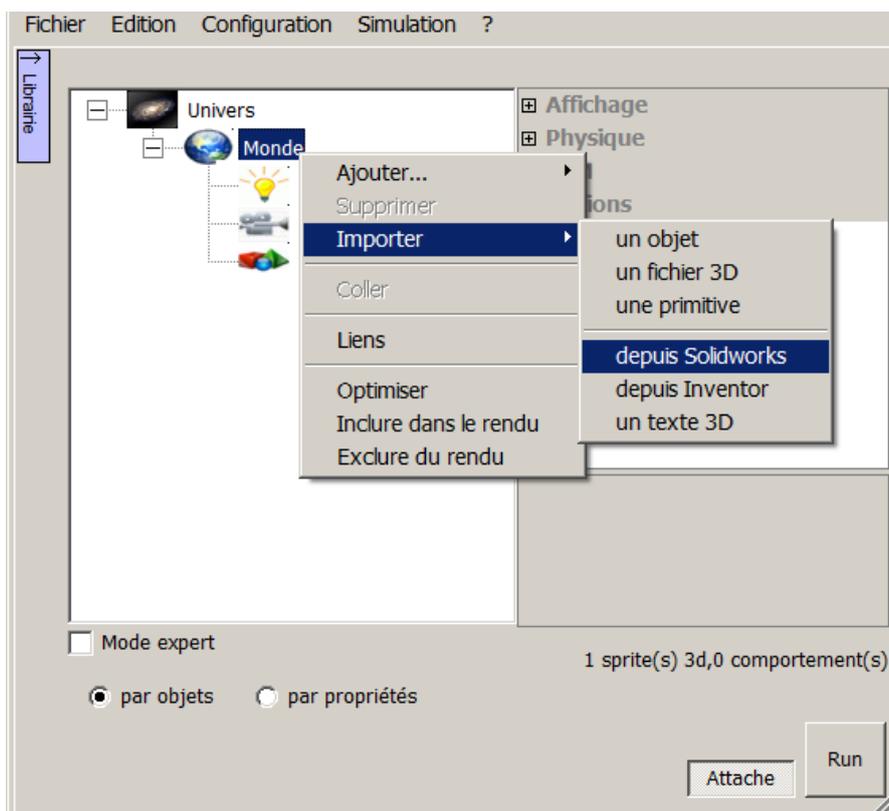
- Au préalable, la donnée 3D complète aura été ouverte dans SolidWorks, puis enregistrée dans un fichier 3DXML.



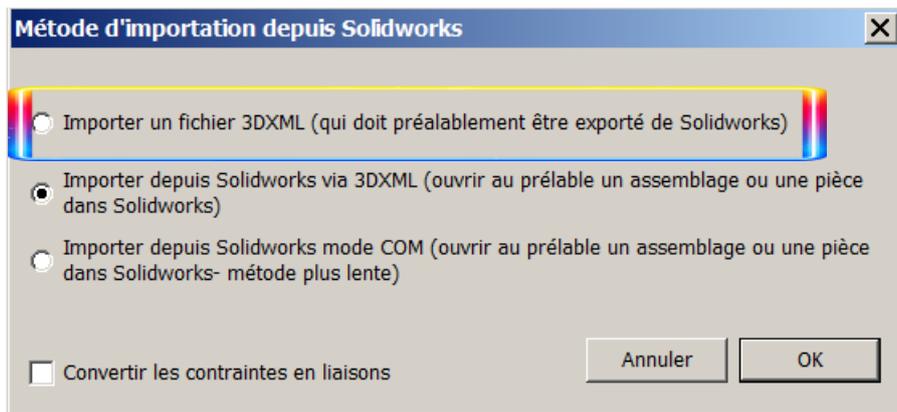
- Dans VIRTUAL UNIVERSE PRO, ouvrez la fenêtre de configuration.



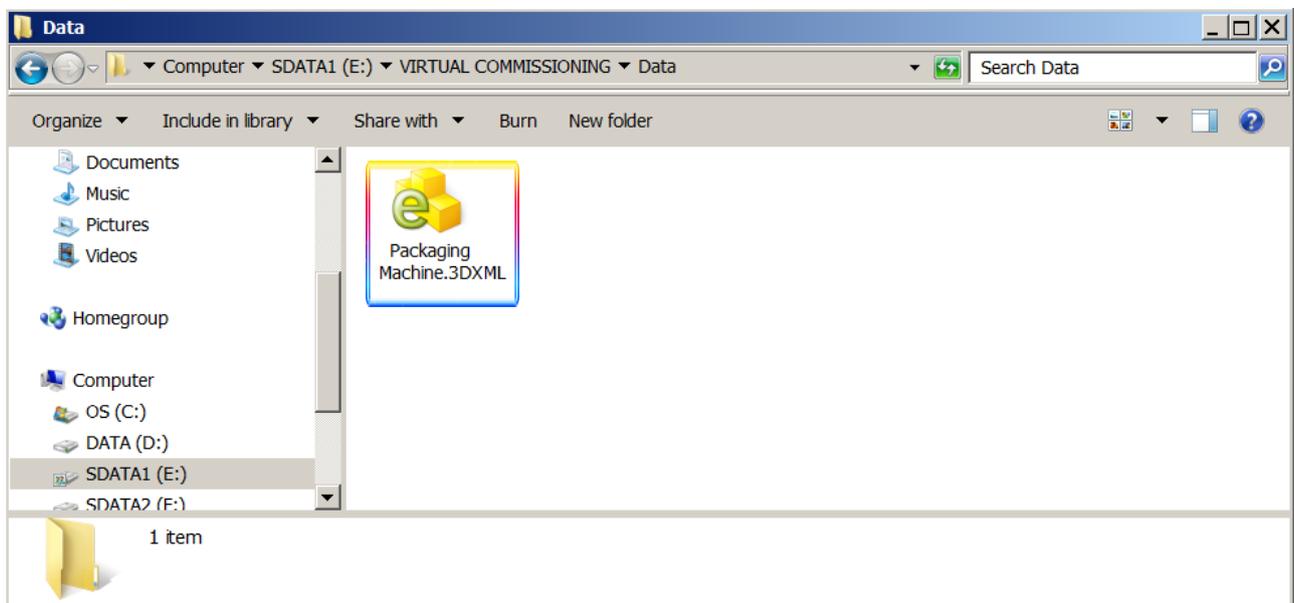
- Sélectionnez le niveau Monde, et cliquez droit pour accéder au menu Importer/depuis SolidWorks. L'importation des données peut également être faite au niveau d'un sprite, quelle que soit sa position dans l'arborescence du projet.



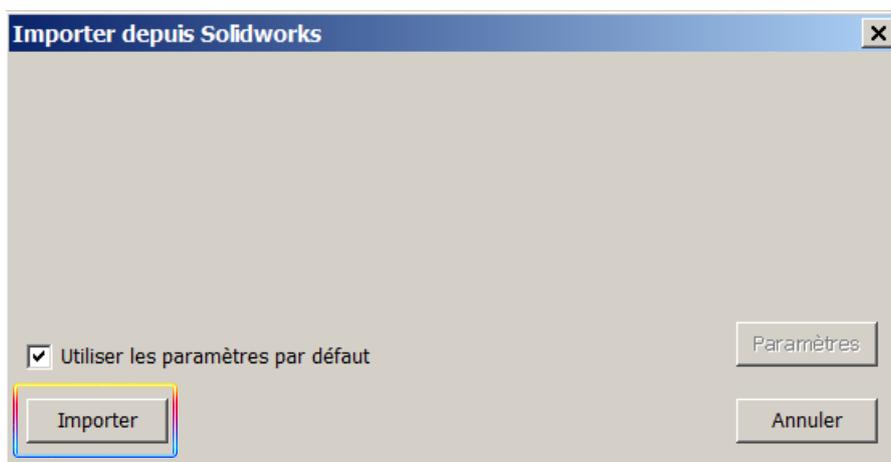
- Une boîte de dialogue s'ouvre présentant plusieurs méthodes d'importation. Choisissez la 1ère méthode « Importer un fichier 3DXML ».



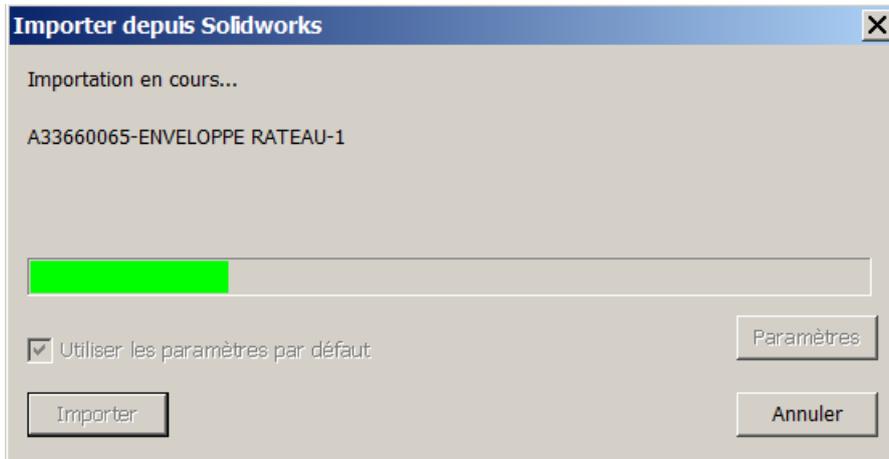
- Une fenêtre d'explorateur s'ouvre, vous pouvez sélectionner le fichier 3DXML à importer.



- Cliquez ensuite sur le bouton « Importer ».

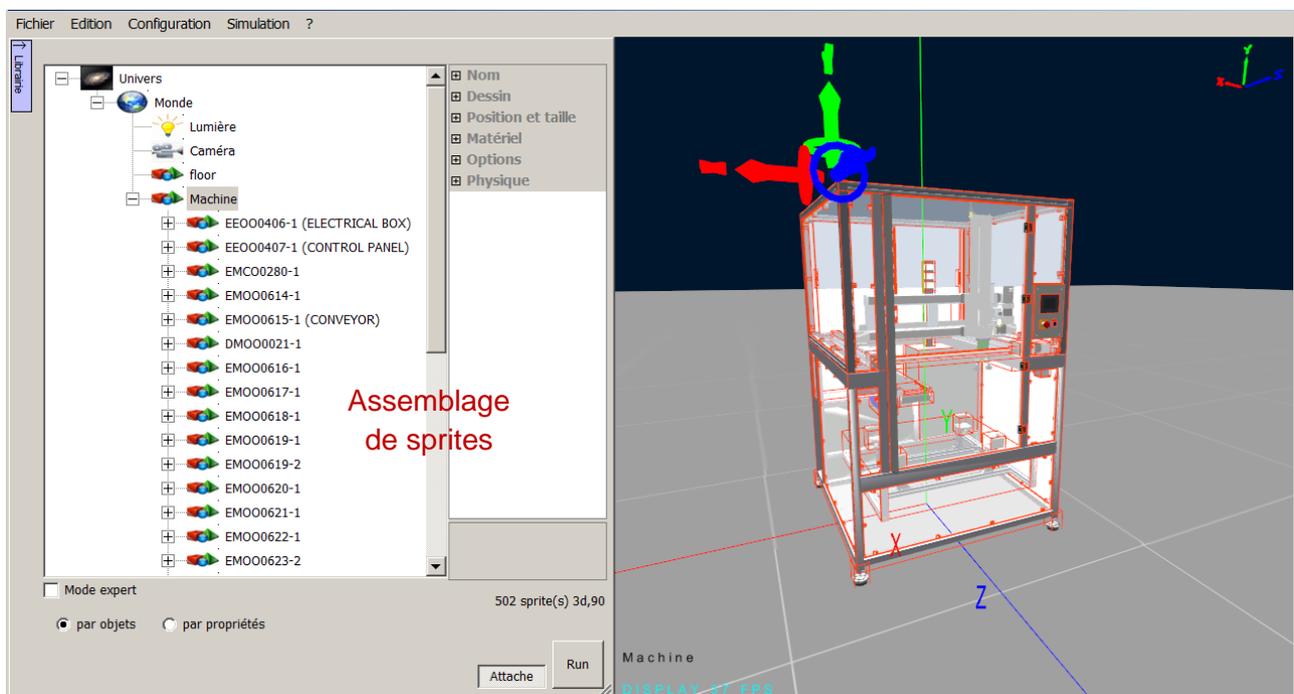


- La procédure d'importation est alors lancée. Une barre indique la progression de l'import. Le temps d'attente va de quelques secondes à quelques minutes, selon la taille du projet importé.



La donnée 3D est alors importée dans VIRTUAL UNIVERSE PRO sous la forme d'une arborescence d'objets 3D (sprites), accessible dans la fenêtre de configuration.

Cette arborescence respecte la structure initiale de la donnée SolidWorks (composée de pièces et d'assemblages). Dans le cas d'un assemblage, chaque pièce composant l'assemblage est importée comme un objet 3D (sprite) indépendant dans VIRTUAL UNIVERSE PRO. Les couleurs et textures initiales sont également récupérées dans VIRTUAL UNIVERSE PRO.



Importer des modèles 3D issus de DS Delmia ou Catia

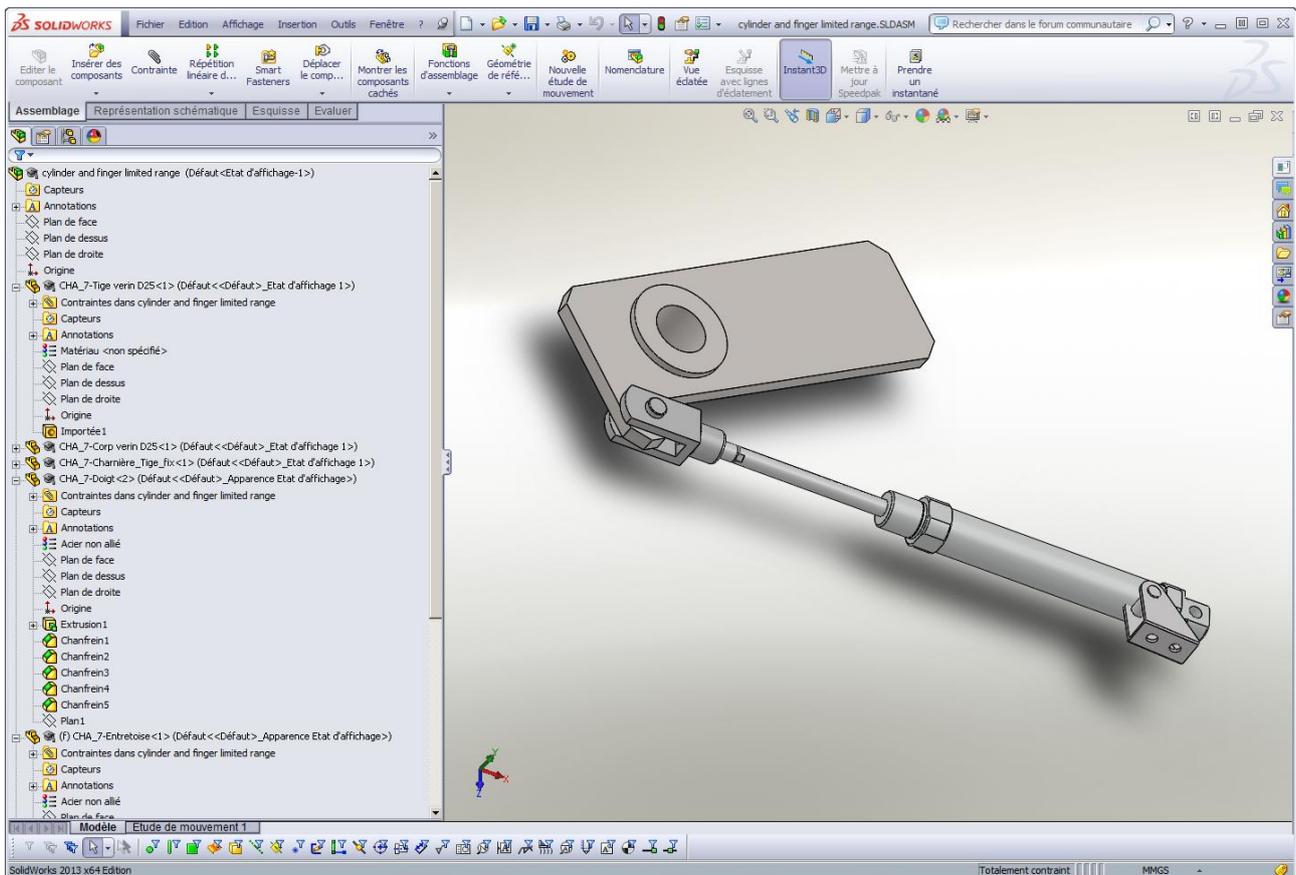
La procédure est identique à celle décrite précédemment pour Solidworks : export au format 3DXML depuis Catia ou Delmia puis import dans VIRTUAL UNIVERSE PRO.

Importer des modèles 3d de Solidworks avec des contraintes

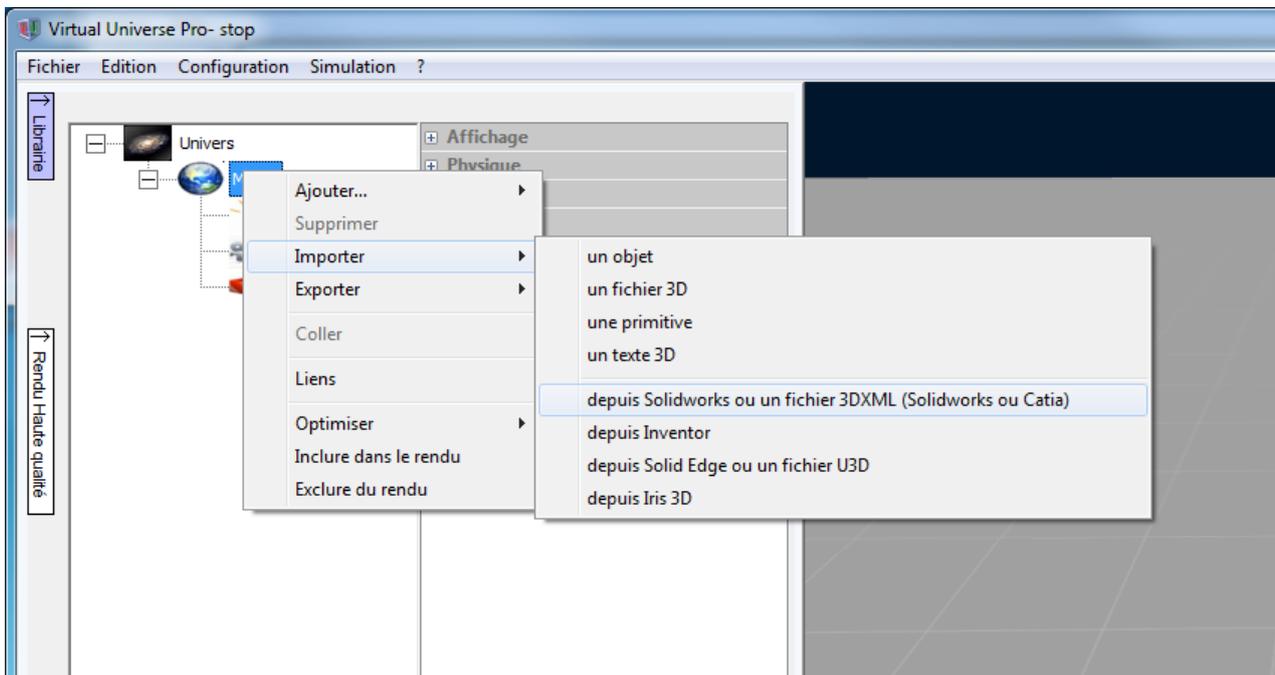
Ce mode permet d'importer, en plus des géométries et de la structure, les contraintes définies dans Solidworks. Ce mode nécessite l'utilisation de Chrono Engine comme moteur physique.

L'assemblage doit être ouvert dans Solidworks sur le même PC que VIRTUAL UNIVERSE PRO.

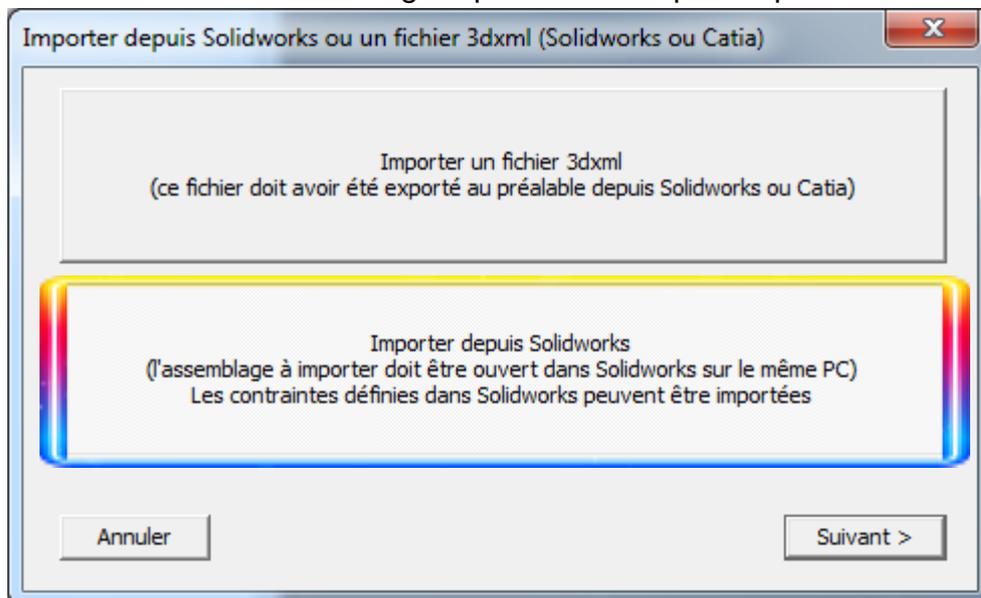
- Ouvrir l'assemblage dans Solidworks



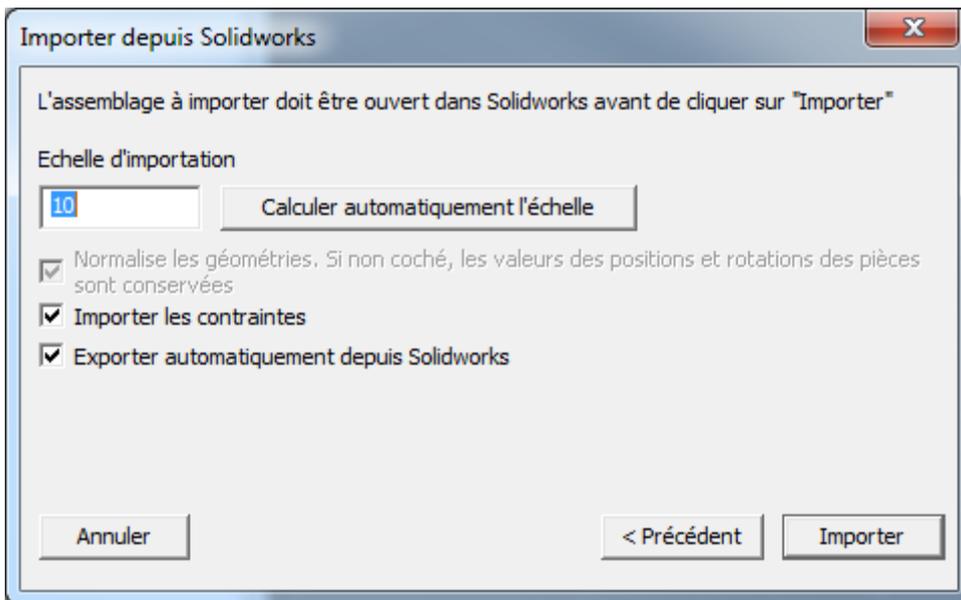
- Faire un clic droit sur le monde ou un sprite 3d et choisir le menu Importer/Depuis Solidworks



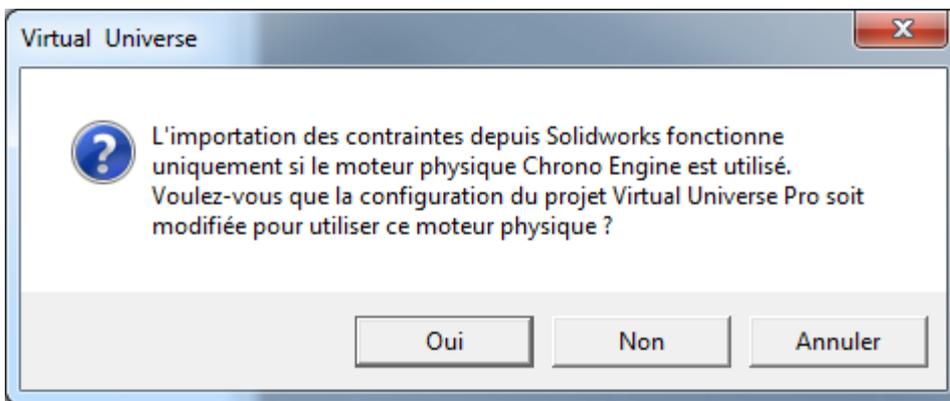
- Choisir dans la boîte de dialogue qui s'ouvre "Import depuis Solidworks"



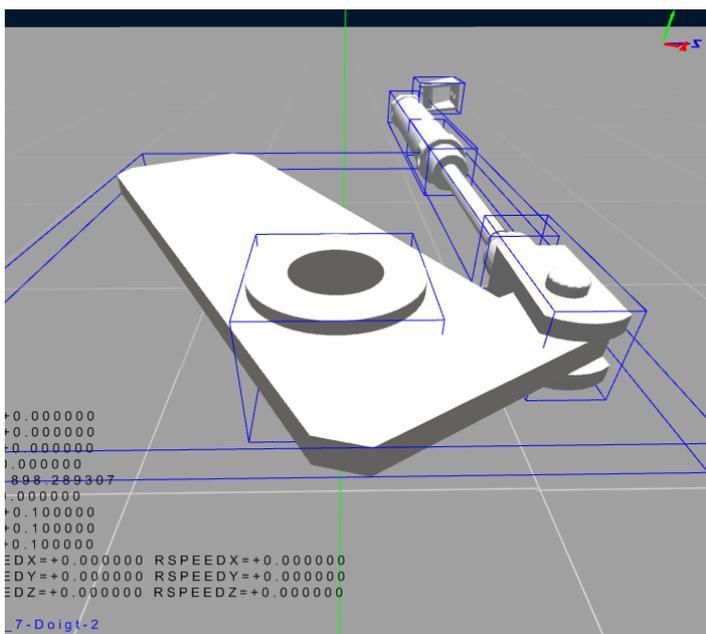
- Cliquer sur "Suivant".



- Puis sur "Importer".

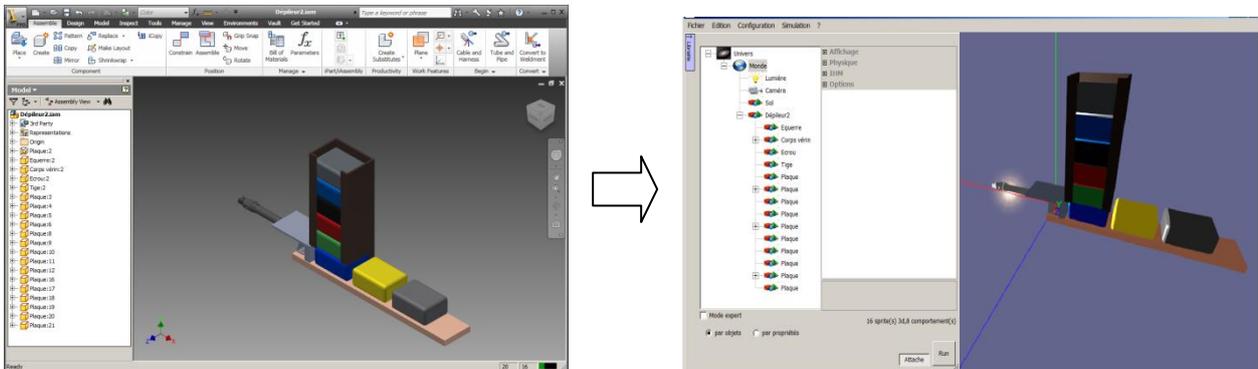


- Cliquer sur "Oui".

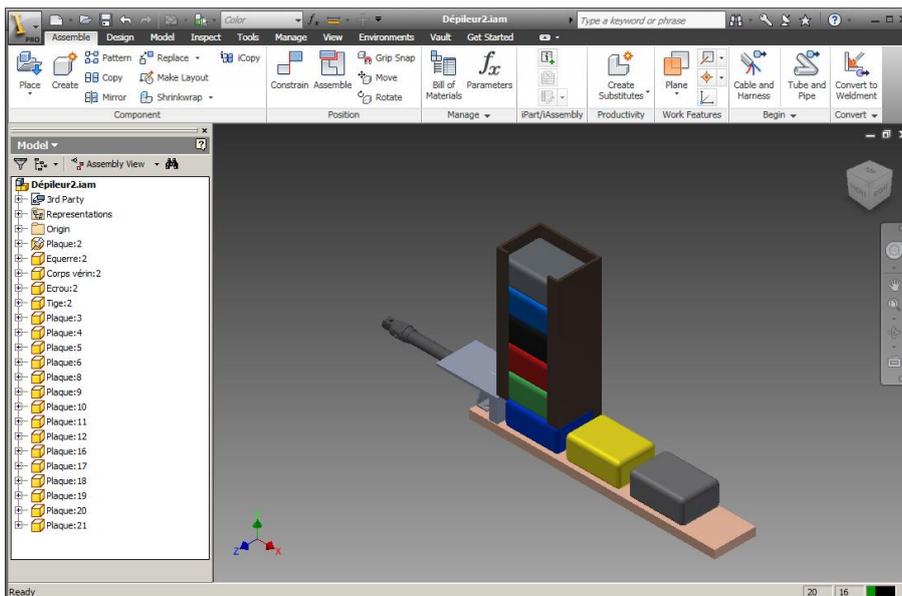


Importer des modèles 3D issus d'Autodesk Inventor

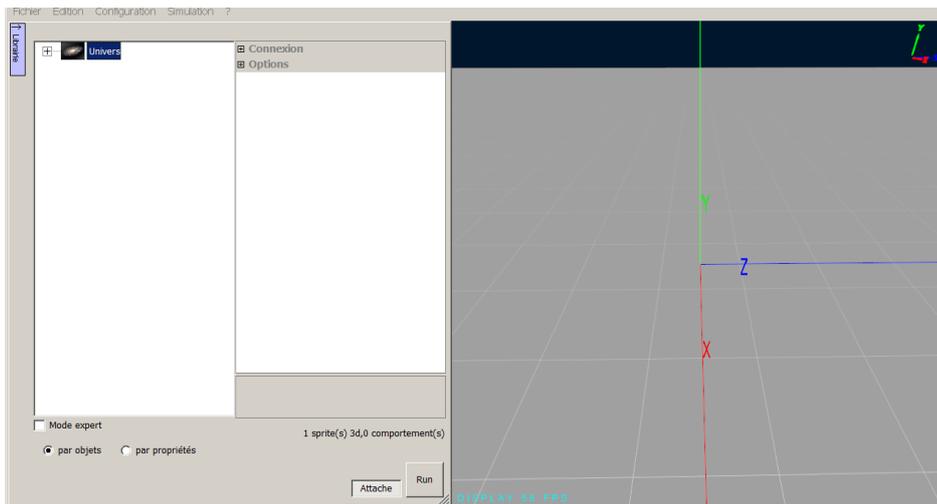
Cette méthode nécessite d'avoir le logiciel Autodesk Inventor ouvert simultanément avec VIRTUAL UNIVERSE PRO sur le même ordinateur.



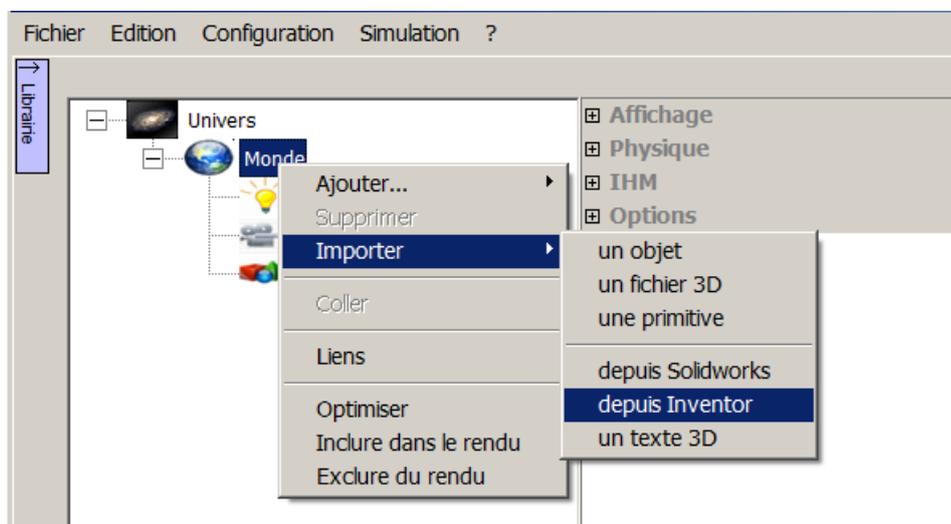
- Tout d'abord, ouvrez la donnée 3D complète dans le logiciel Autodesk Inventor.



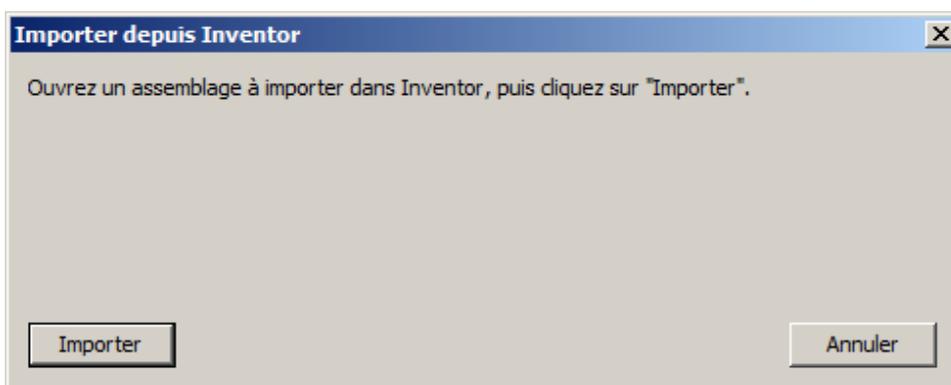
- Dans VIRTUAL UNIVERSE PRO, ouvrez la fenêtre de configuration.



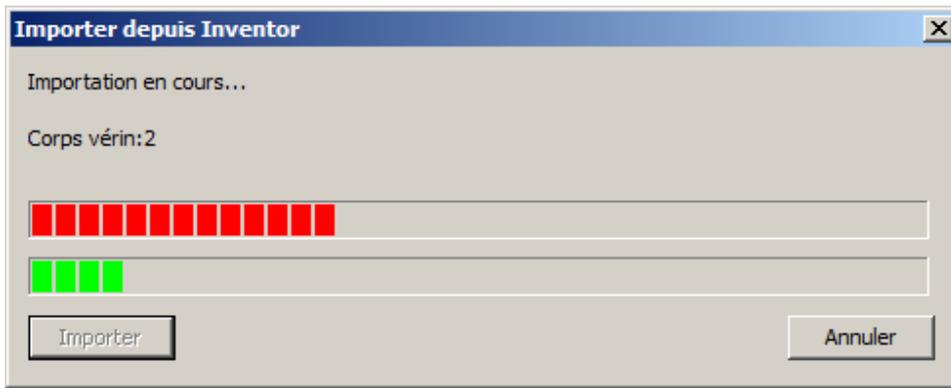
- Sélectionnez le niveau Monde, et cliquez droit pour accéder au menu Importer/depus Inventor. L'importation des données peut également être faite au niveau d'un sprite, quelle que soit sa position dans l'arborescence du projet.



- Cliquez ensuite sur le bouton « Importer ».

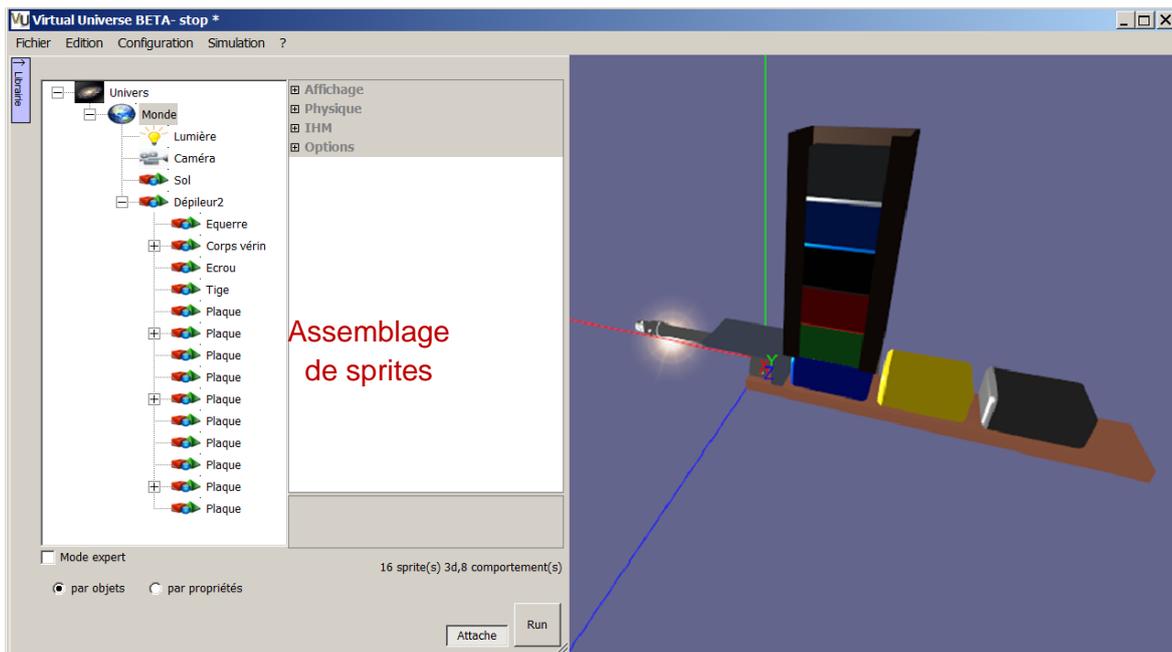


- La procédure d'importation est alors lancée. Une barre indique la progression de l'import. Le temps d'attente va de quelques secondes à quelques minutes, selon la taille du projet importé.



La donnée 3D est alors importée dans VIRTUAL UNIVERSE PRO sous la forme d'une arborescence d'objets 3D (sprites), accessible dans la fenêtre de configuration.

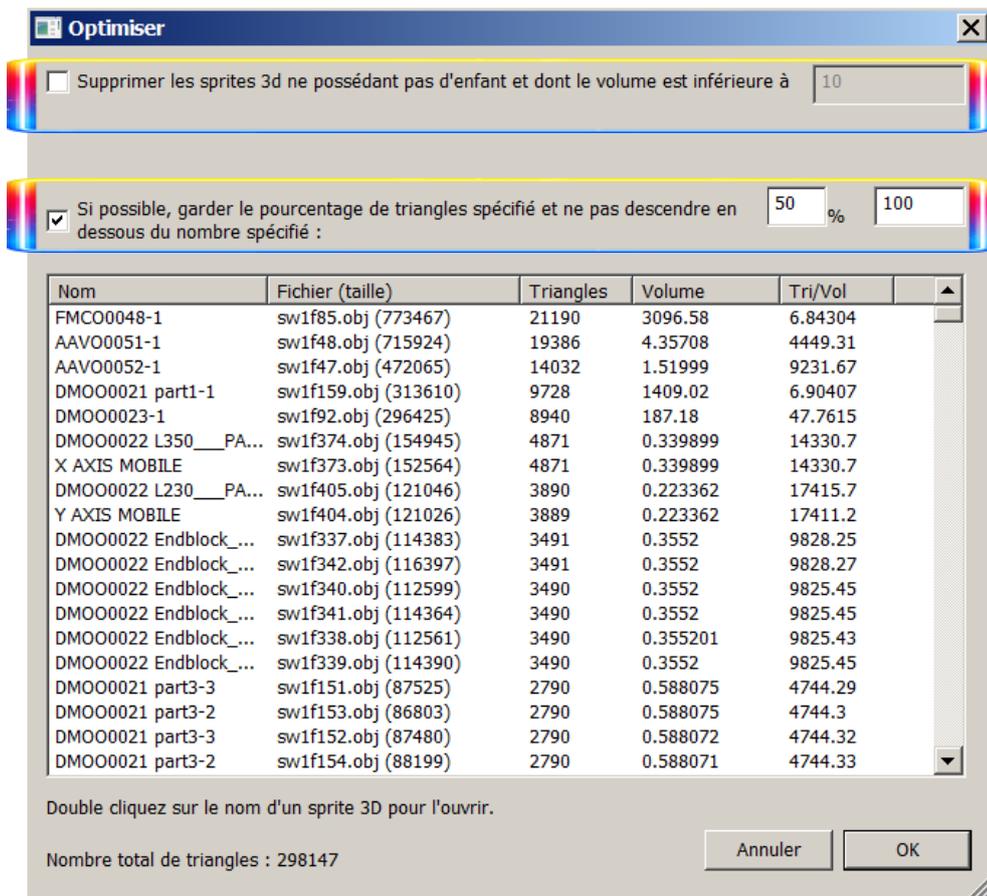
Cette arborescence respecte la structure initiale de la donnée Inventor (composée de pièces et d'assemblages). Dans le cas d'un assemblage, chaque pièce composant l'assemblage est importée comme un objet 3D (sprite) indépendant dans VIRTUAL UNIVERSE PRO. Les couleurs et textures initiales sont également récupérées dans VIRTUAL UNIVERSE PRO.



Alléger les modèles CAO 3D

Les modèles 3D issus des logiciels de CAO possèdent souvent des géométries 3D très complexes (nombre de triangles) et des petites pièces qui ne sont pas toujours nécessaires aux émulateurs 3D VIRTUAL UNIVERSE PRO et peuvent au contraire contribuer à diminuer fortement les performances graphiques.

Pour alléger les modèles CAO 3D, VIRTUAL UNIVERSE possède un outil d'optimisation des géométries 3D qu'il est recommandé d'utiliser systématiquement.

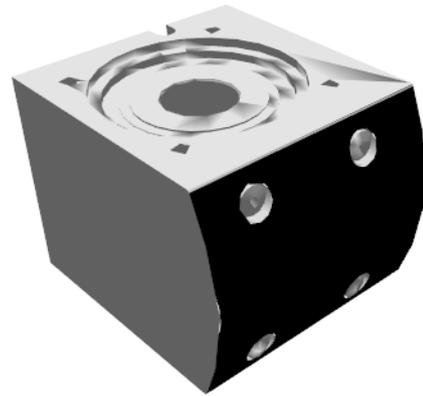
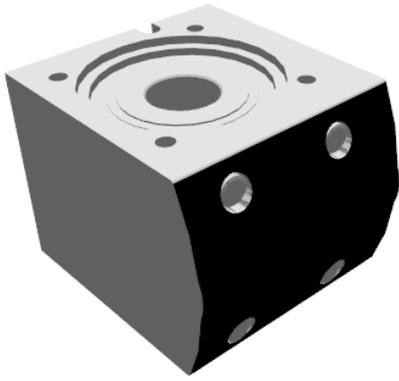


Suppression des petites pièces 3D

Réduction du nombre de triangles

Cet outil d'optimisation permet de :

- Diminuer le nombre de triangles des sprites 3D : sur l'ensemble des sprites 3D sélectionnés, l'outil applique un pourcentage de réduction du nombre de triangles, en respectant un seuil minimal (nombre minimal de triangles) à ne pas dépasser.



Donnée d'origine :

6 982 triangles

Donnée optimisée:

2 093 triangles (- 70 %)



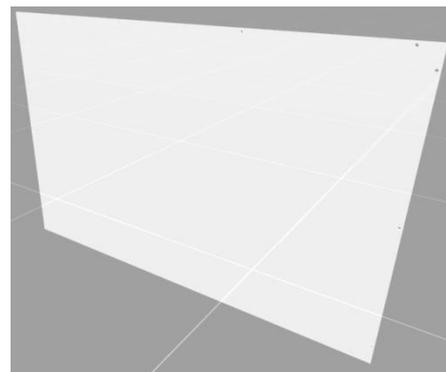
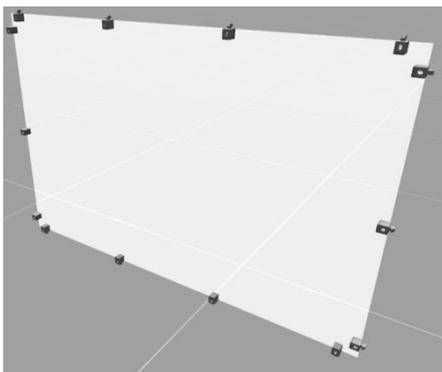
Donnée d'origine :

19 456 triangles

Donnée optimisée:

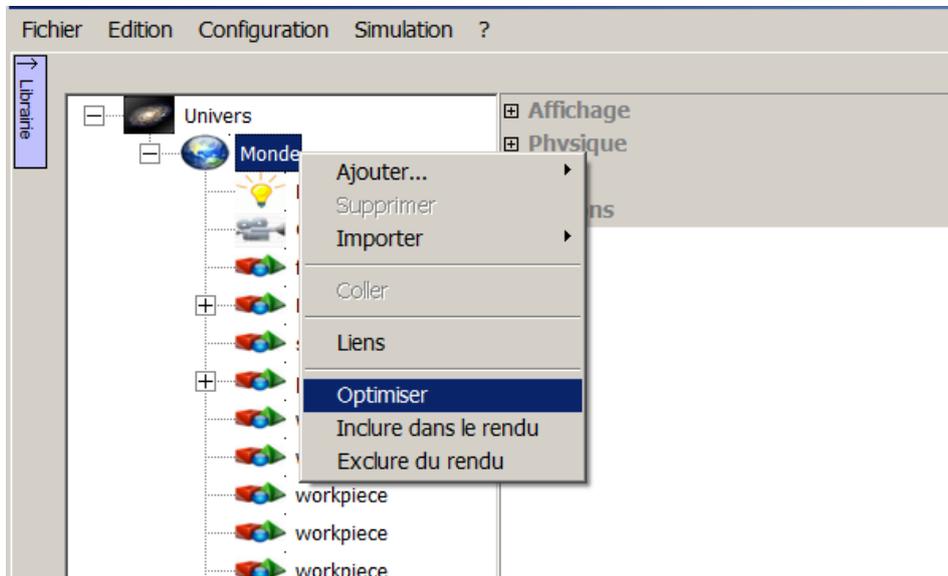
2 234 triangles (- 89 %)

- Supprimer les petites pièces 3D inutiles : sur l'ensemble des sprites 3D sélectionnés, l'outil supprime les pièces 3D dont la taille (le volume) est inférieure à un seuil fixé.

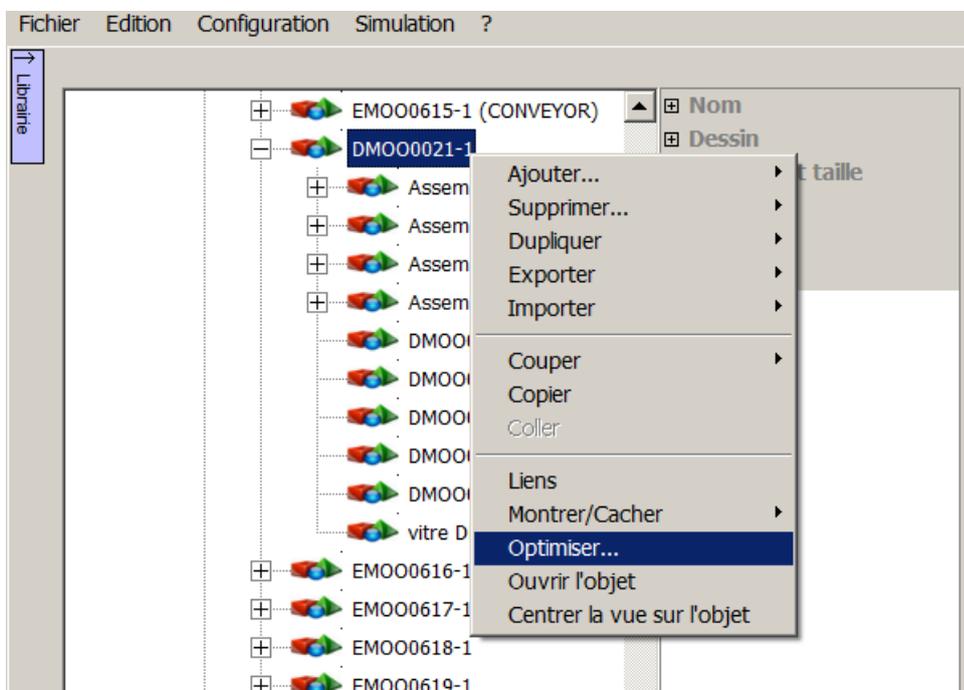


Cet outil d'optimisation est accessible à 2 niveaux :

- au niveau du Monde : l'optimisation s'applique ainsi à l'ensemble des sprites 3D du projet de simulation.



- au niveau d'un sprite 3D : l'optimisation s'applique ainsi uniquement au sprite sélectionné et à l'ensemble de ses sprites 3D enfants.



Pour plus d'information sur la mesure et l'optimisation des performances graphiques d'une simulation, voir [Mesurer les performances graphiques](#).

Concevoir des ressources et systèmes 3D intelligents

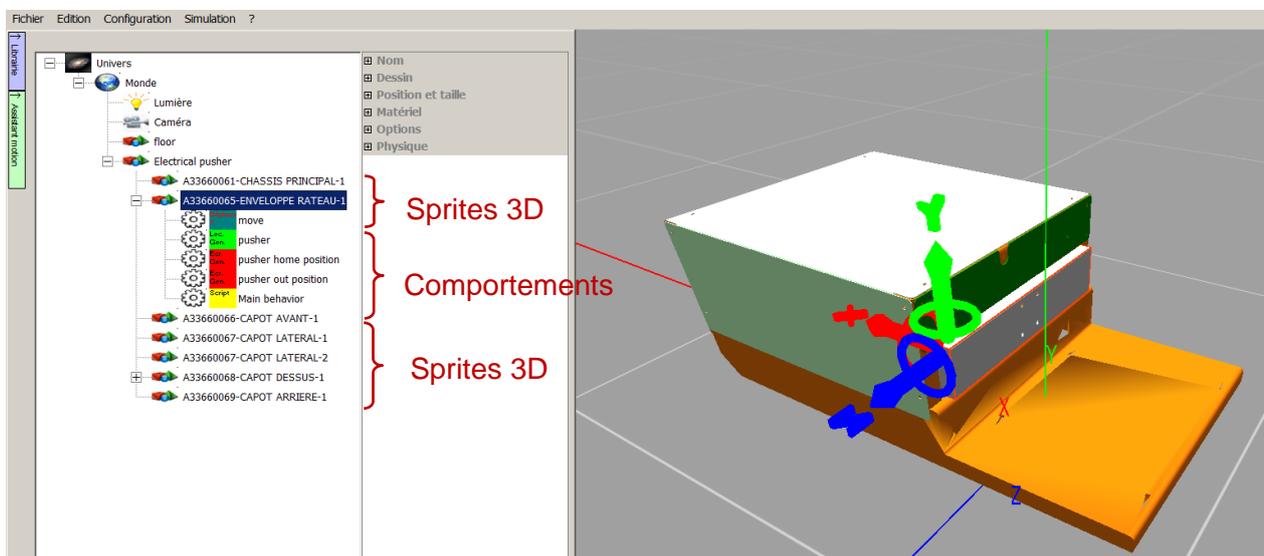
VIRTUAL UNIVERSE PRO permet de concevoir de nouvelles ressources 3D et systèmes virtuels intelligents, à partir des modèles 3D issus des logiciels de CAO (Conception Assistée par Ordinateur).

Ces ressources et systèmes 3D intelligents composent les émulateurs 3D VIRTUAL UNIVERSE PRO. Ils peuvent être sauvegardés indépendamment et venir compléter la bibliothèque des ressources VIRTUAL UNIVERSE PRO, afin d'être réutilisés pour des futurs projets de simulation.

Dans VIRTUAL UNIVERSE PRO, une ressource 3D intelligente est souvent composée de plusieurs sprites 3D (assemblage structuré de sprites) auxquels sont rattachés des comportements. Ces comportements représentent « l'intelligence » de la ressource en simulation.

Exemple

Le poussoir électrique (Electrical Pusher), présent dans la bibliothèque de démo (Demo Library), est une ressource 3D intelligente. Elle comporte un ensemble de sprites structuré ainsi que des comportements permettant de piloter la sortie/rentrée du tiroir (vérin).



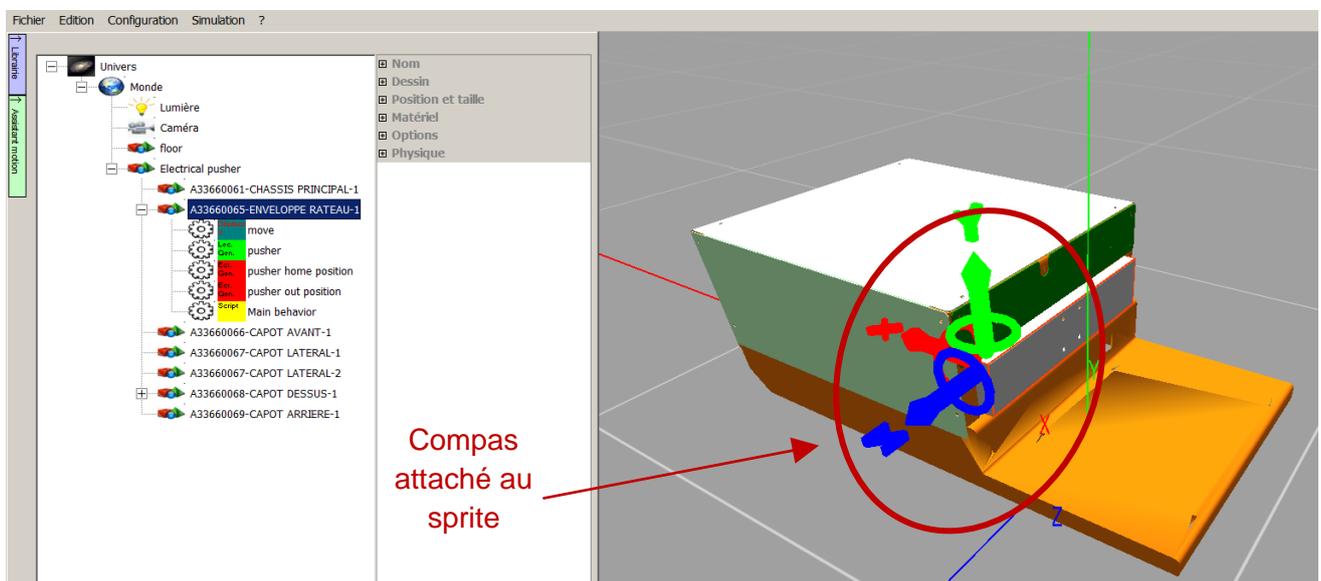
Modifier la position et les dimensions d'un sprite 3D

Dans VIRTUAL UNIVERSE PRO, la position d'un sprite dans le monde 3D est définie en coordonnées relatives, par rapport au sprite parent.

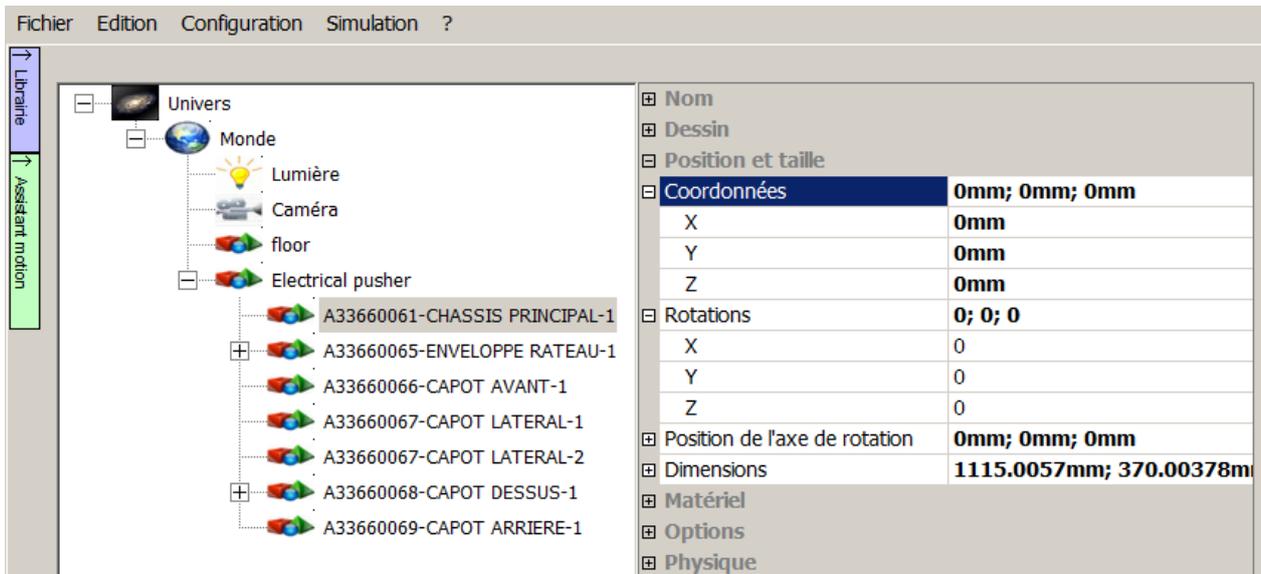
Ainsi, lorsque les données 3D sont importées depuis les logiciels de CAO, chaque sprite créé possède par défaut des coordonnées nulles (en position et en rotation) dans le repère local (position par rapport à son sprite parent).

Deux méthodes sont ensuite possibles pour déplacer un sprite :

- Il peut être déplacé directement via les flèches (translation) ou les cercles (rotation) du compas (X,Y,Z) attaché au sprite.

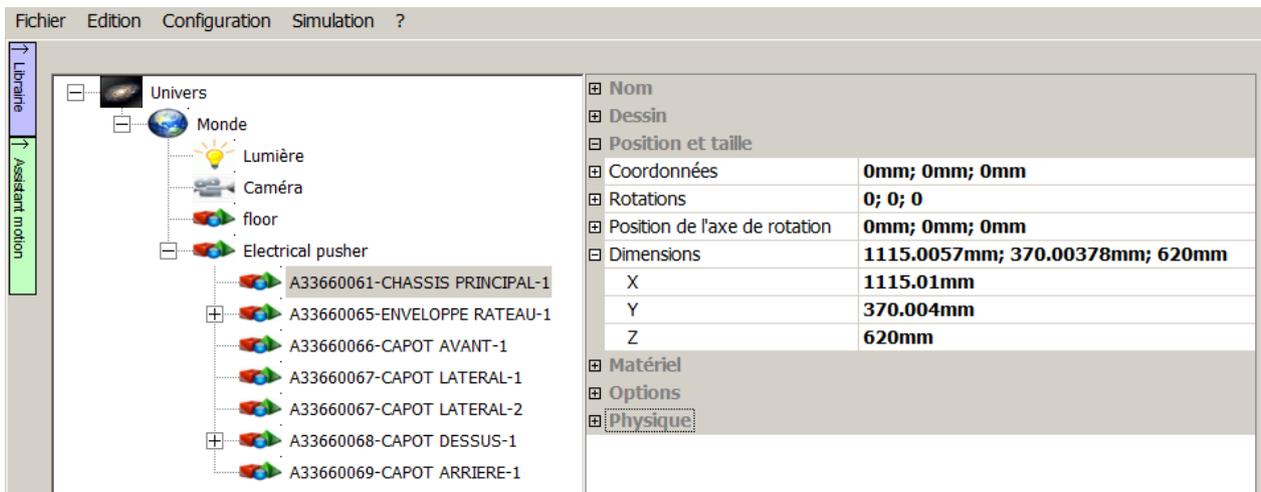


- Il peut aussi être déplacé de manière précise via le réglage des coordonnées locales « Position et taille » : Les unités affichées (mètres ou millimètres) sont celles choisies dans les propriétés du Monde.



- Pour modifier les dimensions d'un sprite, il suffit de modifier les paramètres relatifs aux dimensions du sprite dans l'onglet « Position et taille » des options du sprite.

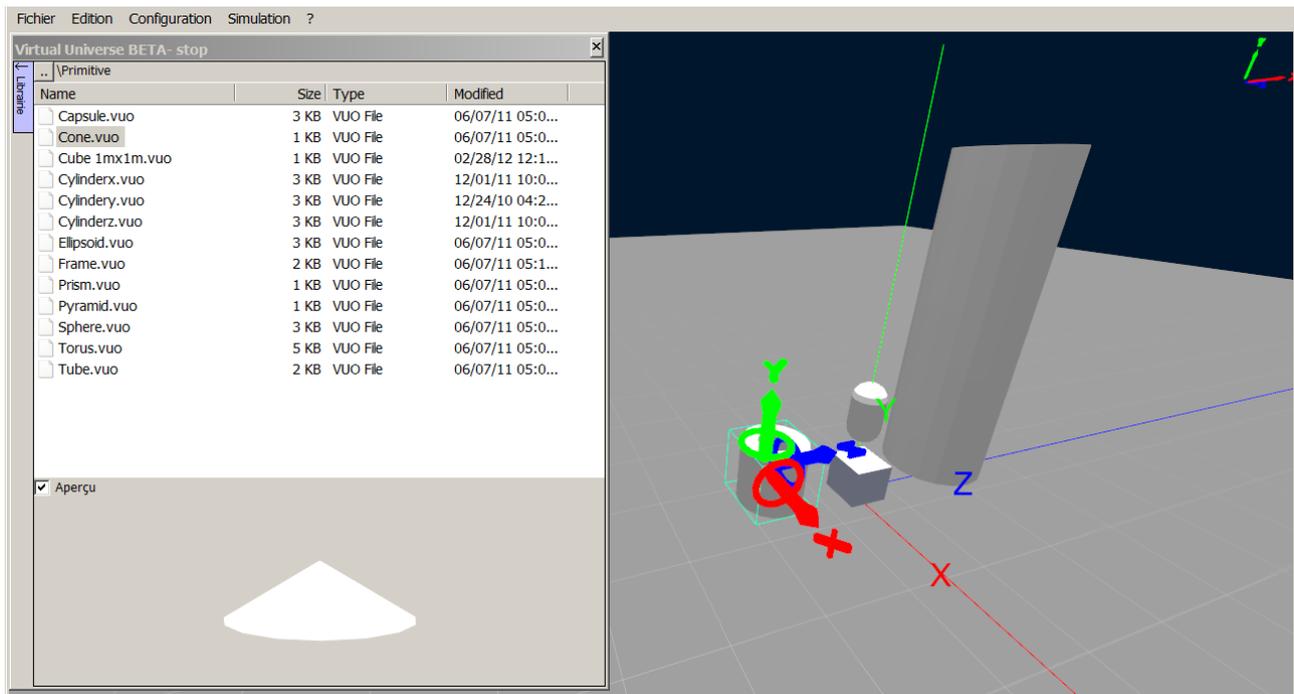
Les unités affichées (mètres ou millimètres) sont celles choisies dans les propriétés du Monde.



Ajouter des formes 3D primitives

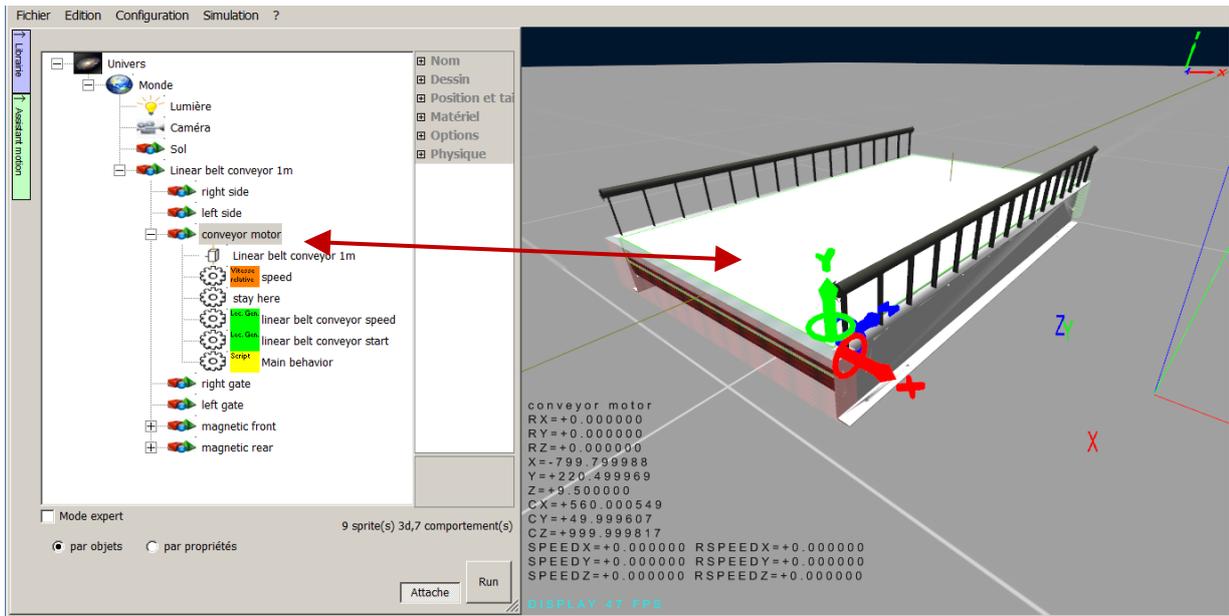
VIRTUAL UNIVERSE PRO fournit un ensemble de formes 3D primitives (cubes, cylindres,..), disponibles en bibliothèque.

Ces formes primitives permettent de représenter des éléments 3D manquants dans les modèles 3D CAO d'origine (comme le faisceau d'un capteur, ou la partie mobile d'un tapis roulant,..).



Exemple

Pour construire les convoyeurs disponibles dans la bibliothèque de démonstration (Demo library), une forme primitive 3D (parallélépipède) a été utilisée pour modéliser la partie mobile du convoyeur (sprite « conveyor motor »). Cette forme primitive a d'abord été insérée dans la ressource Convoyeur, retaillée à la bonne dimension et positionnée au bon endroit, et a enfin été cachée.

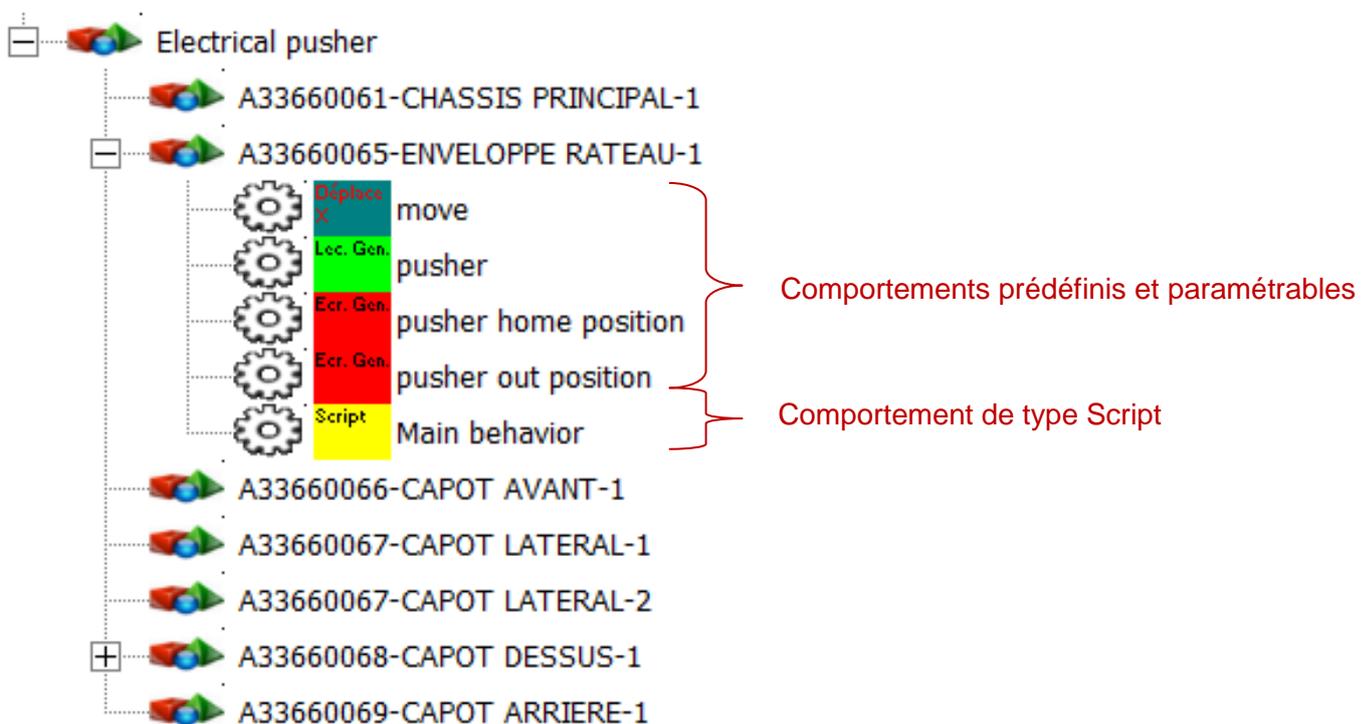


Ajouter des comportements aux sprites 3D

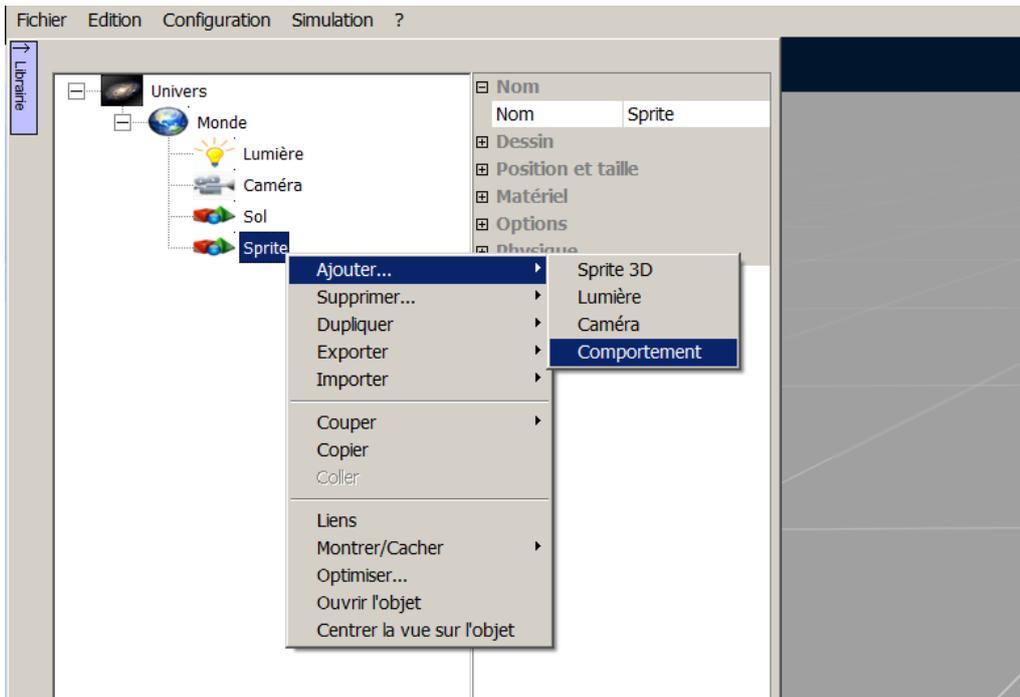
Les comportements représentent l'intelligence apportée aux sprites en cours de simulation.

L'ajout de comportements aux sprites permet de construire des ressources 3D intelligentes, capables de se déplacer dans le monde 3D, d'interagir avec d'autres ressources 3D ou de communiquer entre elles ou avec un logiciel/contrôleur extérieur. Les comportements permettent par exemple de modéliser le fonctionnement des actionneurs et récepteurs de la partie opérative d'un système automatisé.

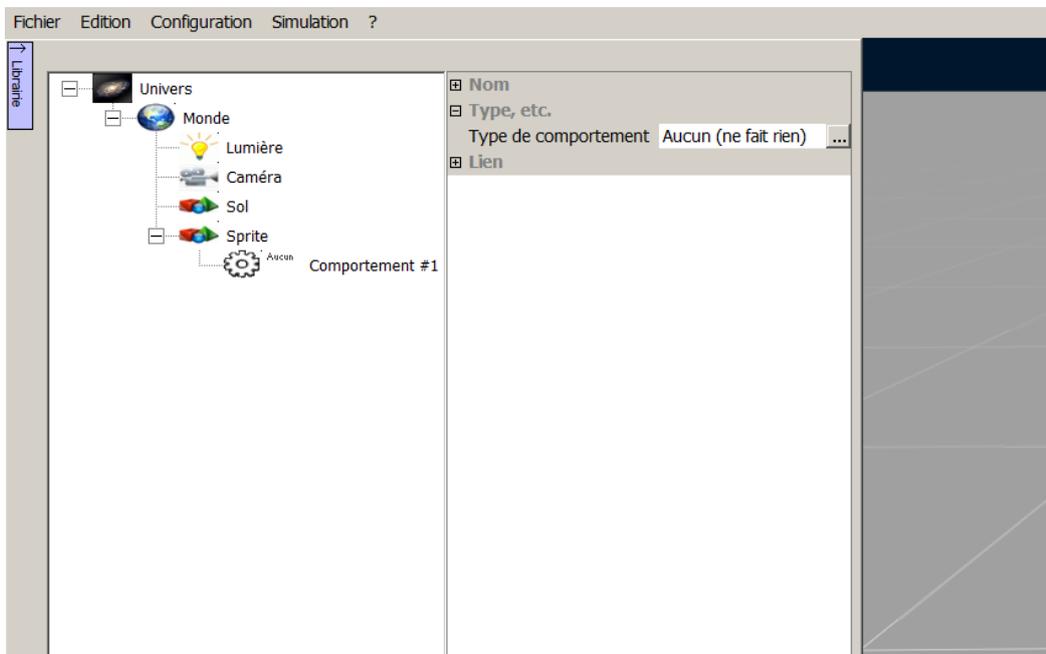
Les comportements sont, soit des comportements prédéfinis et paramétrables, disponibles dans VIRTUAL UNIVERSE PRO, soit des comportements plus élaborés de type Script créés par l'utilisateur avec l'éditeur de scripts intégré à VIRTUAL UNIVERSE PRO. Bien souvent, un script est utilisé pour modéliser le contrôleur interne principal de la ressource (logique comportementale).



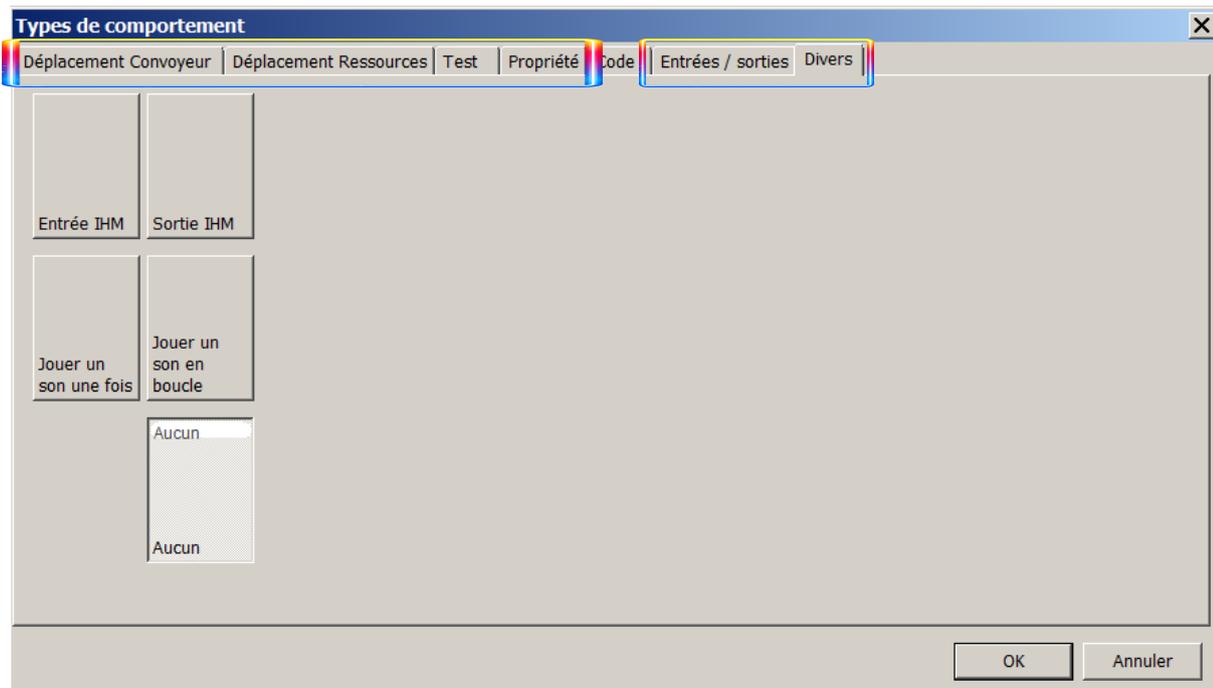
- L'ajout d'un comportement à un sprite s'obtient en cliquant droit sur le sprite sélectionné



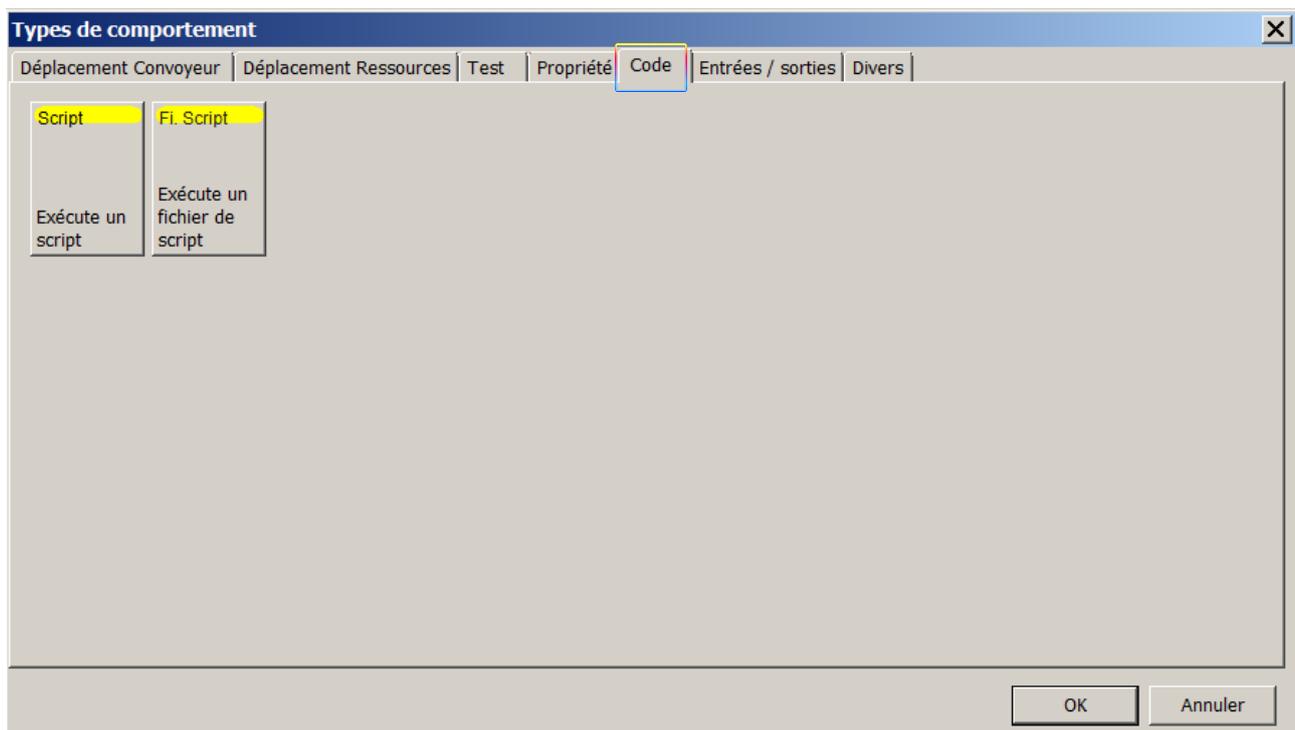
- Dans les propriétés du comportement, il est possible de choisir le type de comportement :



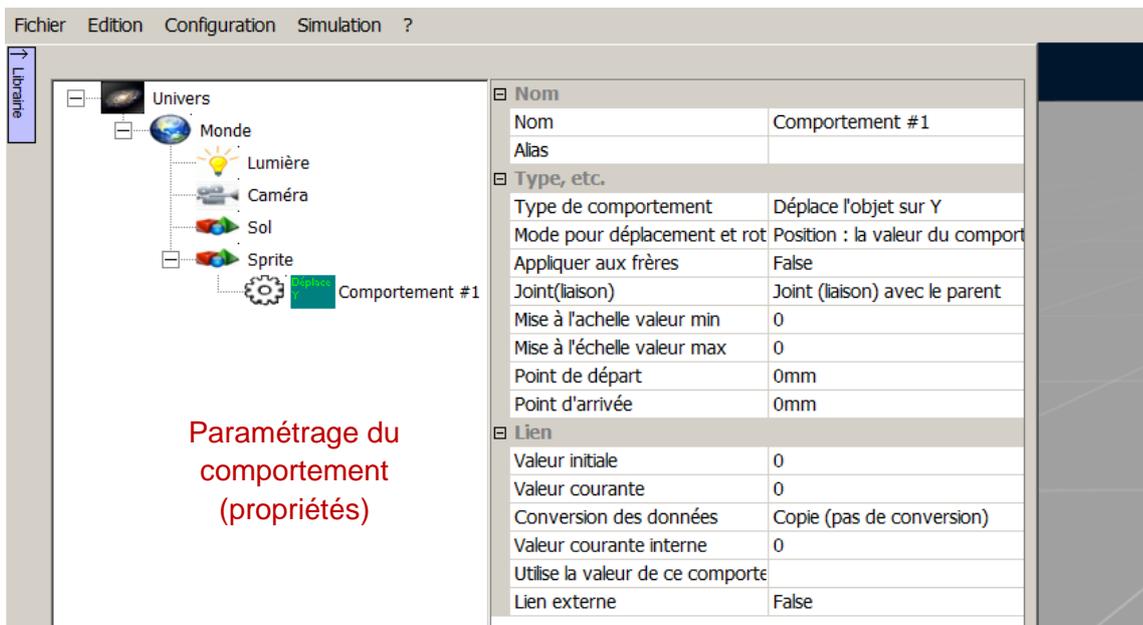
- Les comportements sont organisés en 7 catégories présentées dans la fenêtre « Types de comportement » :
 - les 6 catégories entourées regroupent des comportements prédéfinis de VIRTUAL UNIVERSE PRO



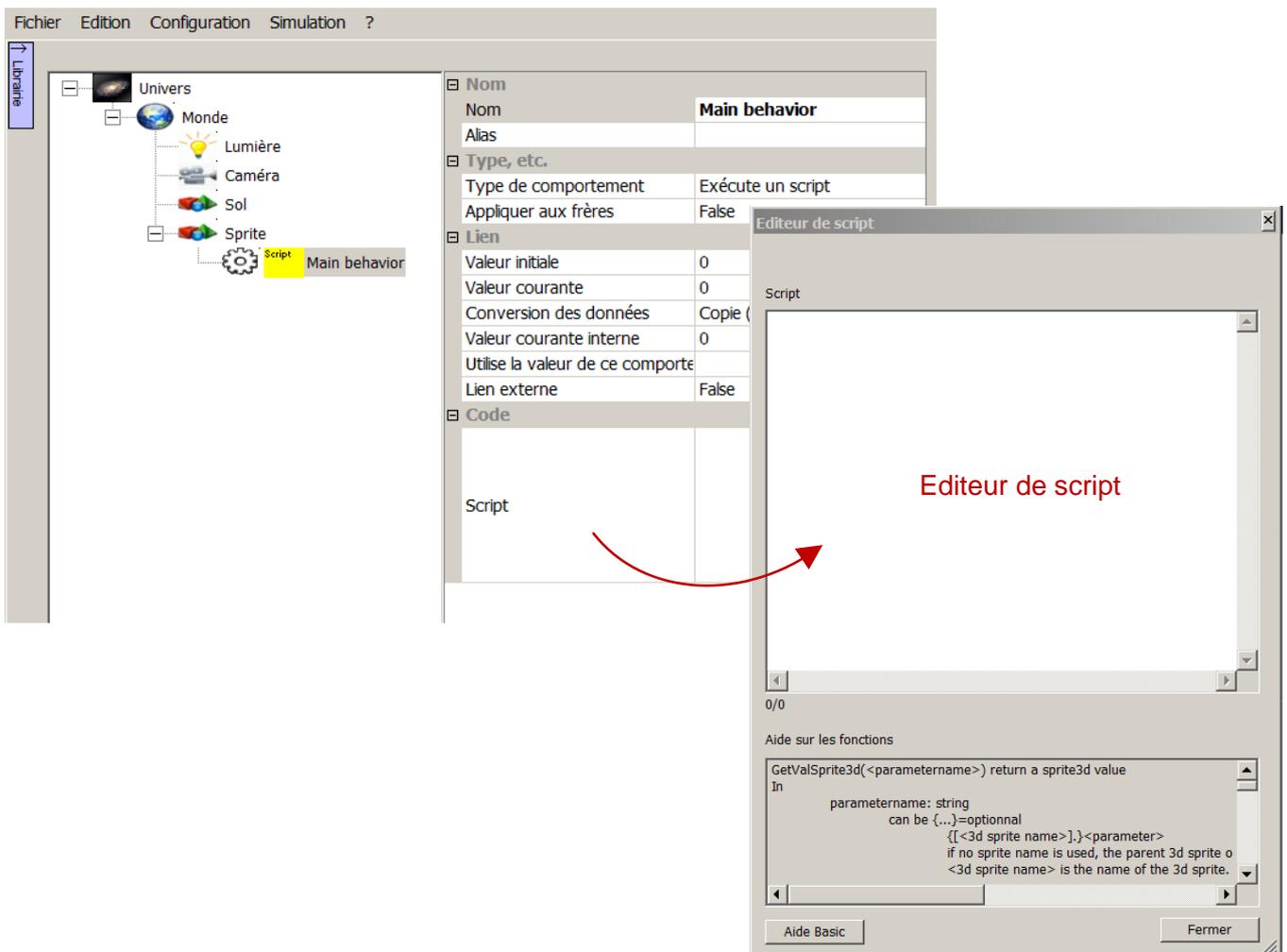
- la catégorie **Code** donne accès aux comportements de type Script



- Une fois le type de comportement choisi, il est possible de paramétrer le comportement en modifiant ses propriétés



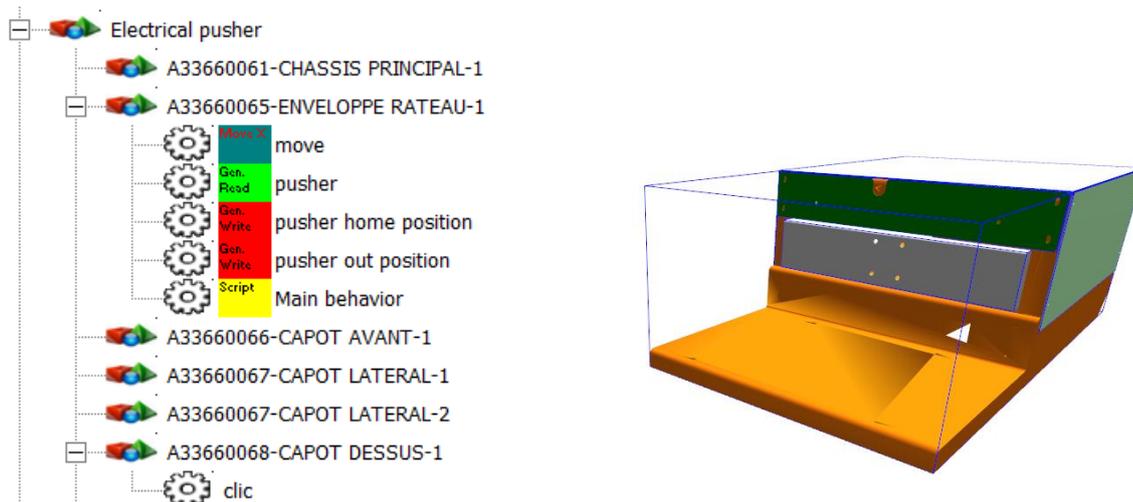
- Dans le cas du comportement Script, l'éditeur de script est accessible dans l'onglet Code



Pour plus d'information sur les types et les propriétés des comportements, voir [Propriétés des Comportements](#).

Exemple 1

Voir l'exemple de la ressource « Electrical Pusher » fournie dans la bibliothèque de démo VIRTUAL UNIVERSE PRO



Exemple 2

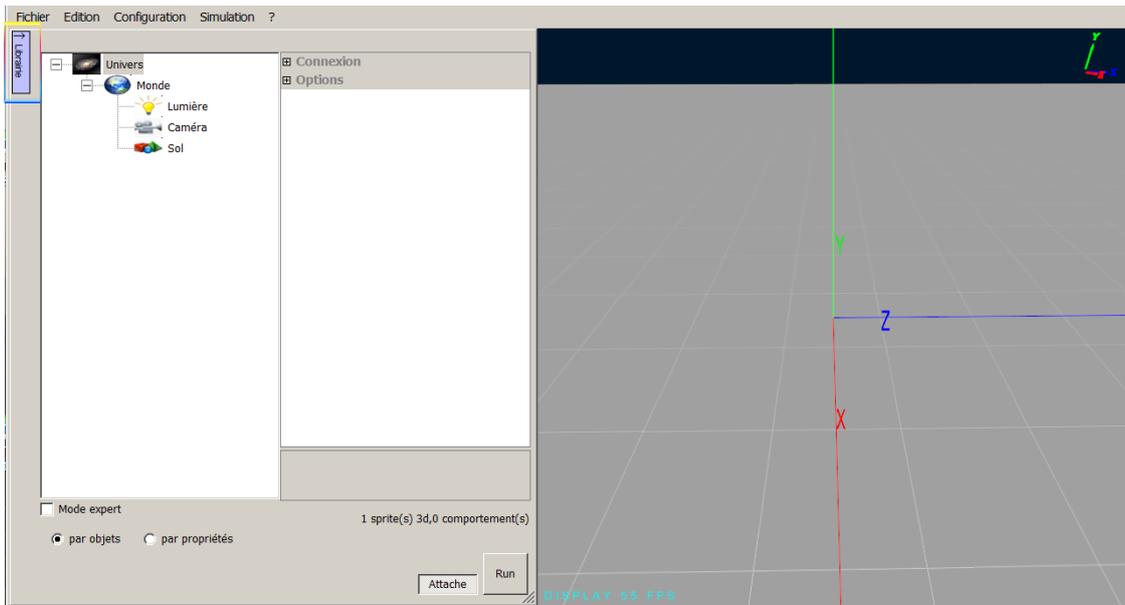
Voir l'exemple de la ressource « Detection sensor » fournie dans la bibliothèque de démo VIRTUAL UNIVERSE PRO



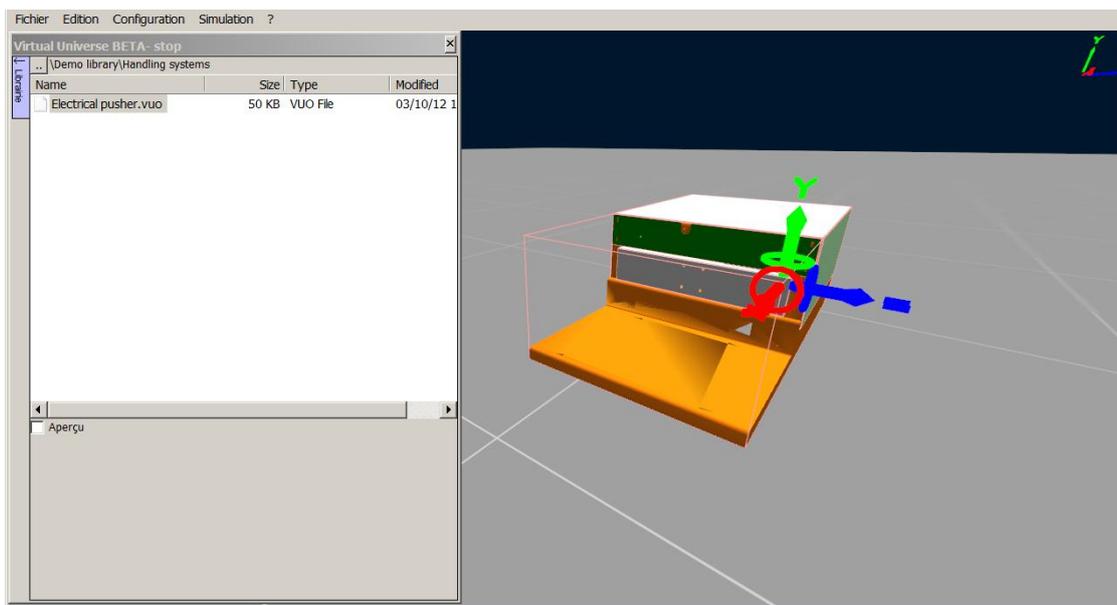
Exercice

L'objectif de ce petit exercice vise à ajouter un nouveau comportement prédéfini au niveau d'une ressource 3D de la bibliothèque (Electrical Pusher) et à modifier le script principal pour prendre en compte ce nouveau comportement.

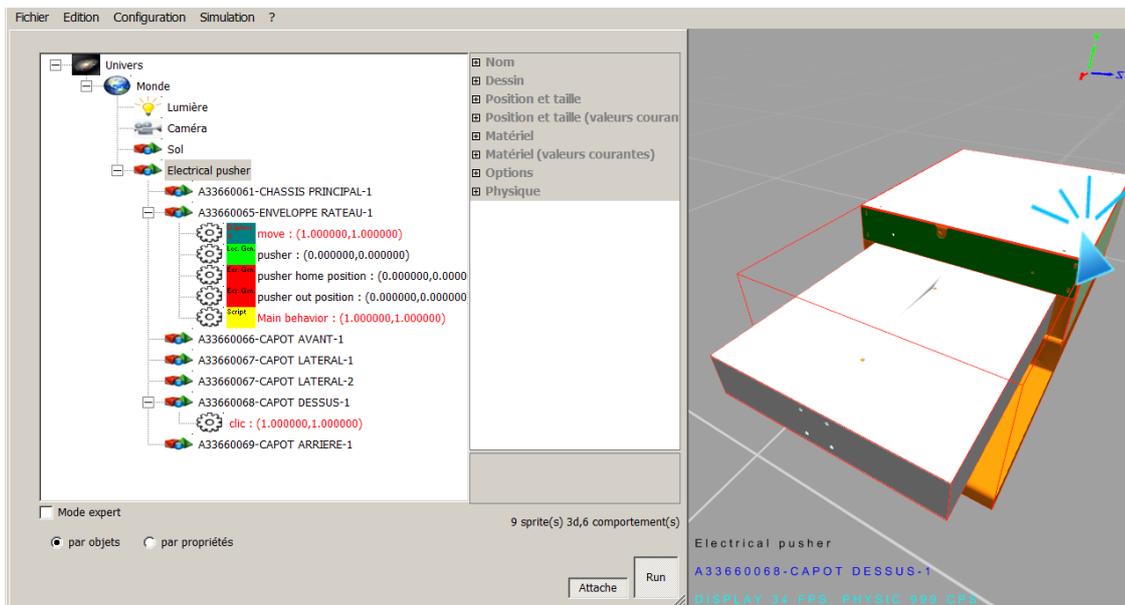
1. Ouvrir un projet vide et accéder à la bibliothèque des ressources de démonstration.



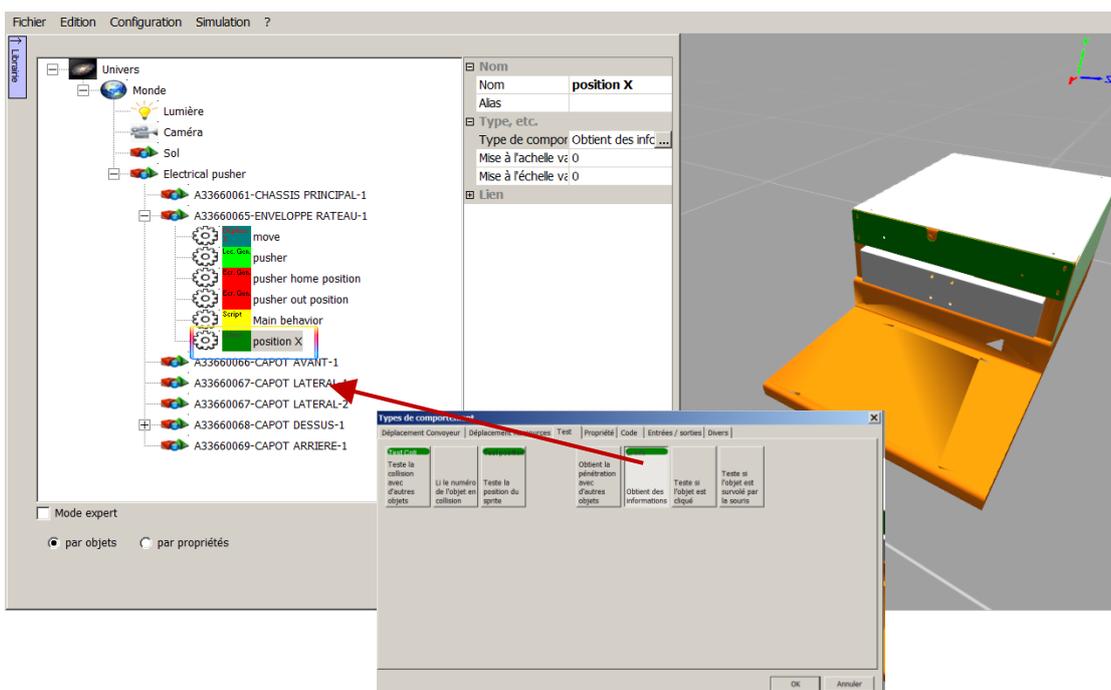
2. Depuis la bibliothèque, insérer la ressource « Electrical Pusher » dans le projet, en glissant la ressource avec la souris.



3. Revenir dans la fenêtre de configuration et lancer la simulation. En cours de simulation, cliquez sur le capot supérieur avec la souris pour faire sortir/rentrer le tiroir. Observez l'évolution des comportements.



4. Hors simulation, au niveau du sprite A33660065-ENVELOPPE, ajoutez un comportement de type « Obtient des informations » et renommez le « position X ».



5. Dans les propriétés du comportement, sélectionnez « Position X » dans le type d'information retournée. Ce comportement retourne donc la position suivant l'axe X du sprite A33660065-ENVELOPPE (dans le repère local au sprite parent Electrical pusher).

Nom	
Nom	position X
Alias	
Type, etc.	
Type de compor	Obtient des inform
Mise à l'échelle v:	0
Mise à l'échelle v:	0
Lien	
Valeur initiale	0
Valeur courante	0
Conversion des (Copie (pas de conv
Valeur courante :	0
Mode d'écriture	Normal
Utilise la valeur d	
Lien externe	False
Sélectionne l'info	Position X

6. Identifiez le comportement « Main behavior » et ouvrez l'éditeur de script associé.

The screenshot shows the software interface with a 3D model of a mechanical assembly on the right and a script editor window on the left. The script editor window is titled 'Editeur de script' and contains the following code:

```

myloop:
curpos=GetValSprite3d("POSX")
if getbehavior("..\pusher.currentvalue")=true or getbehavior("..\A3366006
setbehavior("..\move.values",1)
else
setbehavior("..\move.values",0)
endif
if curpos>=5 then
setbehavior("..\pusher home position.values",1)
else
setbehavior("..\pusher home position.values",0)
endif
if curpos<=-1.2 then
setbehavior("..\pusher out position.values",1)
else
setbehavior("..\pusher out position.values",0)
endif
goto myloop

```

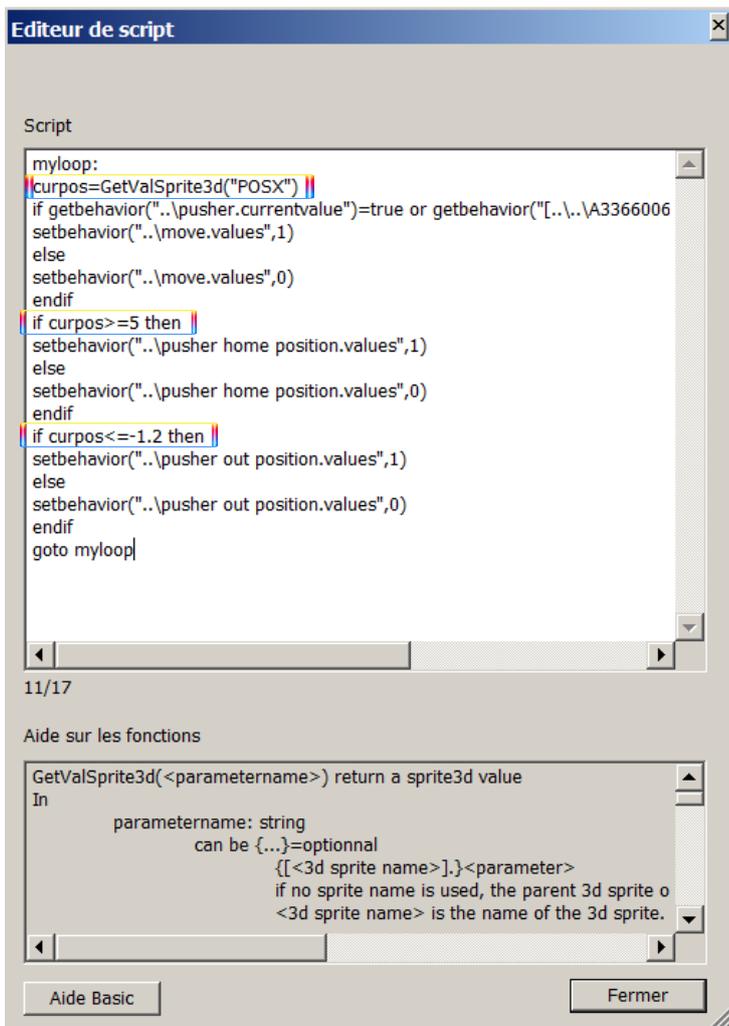
A red arrow points from the 'Main behavior' entry in the hierarchy to the script editor. Below the script editor is a help window titled 'Aide sur les fonctions' with the following text:

```

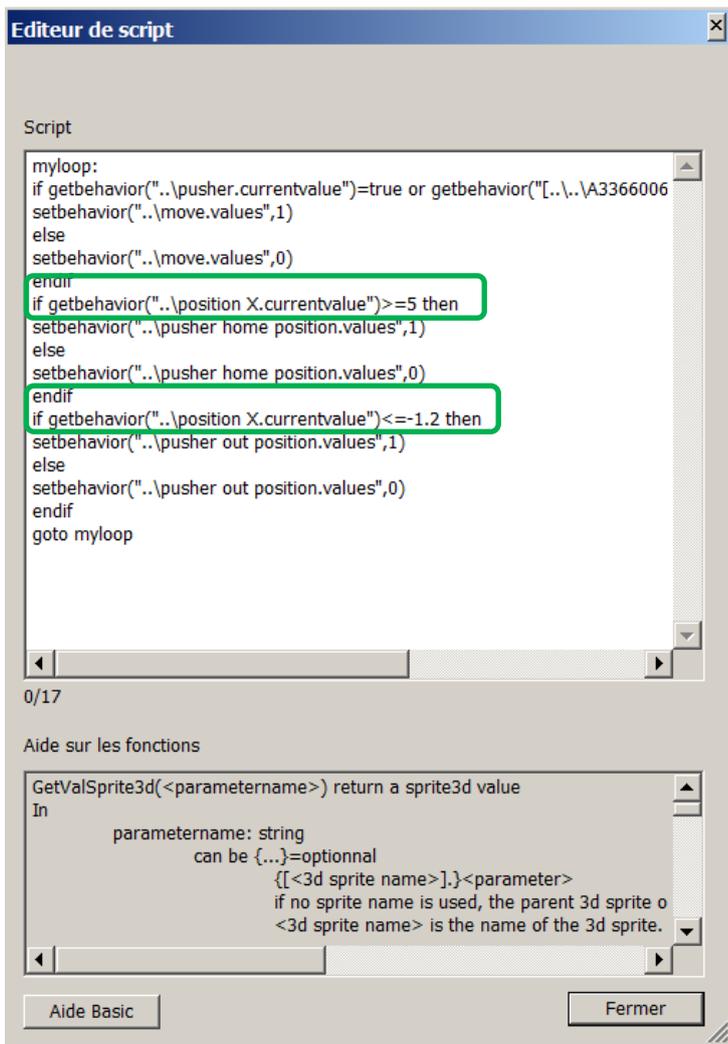
GetValSprite3d(<parametername>) return a sprite3d value
In
  parametername: string
  can be (...)=optional
  {[<3d sprite name>],}<parameter>
  if no sprite name is used, the parent 3d sprite o
  <3d sprite name> is the name of the 3d sprite.

```

7. La modification dans ce script consiste à utiliser le nouveau comportement créé « Position X » à la place de la fonction *curpos=GetValSprite3d("POSX")*.

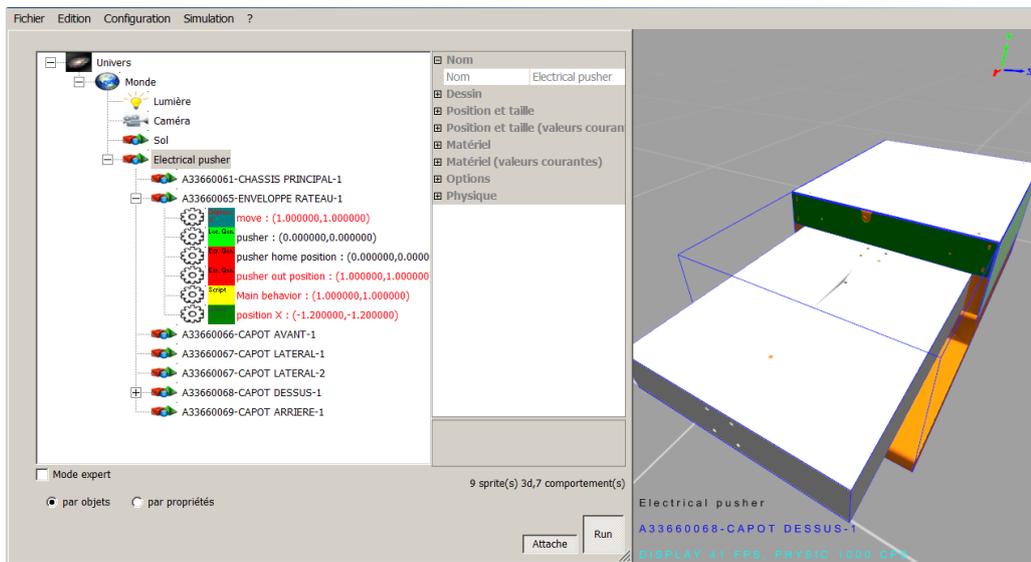


Ancien script



Nouveau script

8. Relancez la simulation avec ce nouveau script et constatez que le comportement du poussoir n'a pas changé.



Définir des profils de mouvement avec le Motion Assistant

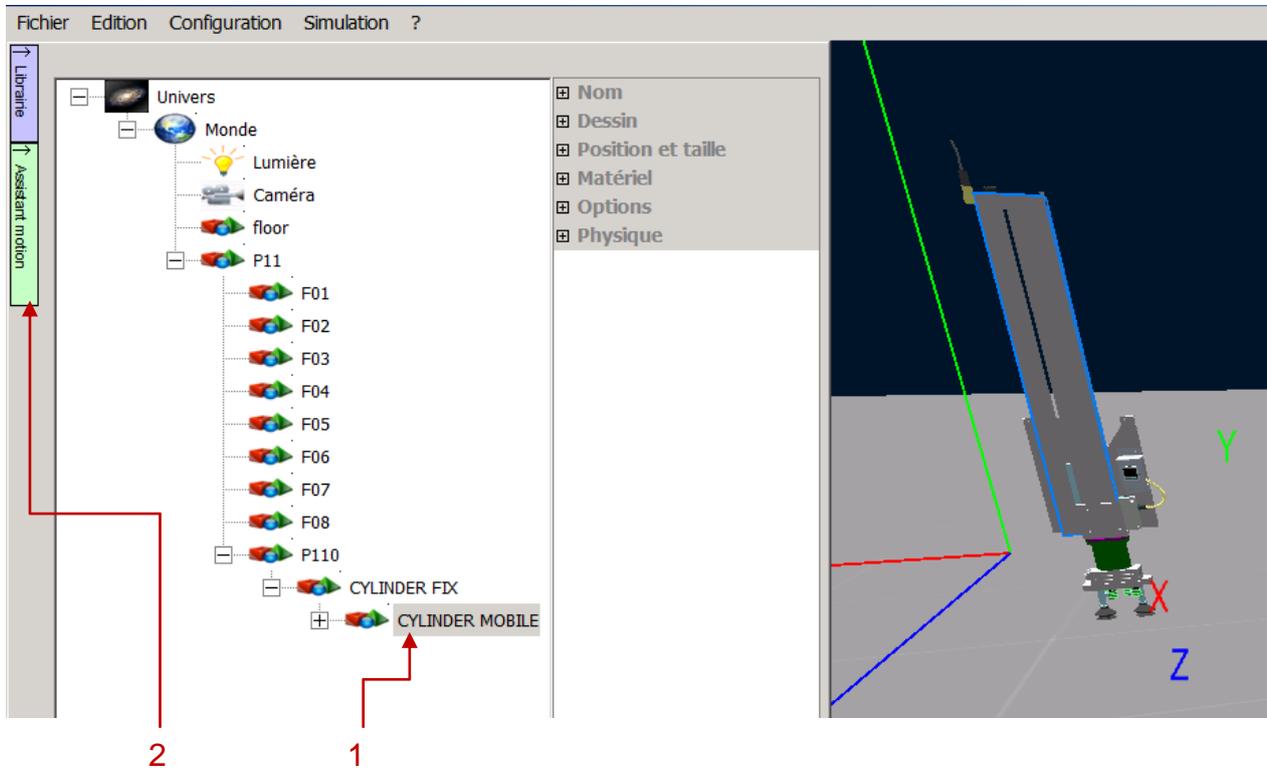
Le Motion Assistant permet de définir simplement les profils de mouvement d'une ressource ou d'un système mécanique 3D, en créant et en paramétrant automatiquement les comportements nécessaires à la simulation de ces profils de mouvement.

Une ressource mécanique peut être par exemple un vérin ou un axe piloté par un ensemble {moteur + variateur de vitesse}. Le Motion Assistant permet, selon le cas, de piloter une ressource de différentes façons:

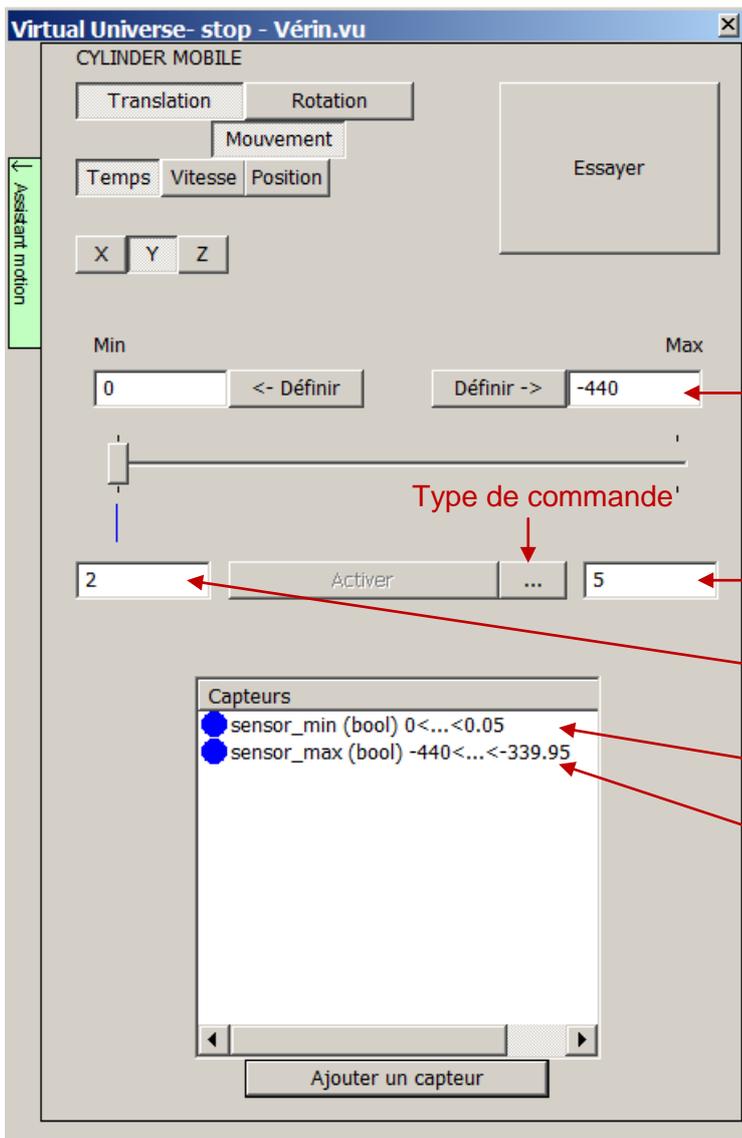
- en temps
- en vitesse
- en position

Pilotage en fonction du temps

C'est typiquement le profil de mouvement utilisé dans le cas d'un vérin piloté en temps par un contrôleur externe.



- Tout d'abord, sélectionnez la partie mobile de la ressource mécanique que vous souhaitez mettre en mouvement dans l'arborescence du projet (il s'agit de la tige du vérin dans l'exemple ci-dessus).
- Cliquez ensuite sur l'onglet vert « Assistant motion » situé à gauche de l'arborescence.
- La fenêtre du Motion Assistant apparaît alors à l'écran.



Définition des butées Min et Max

Type de commande

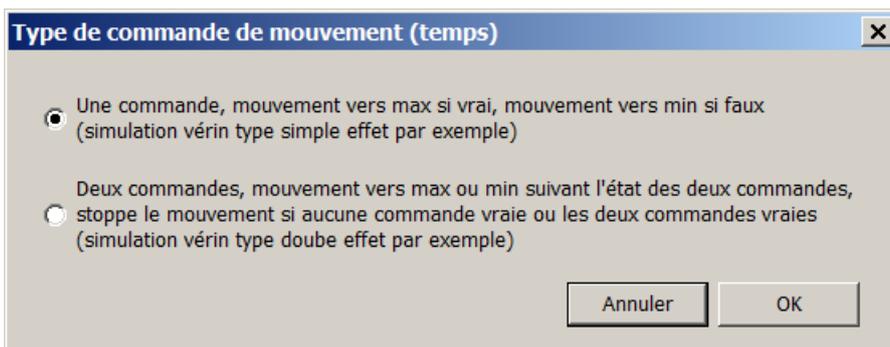
Temps « retour » (de Max à Min)

Temps « aller » (de Min à Max)

Capteur booléen position Min

Capteur booléen position Max

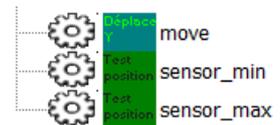
- Définissez les butées Min et Max, soit manuellement, soit directement en déplaçant la ressource dans la fenêtre de rendu (ces butées seront atteintes par la ressource).
- Définissez les temps « aller » et temps « retour » de la ressource mécanique (tige du vérin).
- Sélectionnez le type de commande.



- Il est également possible d'ajouter des capteurs booléens (ou capteurs Tout ou Rien), renvoyant 1 si la ressource mécanique se situe dans la plage de détection du capteur et 0 sinon. L'exemple suivant illustre la création d'un capteur booléen (sensor_max) situé autour de la butée maximale.

- Les comportements générés sont les suivants :

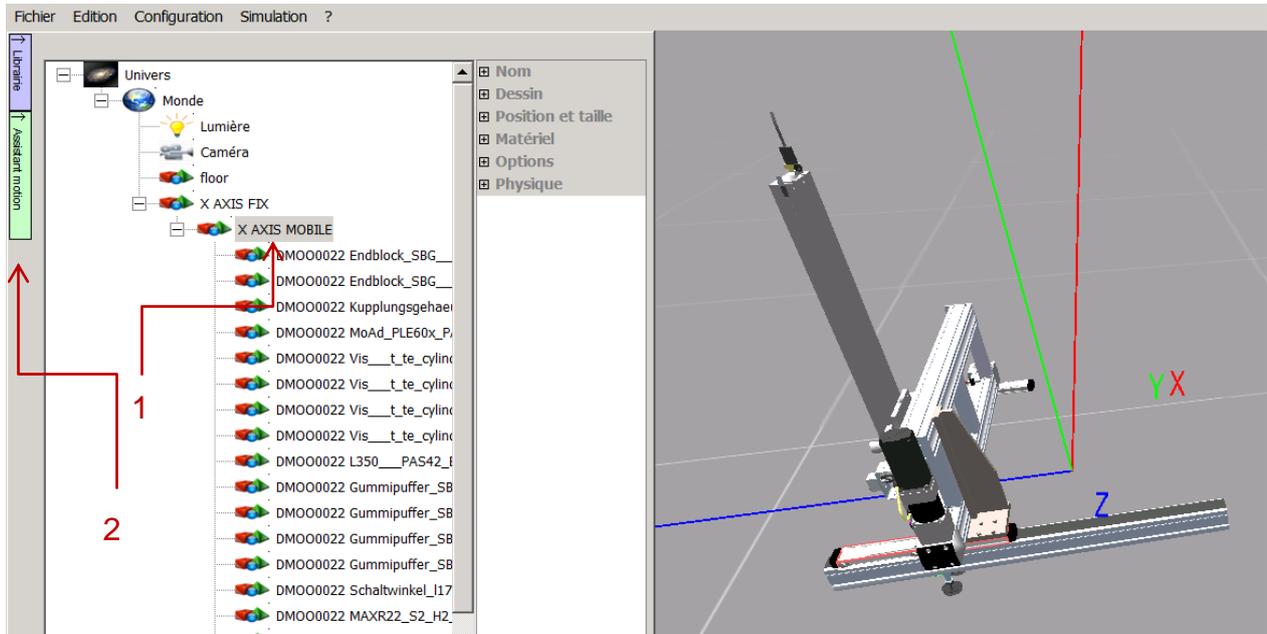
Comportement « move », actionneur du vérin, entrée simulation
Capteur booléen min, sortie simulation
Capteur booléen max, sortie simulation



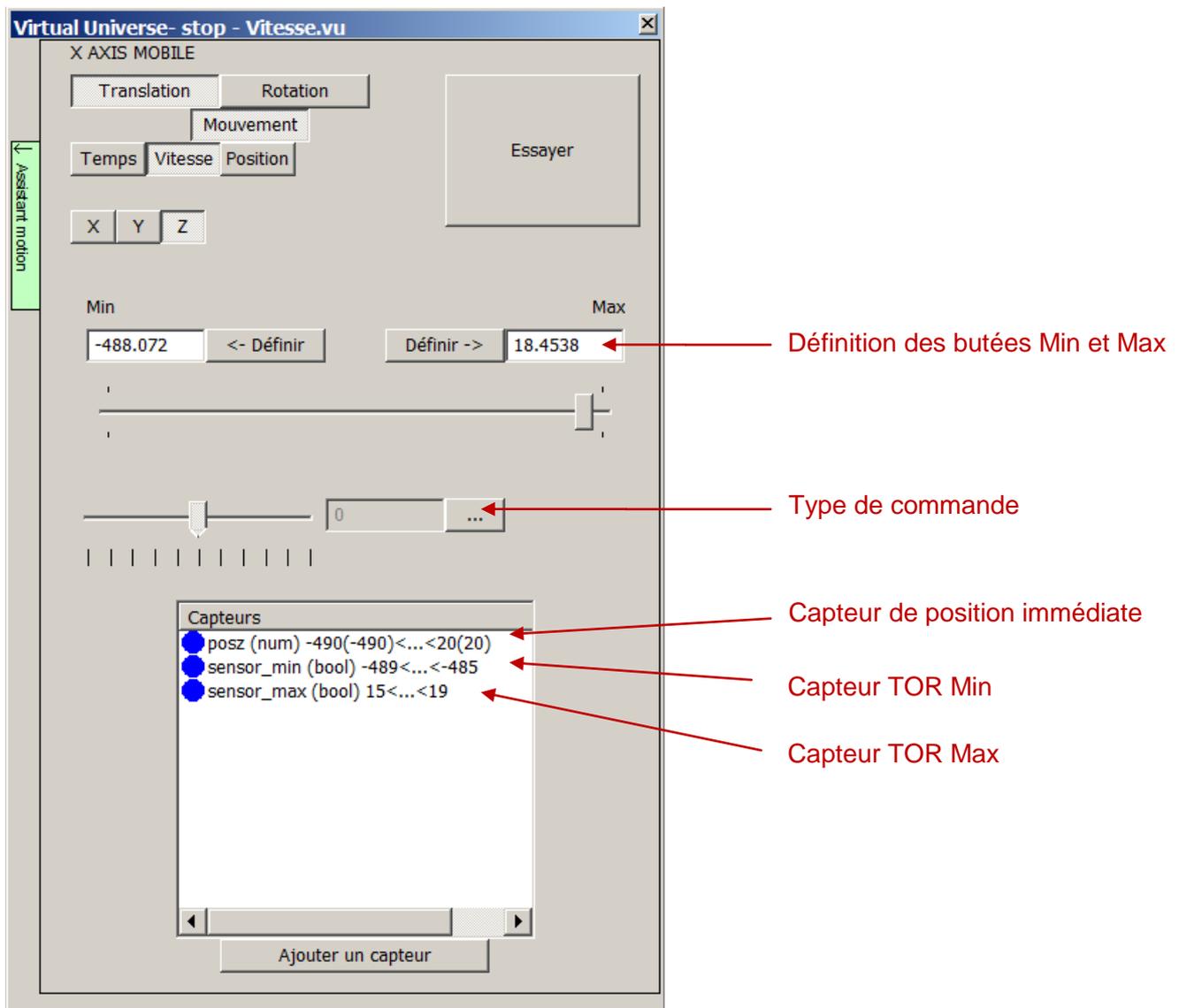
- Une valeur de 0 pour le comportement « move » entrainera un déplacement « retour » et une valeur de 1 un déplacement « aller ».
- Min et Max ont une valeur de 1 lorsque la ressource se situera dans la plage de détection du capteur, 0 sinon.

Pilotage en fonction de la vitesse

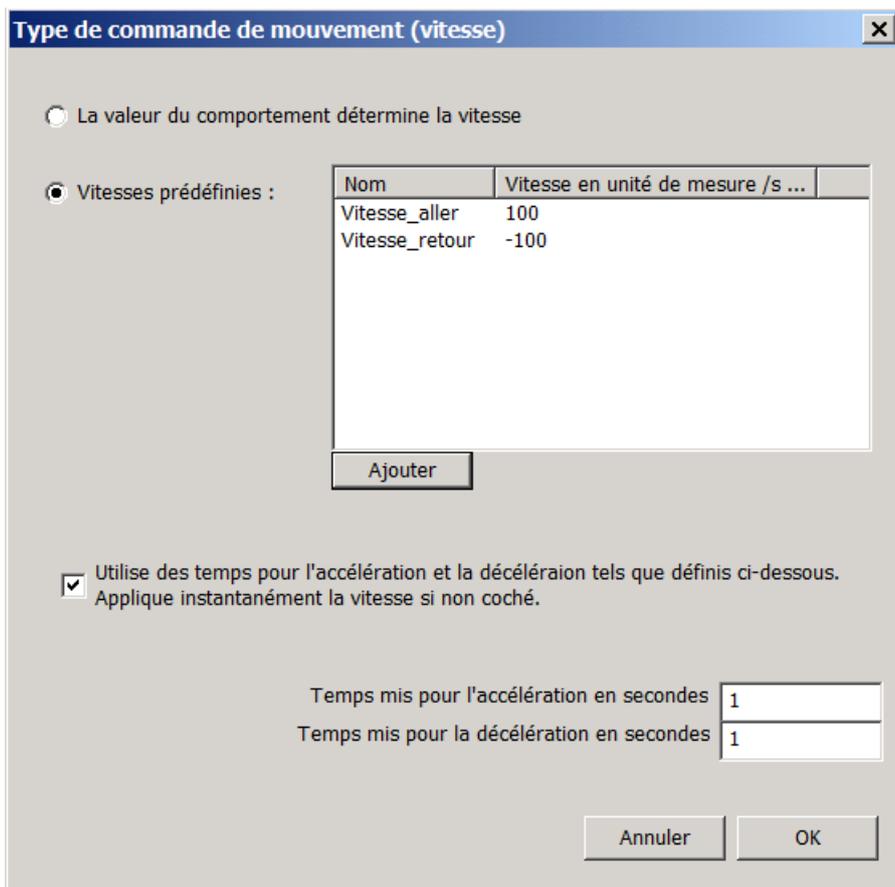
C'est typiquement le profil de mouvement utilisé dans le cas d'un axe, piloté par un ensemble {moteur + variateur de vitesse} recevant des consignes de vitesse d'un automate programmable, externe à la simulation.



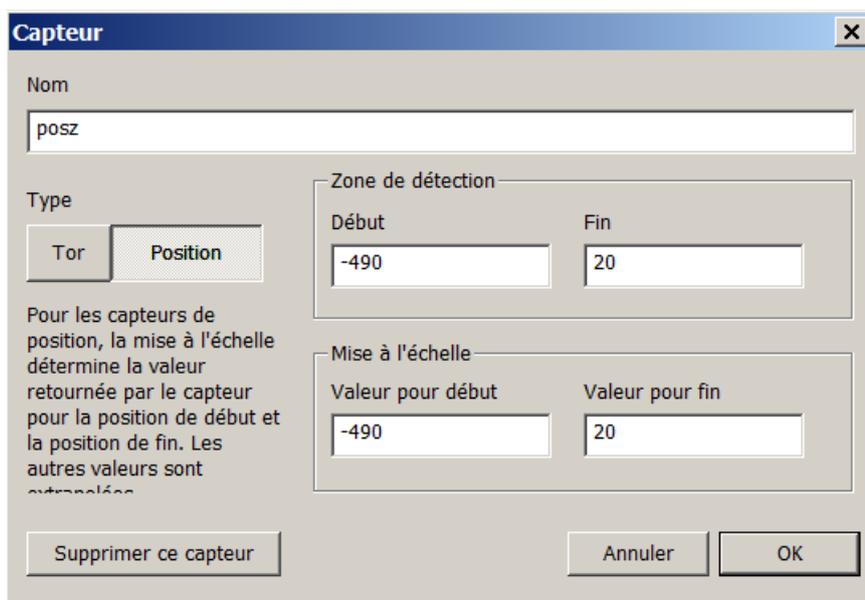
- Tout d'abord, sélectionnez la ressource mécanique que vous souhaitez mettre en mouvement dans l'arborescence du projet.
- Cliquez ensuite sur l'onglet vert « Assistant motion » situé à gauche de l'arborescence (l'image ci-après détaille ces premières étapes).



- Définissez les butées Min et Max, soit manuellement directement en déplaçant la ressource dans la fenêtre de rendu (ces butées ne seront pas atteintes par la ressource).
- Sélectionnez le type de commande en fonction de la commande que vous souhaitez appliquer (vitesse prédéfinies, temps d'accélération / décélération). L'image suivante illustre la création d'un mouvement possédant deux vitesses prédéfinies (100 et -100 mm/s) et d'un profil d'accélération/décélération (une seconde pour atteindre la vitesse appliquée).

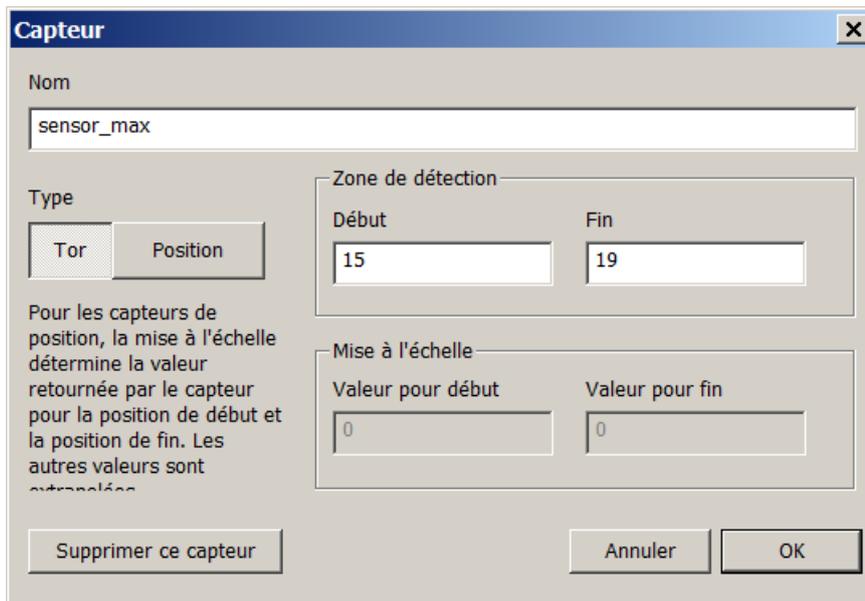


- L'ajout d'un capteur de position « immédiate » (renvoyant directement la position de la ressource selon l'axe du mouvement) permet de disposer d'un retour de position. L'image suivante illustre la création d'un tel capteur posz.

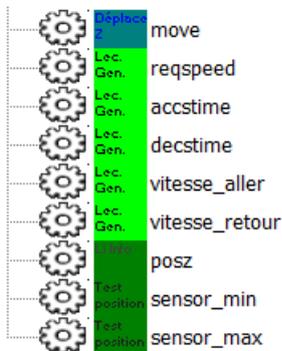


- Il est également possible d'ajouter des capteurs TOR (tout ou rien), renvoyant 1 si la ressource mécanique se situe dans la plage de détection du capteur. L'exemple

suisant illustre la création d'un capteur TOR (sensor_max) situé autour de la butée maximale qui peut, par exemple, faire office de capteur de sécurité.



- Les comportements générés sont les suivants :

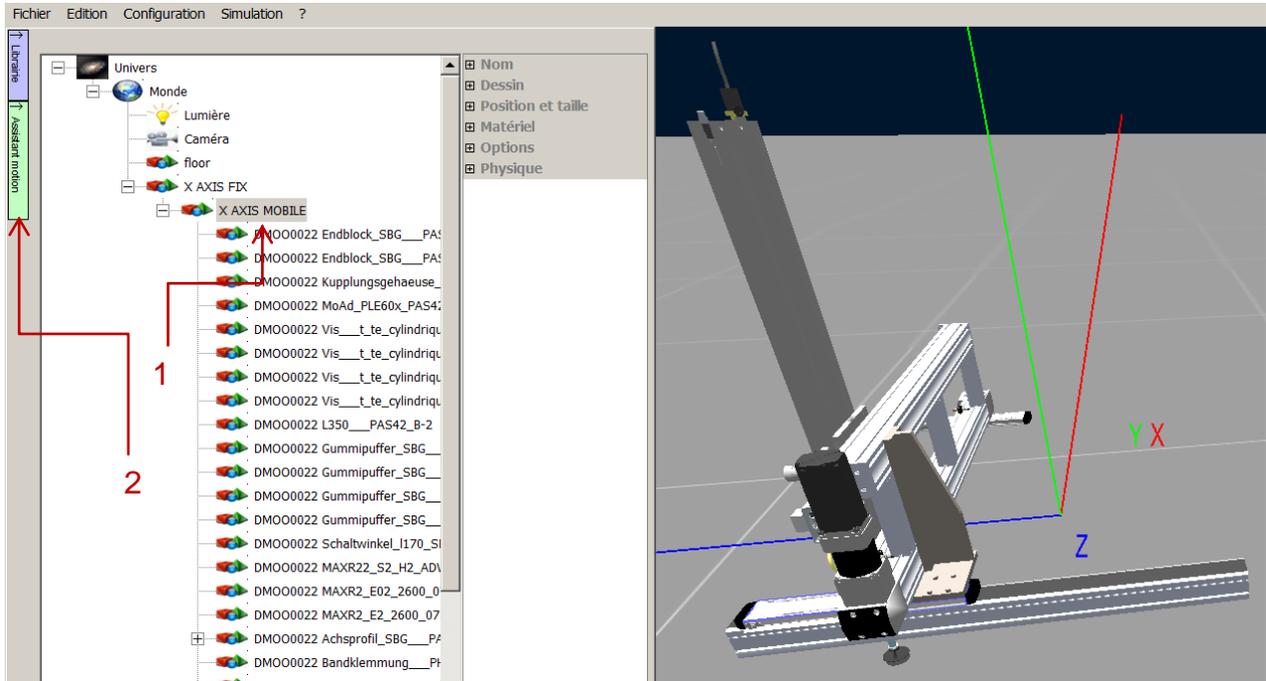


La valeur du comportement détermine la vitesse
Vitesse de consigne, entrée simulation
Temps d'accélération, entrée simulation
Temps de décélération, entrée simulation
Vitesse prédéfinie 1, entrée simulation
Vitesse prédéfinie 2, entrée simulation
Capteur de position « immédiate », sortie simulation
Capteur booléen min, sortie simulation
Capteur booléen max, sortie simulation

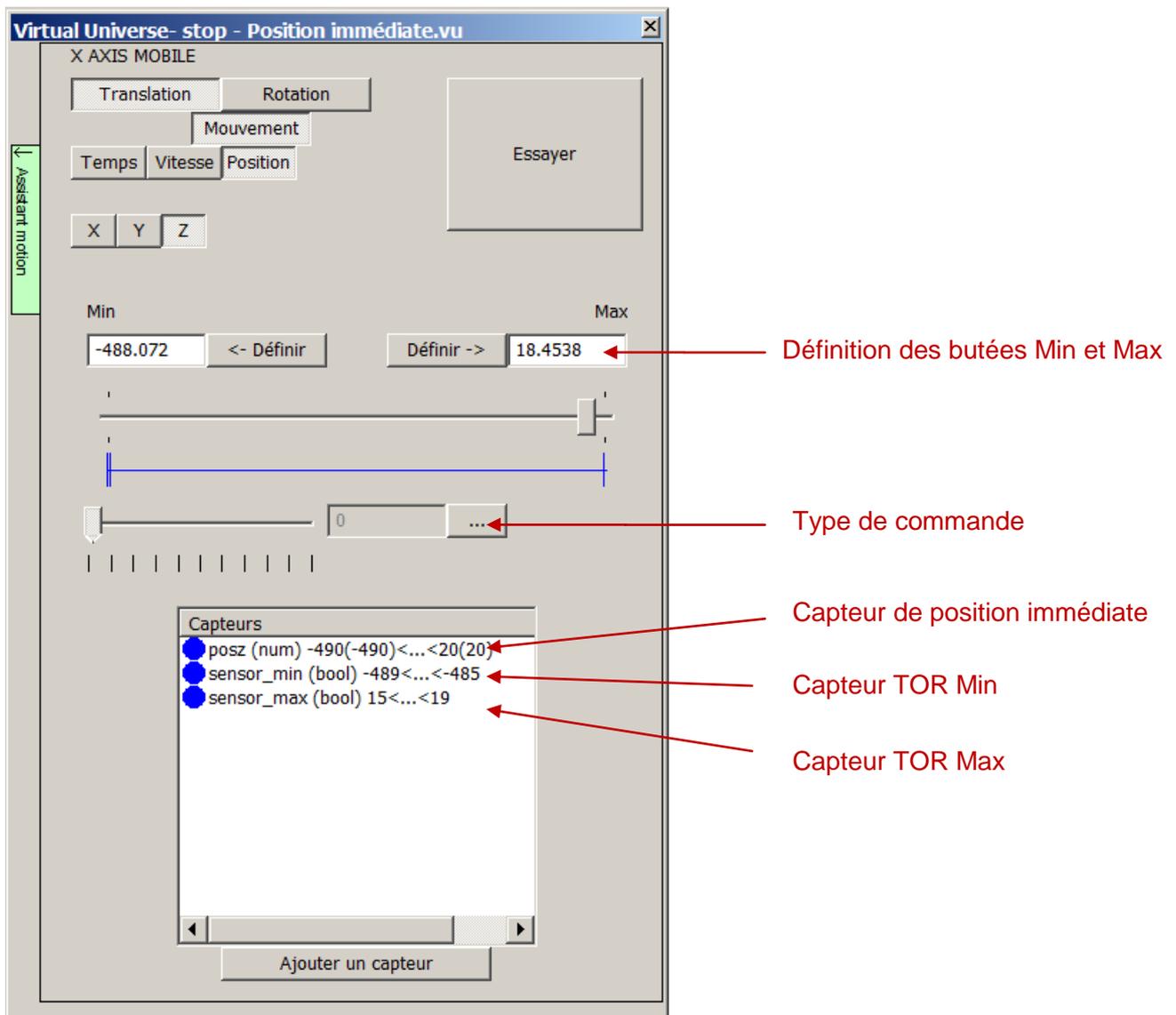
- Une valeur de 1 pour un comportement « Vitesse prédéfinie », force la vitesse de consigne à cette valeur.
- Reqspeed, accstime, decstime sont créés uniquement si l'utilisateur définit un profil d'accel/decel. Si aucun profil n'est défini, la partie commande écrira directement « move » qui sera alors une entrée de la simulation.

Pilotage en fonction de la position immédiate

C'est typiquement le profil de mouvement utilisé dans le cas d'un axe, piloté par un ensemble {moteur + variateur de vitesse} recevant des consignes de position d'un contrôleur d'axe, externe à la simulation.



- Tout d'abord, sélectionnez la ressource mécanique que vous souhaitez mettre en mouvement dans l'arborescence du projet.
- Cliquez ensuite sur l'onglet vert « Assistent motion » situé à gauche de l'arborescence (l'image ci-après détaille ces premières étapes).
- La fenêtre du Motion Assistant apparaît alors à l'écran.



- Définissez les butées Min et Max, soit manuellement directement en déplaçant la ressource dans la fenêtre de rendu (ces butées ne seront pas atteintes par la ressource).
- Sélectionnez le type de commande (position directe ou relative).

Type de commande de mouvement (position)

La valeur du comportement détermine la position
 La valeur du comportement comprise entre les deux valeurs (Début et Fin) sélectionnées ci-dessous détermine la position entre min et max

Début
 Fin

Utilise un profil accélération/décélération pour atteindre la position avec les paramètres définis ci-dessous. Atteint instantanément la position si non coché.

Vitesse maximum en unités de mesure /s (degrés /s pour les
 Temps mis pour atteindre la vitesse maximum en secondes
 Temps mis pour atteindre la vitesse nulle en seconde (décélération)

← Commande en position directe

← Commande en position relative

- L'ajout d'un capteur de position « immédiate » (renvoyant directement la position de la ressource selon l'axe du mouvement) permet de disposer d'un retour de position. L'image suivante illustre la création d'un tel capteur posz.

Capteur

Nom

Type

Pour les capteurs de position, la mise à l'échelle détermine la valeur retournée par le capteur pour la position de début et la position de fin. Les autres valeurs sont extrapolées.

Zone de détection
 Début Fin

Mise à l'échelle
 Valeur pour début Valeur pour fin

- Il est également possible d'ajouter des capteurs TOR (tout ou rien), renvoyant 1 si la ressource mécanique se situe dans la plage de détection du capteur. L'exemple suivant illustre la création d'un capteur TOR (sensor_max) situé autour de la butée maximale qui peut, par exemple, faire office de capteur de sécurité.

Capteur [X]

Nom

Type
 Tor Position

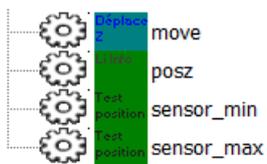
Zone de détection
 Début Fin

Mise à l'échelle
 Valeur pour début Valeur pour fin

Supprimer ce capteur Annuler OK

Pour les capteurs de position, la mise à l'échelle détermine la valeur retournée par le capteur pour la position de début et la position de fin. Les autres valeurs sont extrapolées.

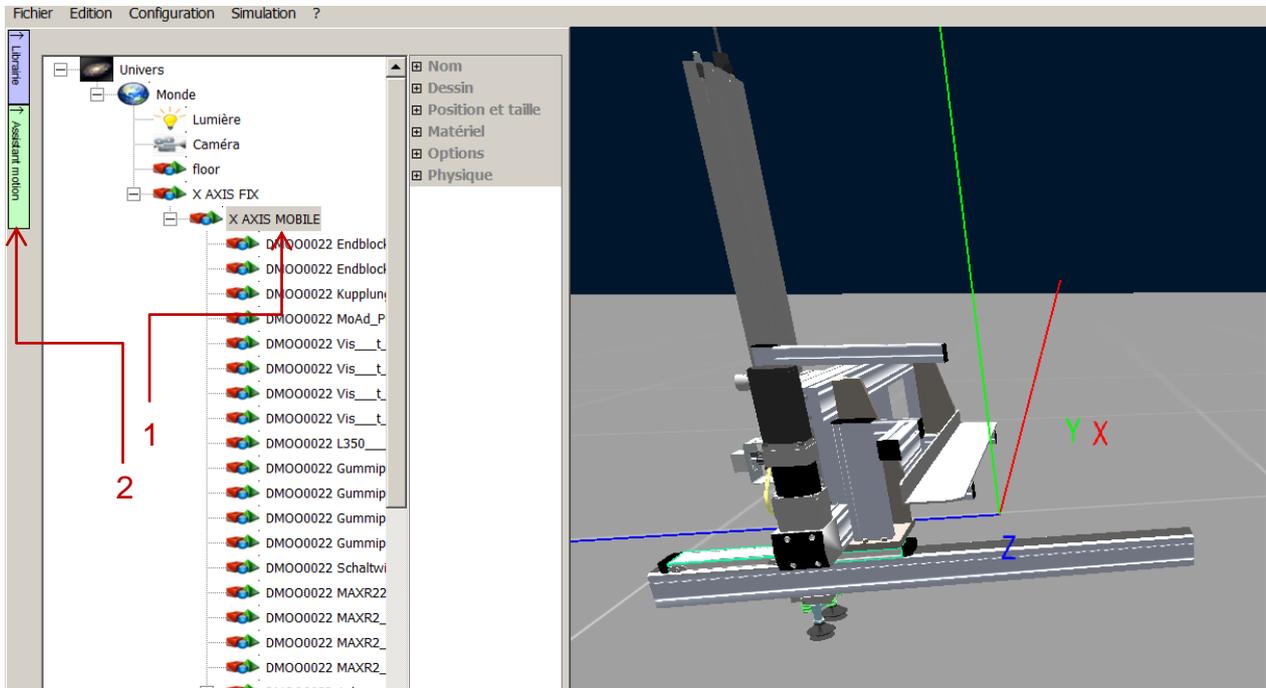
- Les comportements générés sont les suivants :



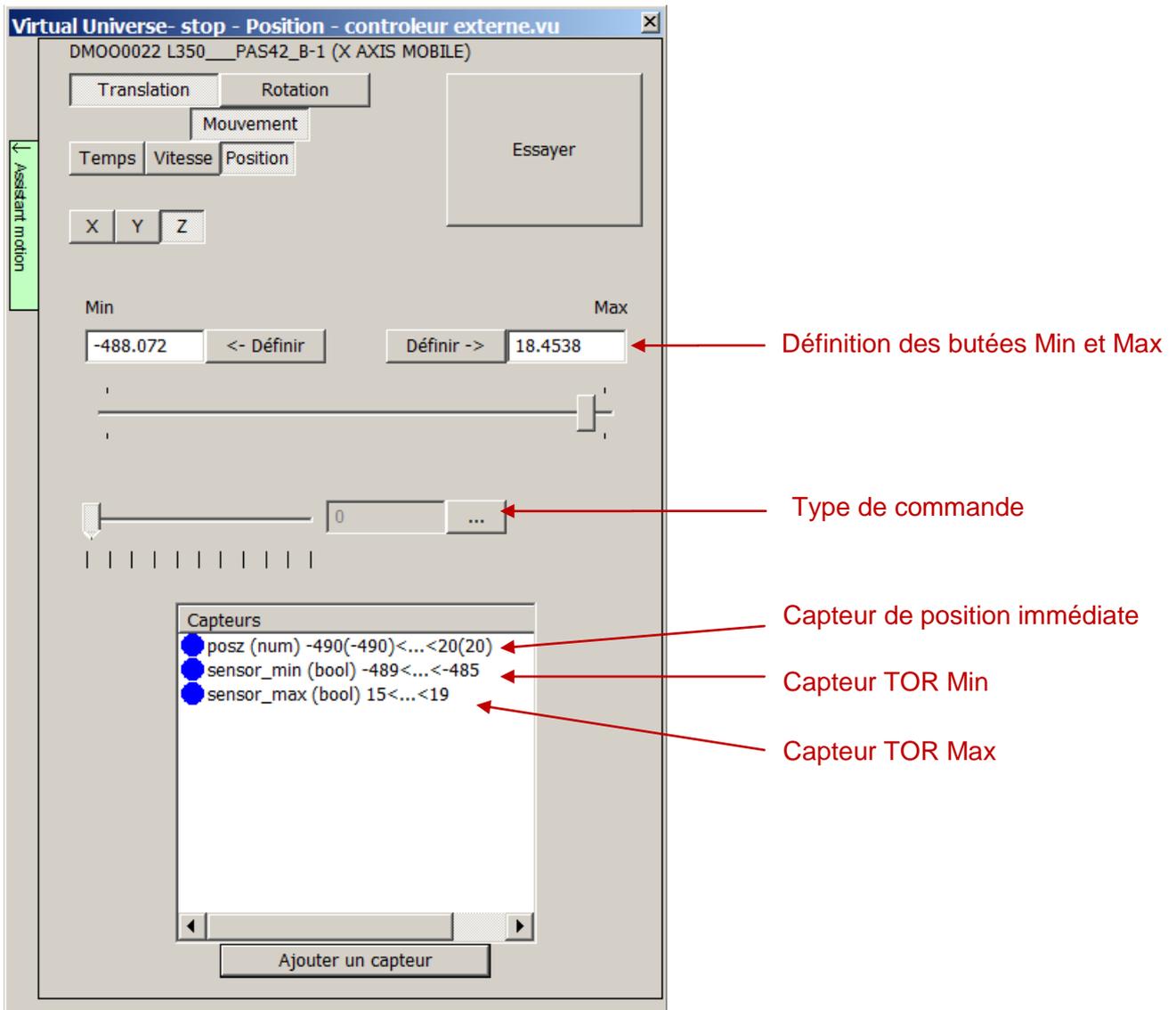
écriture directe de la position de la ressource, entrée simulation
Capteur de position « immédiate », sortie simulation
Capteur booléen min, sortie simulation
Capteur booléen max, sortie simulation

Pilotage en fonction de la position avec accélération et décélération

C'est typiquement le profil de mouvement utilisé dans le cas d'un axe, piloté par un ensemble {moteur + variateur de vitesse} recevant des consignes de position d'un contrôleur d'axe, interne à la simulation.



- Tout d'abord, sélectionnez la ressource mécanique que vous souhaitez mettre en mouvement dans l'arborescence du projet.
- Cliquez ensuite sur l'onglet vert « Assistant motion » situé à gauche de l'arborescence (l'image ci-après détaille ces premières étapes).
- La fenêtre du Motion Assistant apparaît alors à l'écran.



- Définissez les butées Min et Max, soit manuellement directement en déplaçant la ressource dans la fenêtre de rendu (ces butées ne seront pas atteintes par la ressource).
- Sélectionnez le type de commande (position directe ou relative).
- Sélectionnez l'option « Utilise un profil d'accélération/décélération pour atteindre la position... » et définissez les paramètres associés au profil.

Type de commande de mouvement (position) [X]

La valeur du comportement détermine la position
 La valeur du comportement comprise entre les deux valeurs (Début et Fin) sélectionnées ci-dessous détermine la position entre min et max

Début
 Fin

Utilise un profil accélération/décélération pour atteindre la position avec les paramètres définis ci-dessous. Atteint instantanément la position si non coché.

Vitesse maximum en unités de mesure /s (degrés /s pour les
 Temps mis pour atteindre la vitesse maximum en secondes
 Temps mis pour atteindre la vitesse nulle en seconde (décélération)

- L'ajout d'un capteur de position « immédiate » (renvoyant directement la position de la ressource selon l'axe du mouvement) permet de disposer d'un retour de position. L'image suivante illustre la création d'un tel capteur posz.

Capteur [X]

Nom

Type

Pour les capteurs de position, la mise à l'échelle détermine la valeur retournée par le capteur pour la position de début et la position de fin. Les autres valeurs sont extrapolées.

Zone de détection
 Début Fin

Mise à l'échelle
 Valeur pour début Valeur pour fin

- Il est également possible d'ajouter des capteurs TOR (tout ou rien), renvoyant 1 si la ressource mécanique se situe dans la plage de détection du capteur. L'exemple suivant illustre la création d'un capteur TOR (sensor_max) situé autour de la butée maximale qui peut, par exemple, faire office de capteur de sécurité.

Capteur [X]

Nom

Type
 Tor Position

Zone de détection
 Début: Fin:

Mise à l'échelle
 Valeur pour début: Valeur pour fin:

Supprimer ce capteur [Annuler] [OK]

Pour les capteurs de position, la mise à l'échelle détermine la valeur retournée par le capteur pour la position de début et la position de fin. Les autres valeurs sont extrapolées.

- Les comportements générés sont les suivants :

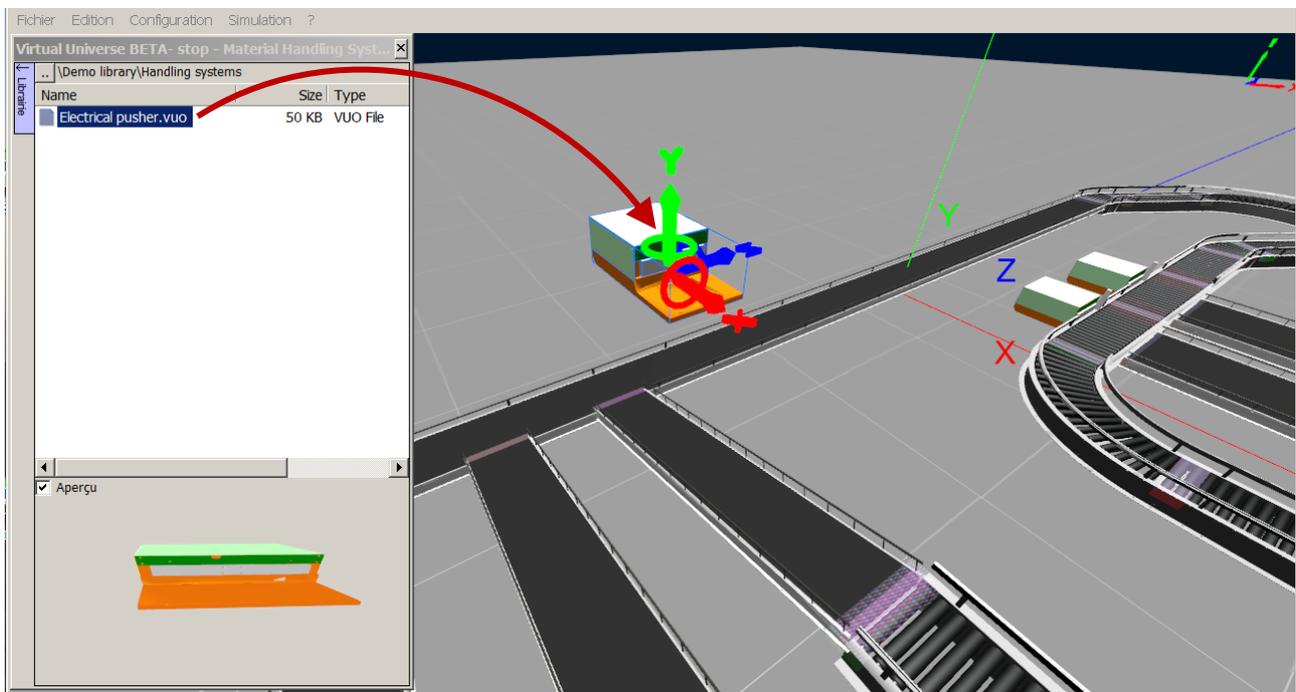
	Depth	move
	Posz	posz
	Posz_min	sensor_min
	Posz_max	sensor_max
	Reqpos	reqpos
	Maxspeed	maxspeed
	Acctime	acctime
	Decctime	decctime

Ecriture directe de la position de la ressource, variable interne
Capteur de position « immédiate », sortie simulation
Capteur booléen min, sortie simulation
Capteur booléen max, sortie simulation
Position de consigne, entrée simulation
Vitesse maximum en unité/s (mm ou degrés), entrée simulation
Temps mis pour atteindre la vitesse maximum, entrée simulation
Temps mis pour atteindre la vitesse nulle, entrée simulation

Utiliser une bibliothèque de ressources 3D intelligentes

VIRTUAL UNIVERSE PRO permet d'accéder à une bibliothèque de ressources 3D afin de capitaliser vos propres ressources 3D intelligentes dans le but de les réutiliser ultérieurement pour construire de nouveaux projets de simulation.

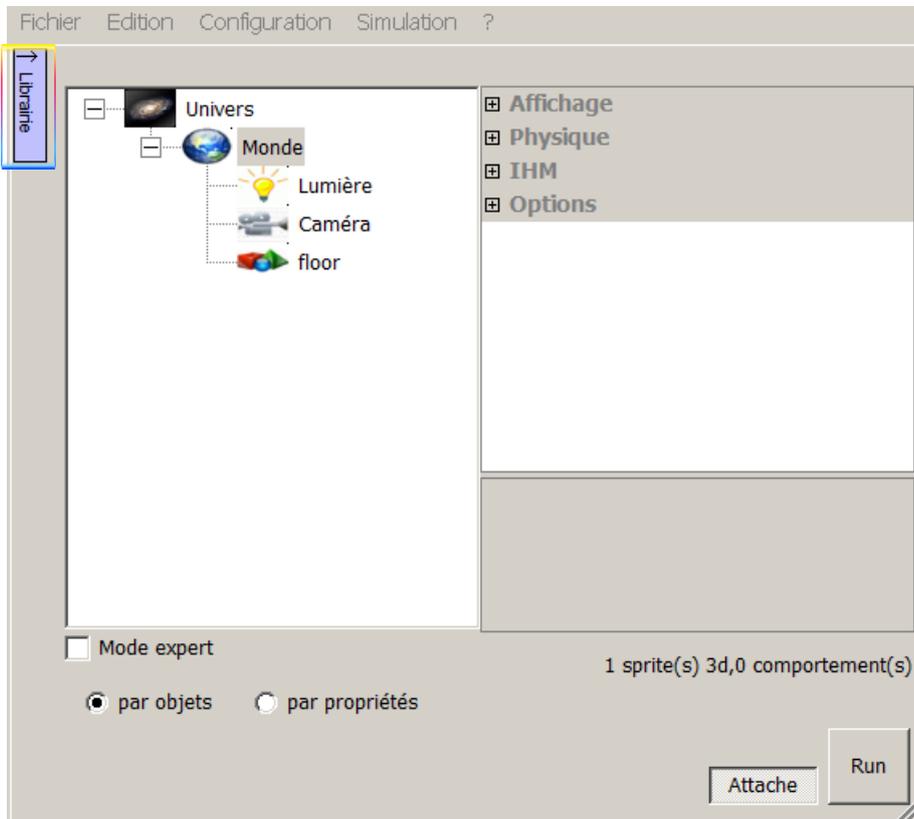
A titre d'exemple, VIRTUAL UNIVERSE PRO fournit une première bibliothèque de démonstration (appelée Demo library) comprenant quelques ressources 3D intelligentes (convoyeurs, pousoir électrique, armoire électrique, capteurs, source, puits,..) qui vous aideront à vous former et à concevoir vos propres ressources 3D intelligentes.



La bibliothèque de VIRTUAL UNIVERSE PRO se trouve dans le dossier « library », lui-même localisé dans le répertoire d'installation de VIRTUAL UNIVERSE PRO.

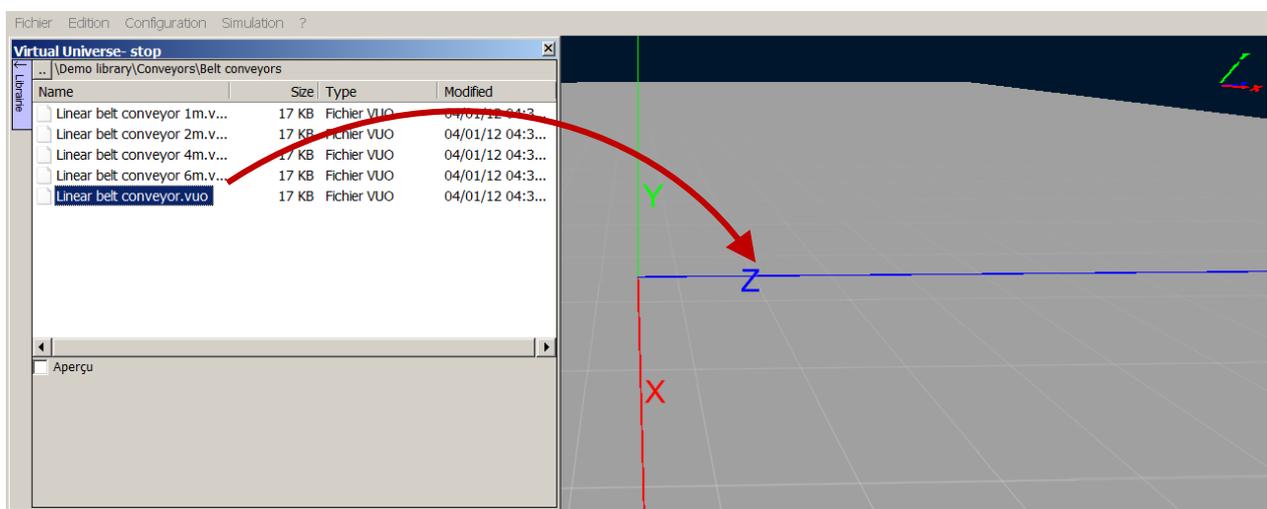
Importer une ressource 3D depuis la bibliothèque

- Pour accéder à une ressource 3D déjà disponible en bibliothèque, cliquer sur l'onglet bleu « Librairie » situé à gauche de la fenêtre de configuration.

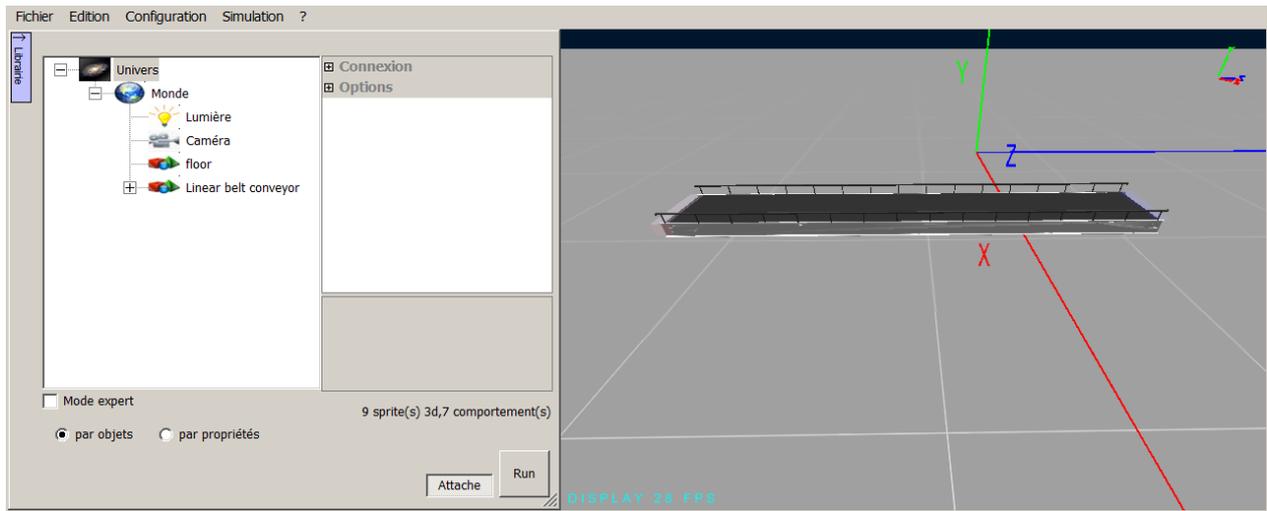


Un explorateur s'ouvre alors, vous permettant d'accéder à l'ensemble des ressources 3D présentes en bibliothèque.

- Sélectionnez dans la bibliothèque la ressource à ajouter au projet et faites un glisser-déplacer depuis la bibliothèque vers la fenêtre du rendu 3D.



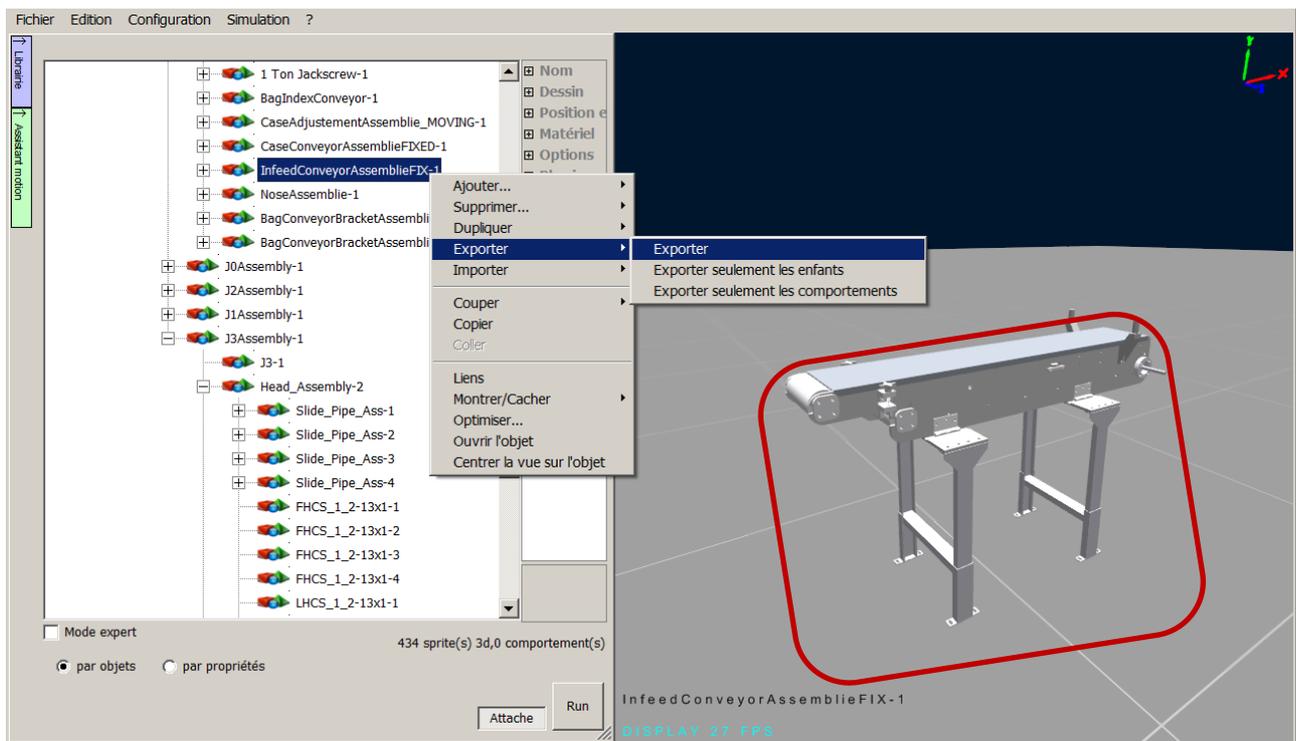
- La ressource 3D est immédiatement ajoutée au projet, et visible dans l'arborescence du projet.



Exporter une ressource 3D vers la bibliothèque

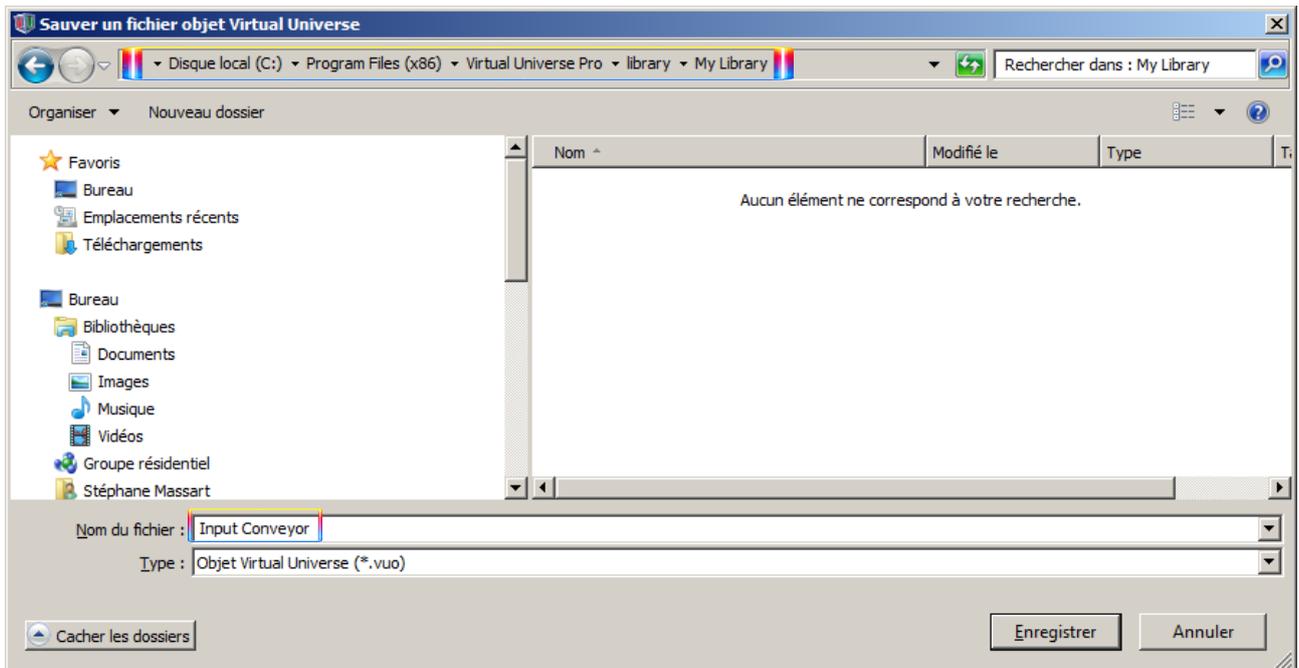
VIRTUAL UNIVERSE PRO permet de capitaliser vos propres ressources 3D intelligentes dans le but de les réutiliser ultérieurement pour construire de nouveaux projets de simulation.

- Dans l'arborescence du projet, sélectionnez la ressource 3D à ajouter dans la bibliothèque et faites un clic droit. Sélectionnez Exporter/Exporter.

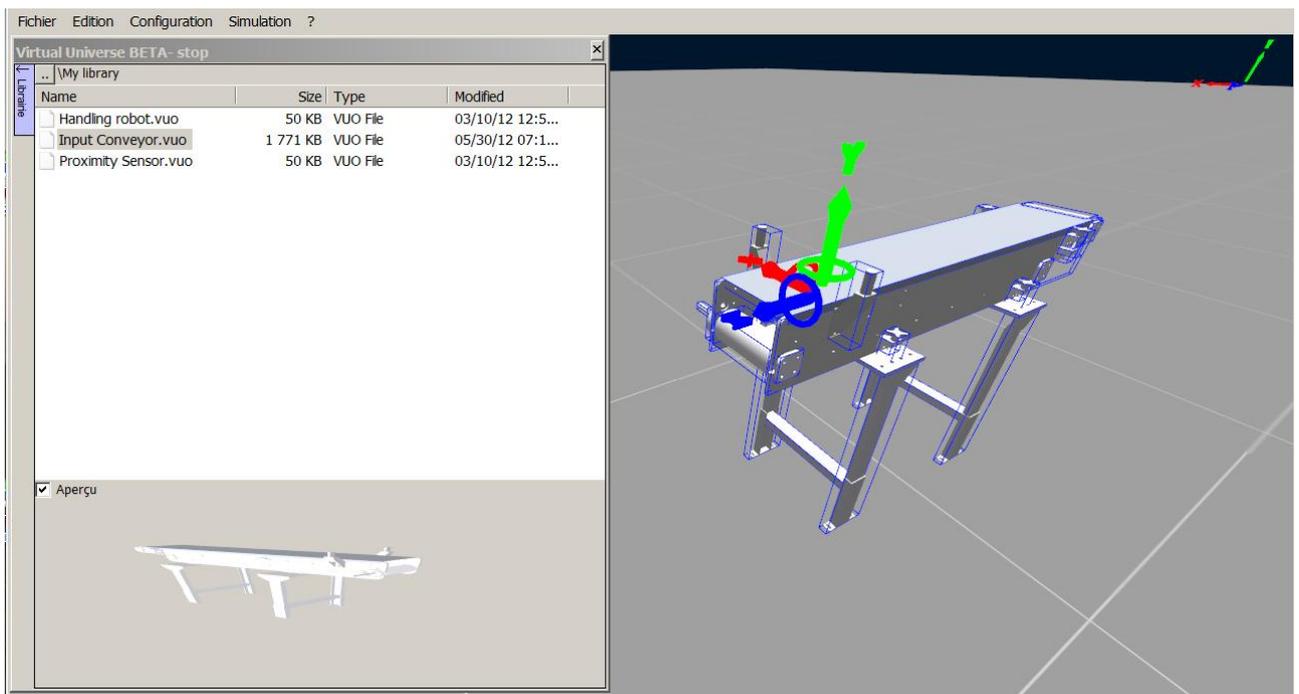


Un explorateur Windows s'ouvre alors. Dans le répertoire d'installation de VIRTUAL UNIVERSE PRO, sélectionnez le dossier « library ». Ce dossier n'est pas protégé en écriture, vous permettant de l'organiser de la manière souhaitée (ajout de nouveaux répertoires).

- Après avoir sélectionné le chemin de sauvegarde et nommé votre ressource, cliquez sur « Enregistrer ».



- La nouvelle ressource 3D est alors présente et accessible dans la bibliothèque de VIRTUAL UNIVERSE PRO, prête à être réutilisée.



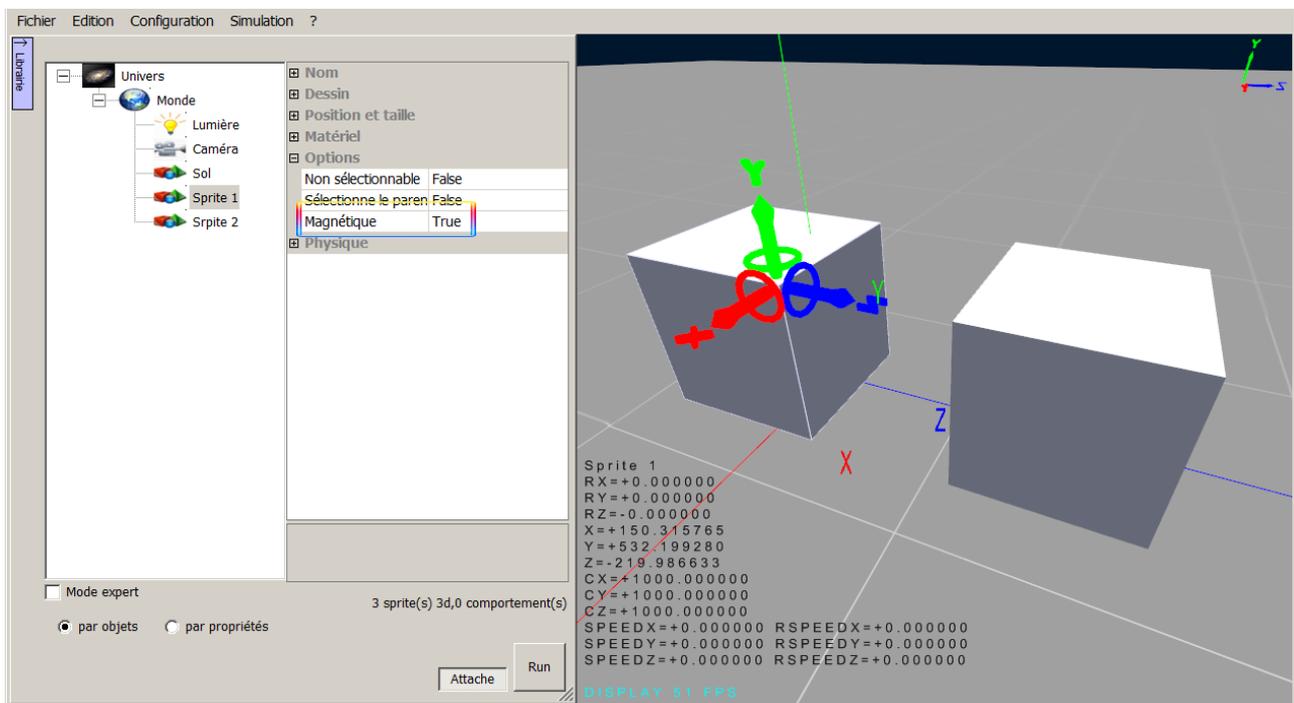
Connecter rapidement des ressources 3D avec l'option « Magnétique »

VIRTUAL UNIVERSE PRO possède une fonctionnalité d'assemblage des ressources 3D appelée « Magnétique » qui permet de connecter rapidement des ressources 3D entre elles avec la souris.

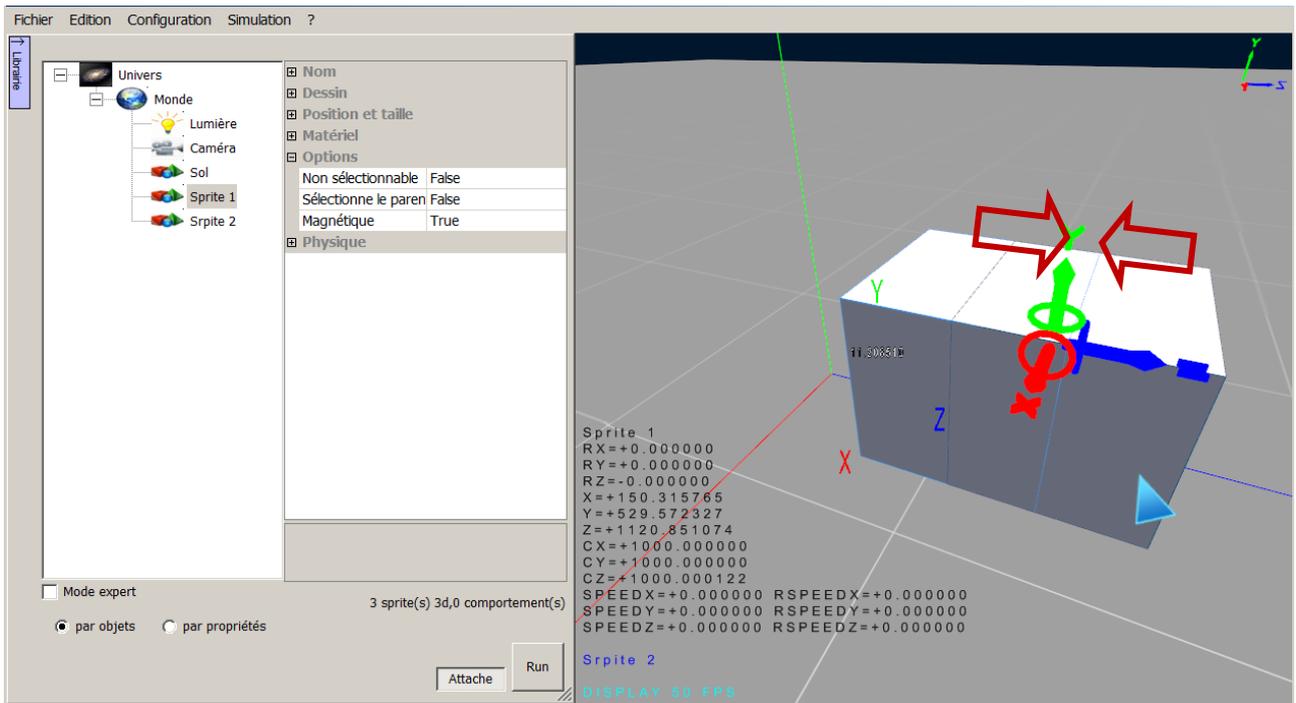
Cette fonctionnalité est particulièrement utile pour positionner rapidement dans le monde 3D les ressources d'une chaîne de production.

Dans la ressource 3D, un sprite est désigné comme le connecteur et aura cette particularité d'être « aimanté » avec les autres sprites 3D qui utilisent également la fonction « Magnétique ».

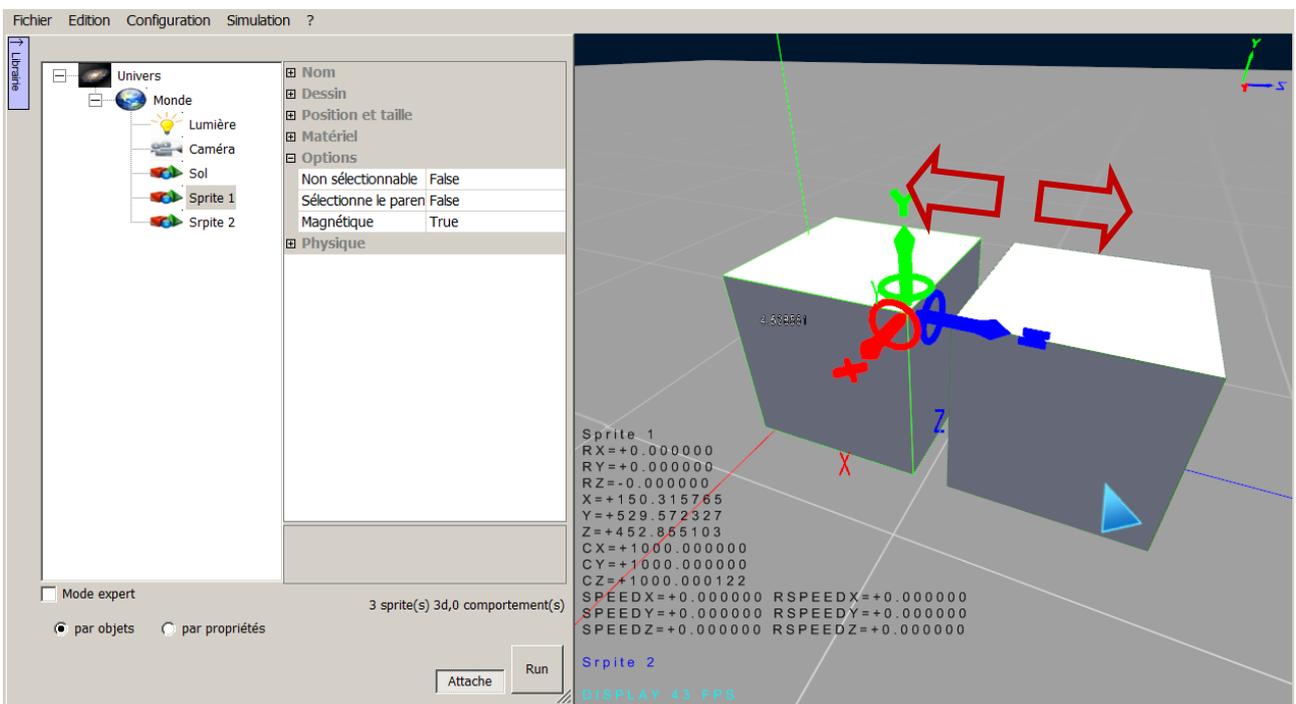
- La fonction « Magnétique » est disponible dans les propriétés du sprite.



- Lorsque les sprites exploitant la fonction « Magnétique » sont déplacés dans le monde 3D et approchés l'un de l'autre avec la souris (hors simulation), ils sont « aimantés » l'un vers l'autre. La connexion des sprites est totale lorsque la superposition des sprites est maximum.



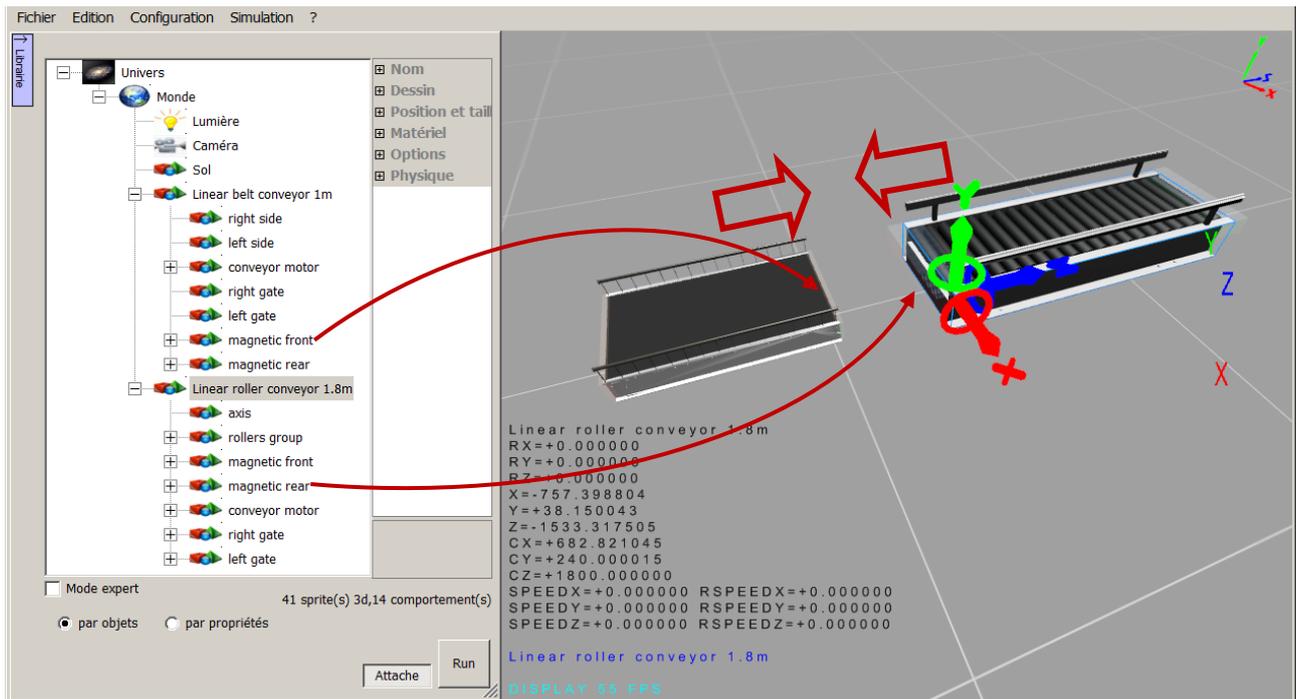
- Pour désactiver la fonction « Magnétique » et déconnecter les sprites, il suffit de maintenir la touche « Alt » enfoncée tout en déplaçant le sprite avec la souris.



Exemple

Tous les convoyeurs présents dans la bibliothèque de démonstration de VIRTUAL UNIVERSE PRO exploitent la fonction « Magnétique ».

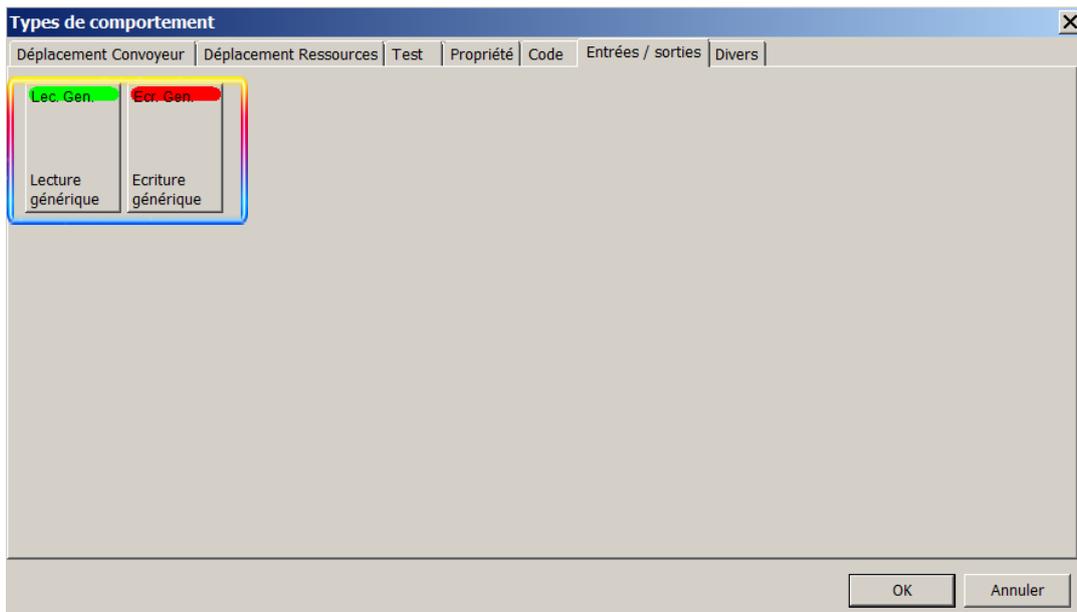
Lorsqu'un convoyeur est approché d'un autre convoyeur à l'aide la souris, il se positionne et se connecte parfaitement.



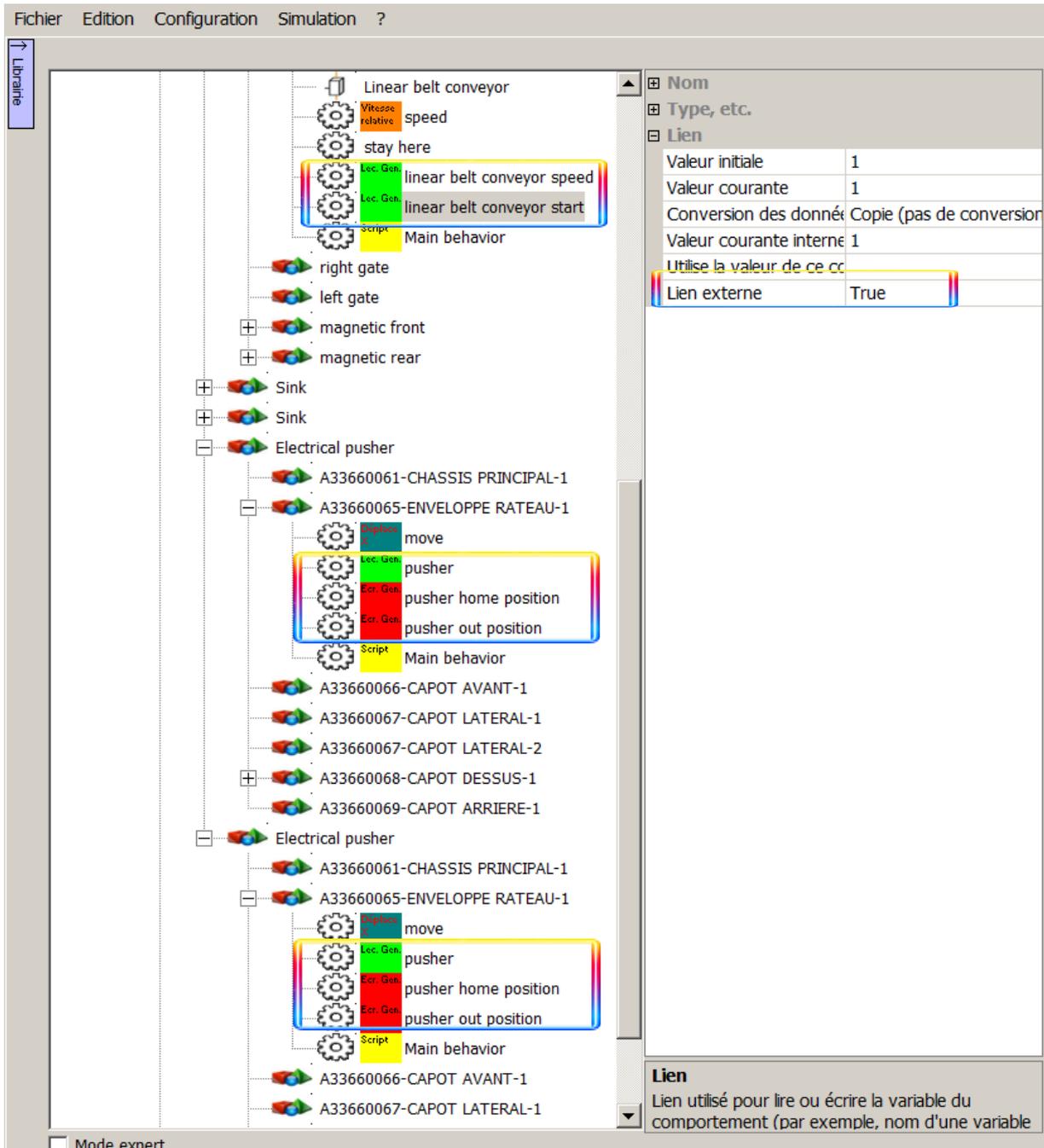
Connecter une simulation à un logiciel/contrôleur externe

Définir la liste des entrées/sorties d'une simulation

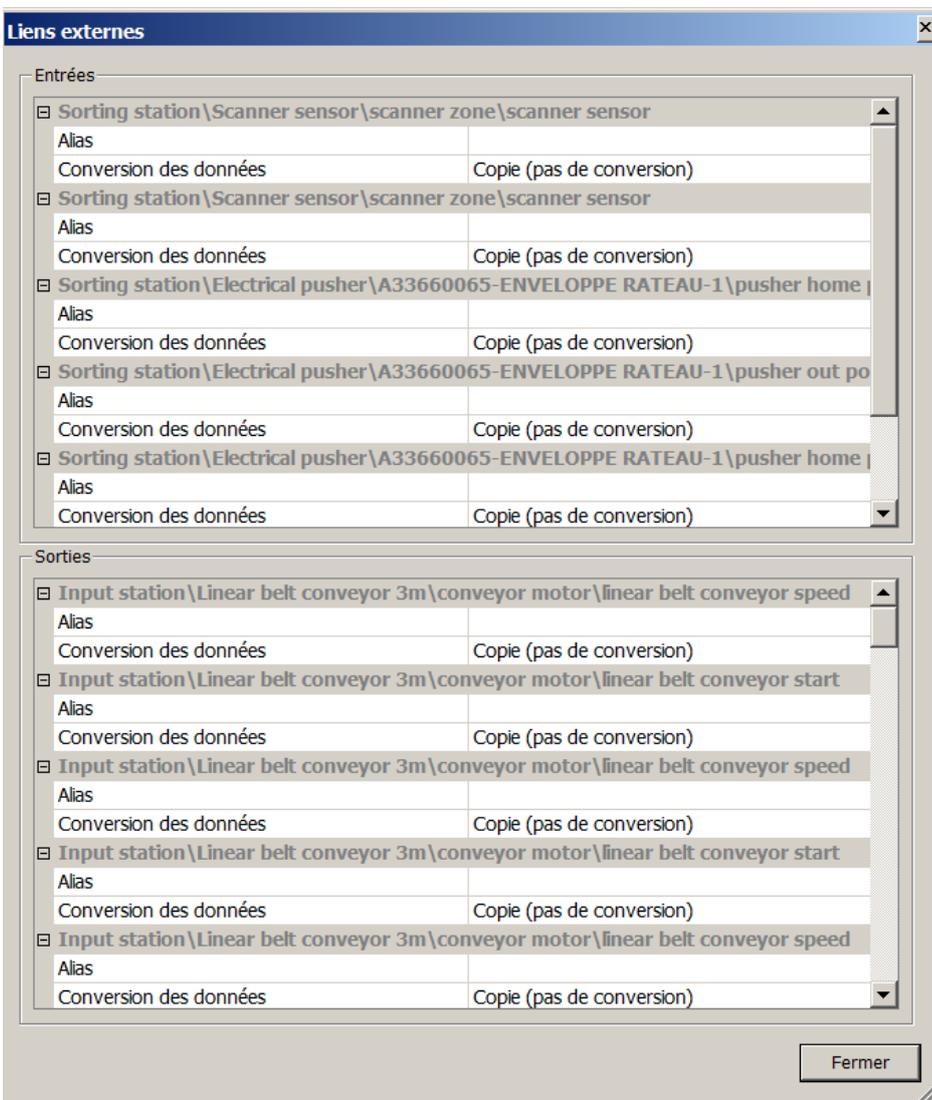
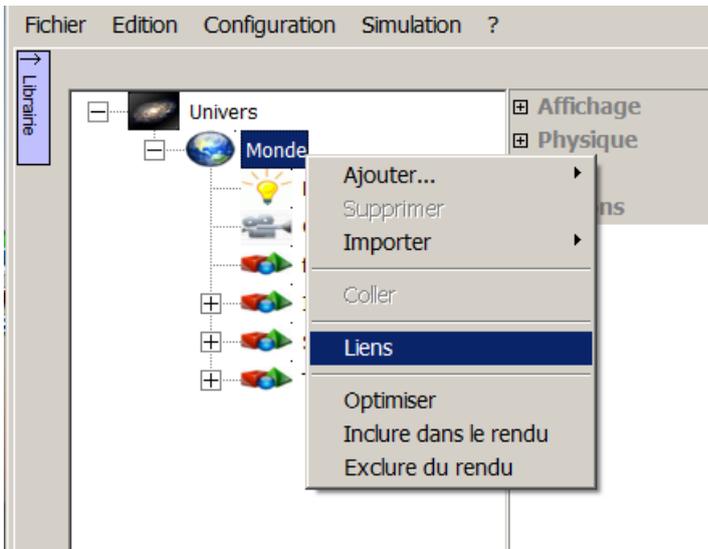
La liste des entrées/sorties d'une simulation correspond à l'ensemble des comportements de type « Lecture générique » et « Ecriture générique » ajoutés dans le projet de simulation et déclarés comme « liens externes ».



La propriété « Lien externe » est accessible dans les propriétés du comportement.



La liste des entrées/sorties du projet de simulation est visible au niveau du monde, dans la fenêtre « Liens externes ».

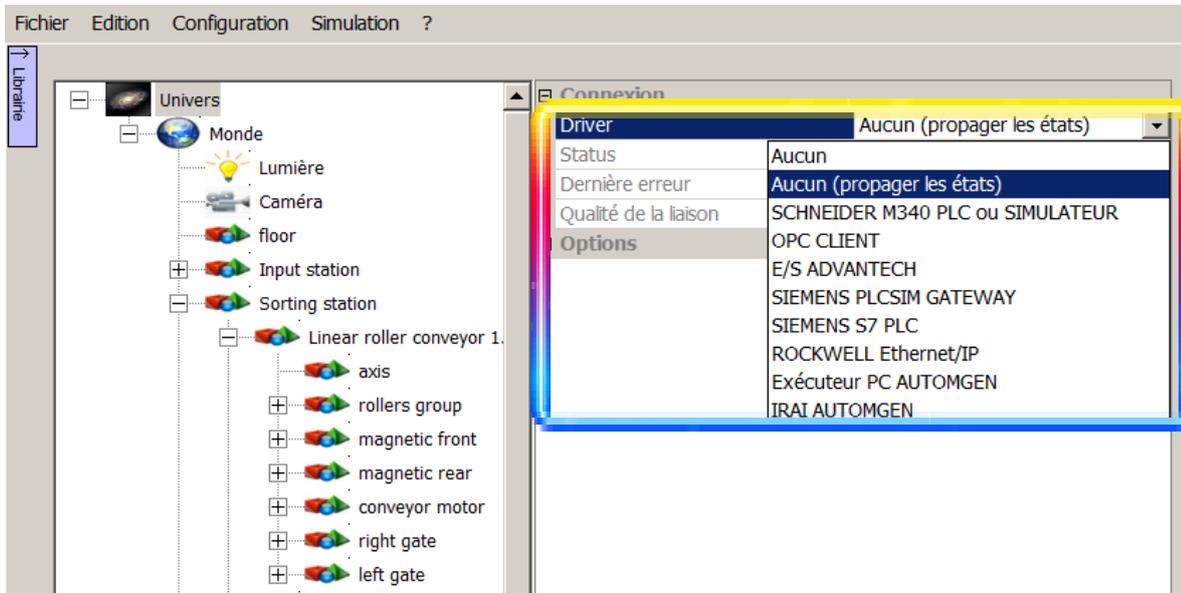


Liste des entrées/sorties de la simulation

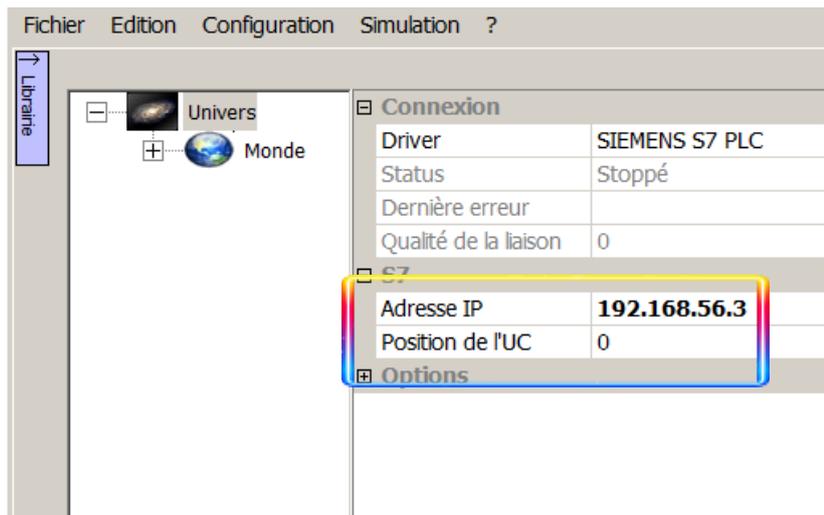
Définir la connexion avec le logiciel/contrôleur externe

Par défaut, le projet de simulation n'est pas raccordé à un logiciel/contrôleur externe.

La configuration d'une connexion avec un logiciel/contrôleur externe (automate programmable industriel, émulateur d'automate, serveur OPC,..) se fait au niveau de l'Univers, dans l'onglet « Connexion », par le choix d'un type de connecteur (driver).



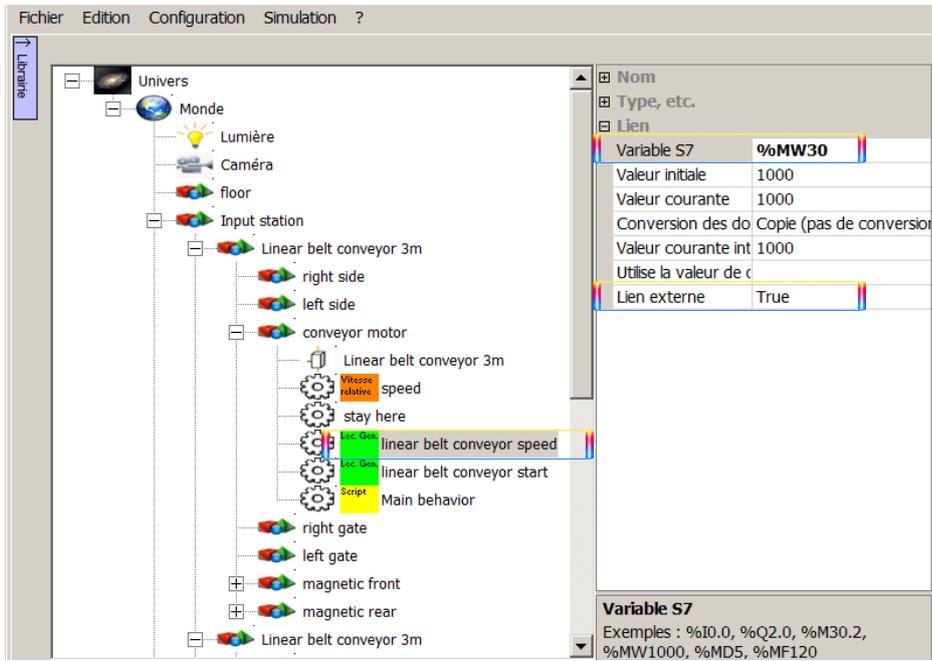
Une fois le « Driver » sélectionné, le paramétrage de la connexion (adresse IP de l'automate, position de l'UC sur le rack, nom du serveur OPC, ...) s'effectue dans l'onglet propre au driver.



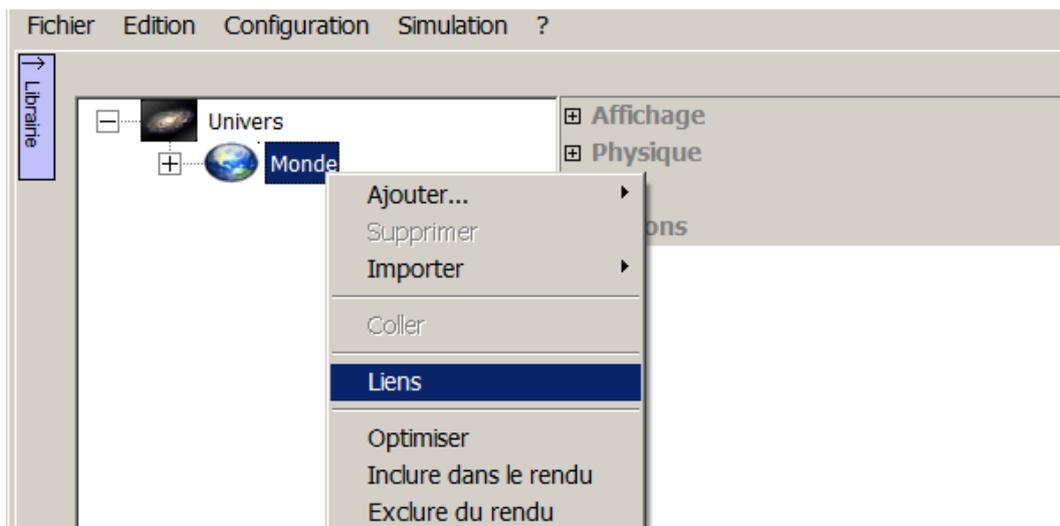
Lier les entrées/sorties de la simulation aux variables du logiciel externe

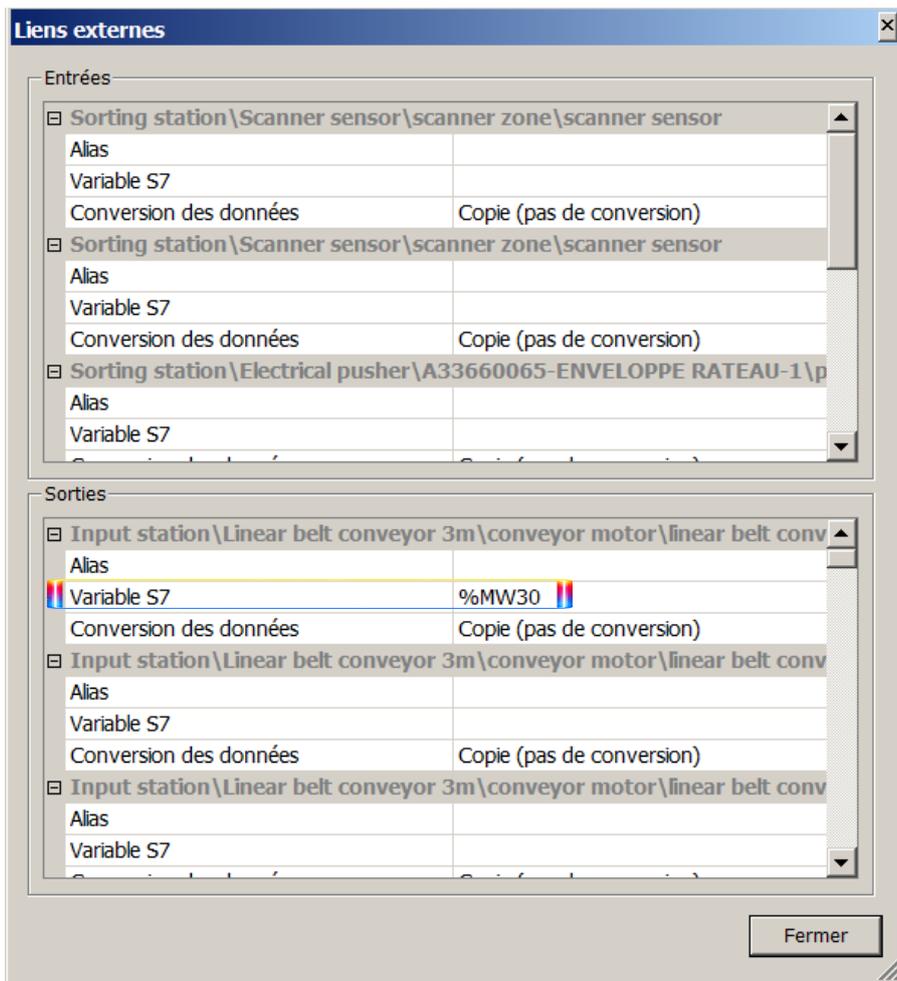
La liaison des entrées/sorties de la simulation VIRTUAL UNIVERSE PRO aux variables du logiciel/contrôleur externe peut s'effectuer à 2 niveaux :

- soit au niveau du comportement entrée/sortie (déclaré comme lien externe)

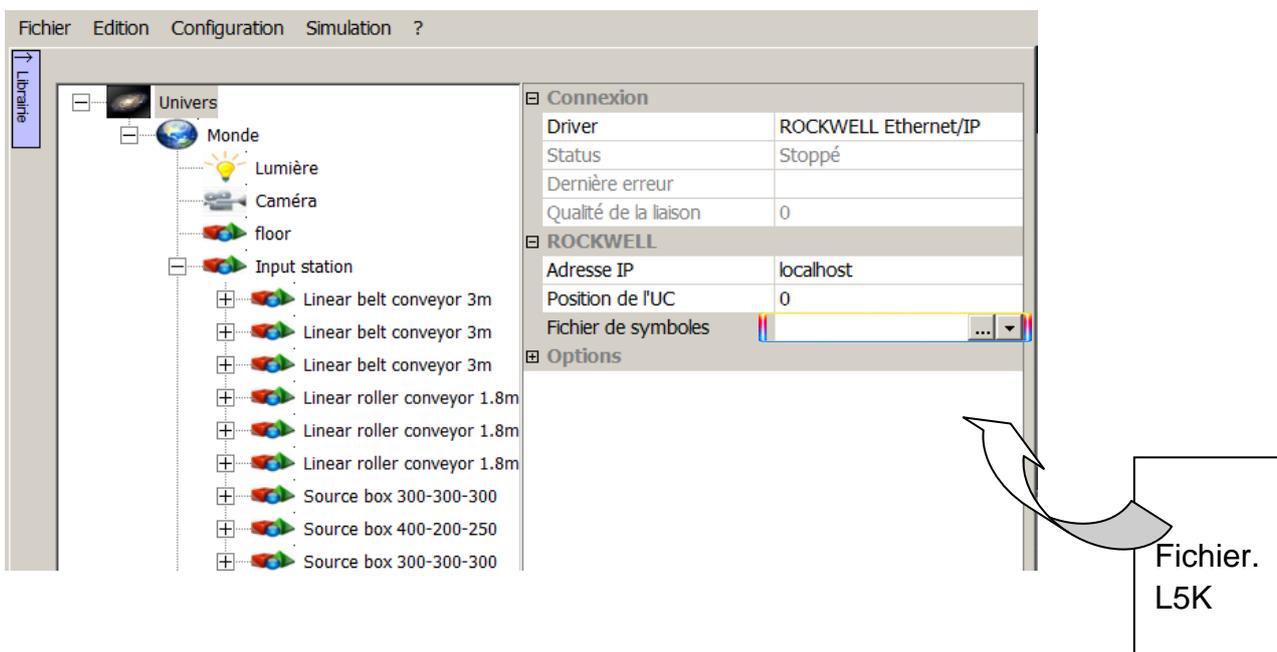


- soit au niveau du Monde, dans la fenêtre des liens externes (listant l'ensemble des entrées/sorties de la simulation), accessible par clic droit.





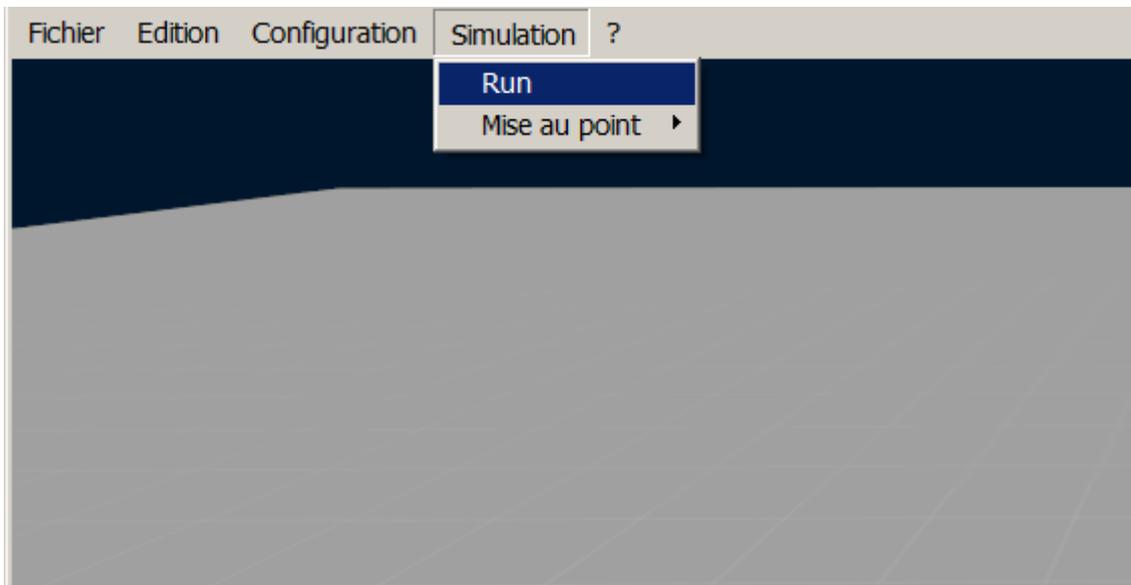
Dans le cas particulier des automates de la marque ROCKWELL/Allen Bradley, il est possible d'importer dans VIRTUAL UNIVERSE PRO une liste des tags utilisés dans le programme automate, sous forme d'un fichier au format .L5K, préalablement exporté depuis l'atelier de programmation ROCKWELL/Allen Bradley.



Tester et mettre au point une simulation

Lancement de la simulation

Le lancement de la simulation s'effectue en sélectionnant « Run » dans le menu Simulation.



Diagnostic de la simulation

En cours de simulation, un diagnostic de l'état de simulation est effectué en permanence.

- Si un problème (warning, ou erreur fatale) intervient au démarrage ou durant la simulation (erreur de conception dans la simulation, problème de communication avec le logiciel/contrôleur externe,...), une fenêtre de messages s'ouvre avec la liste des problèmes recensés :

Date	Source	Message
11:05:36 PM	Système	Passage en run
11:05:39 PM	Système	Driver connecté
12:25:08 AM	Système	Driver déconnecté
12:25:08 AM	\Sorting station\Scan...	Le comportement #0\id ne peut être trouvé
12:25:08 AM	Système	Passage en stop
12:25:19 AM	Système	Passage en run
12:25:20 AM	Système	Driver connecté
12:25:28 AM	Système	Driver déconnecté
12:25:28 AM	Système	Passage en stop
12:25:58 AM	Système	Passage en run
12:25:59 AM	\Transport line\Curve...	Erreur dans le script du comportement :line : ...
12:26:01 AM	Système	Driver connecté
12:26:01 AM	Système	Driver déconnecté
12:26:01 AM	Système	Passage en stop

Vider

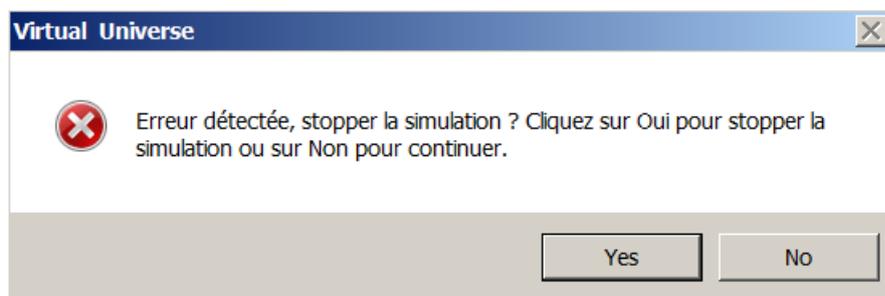
Voici la signification des couleurs :

Bleu clair: fonctionnement normal de la simulation

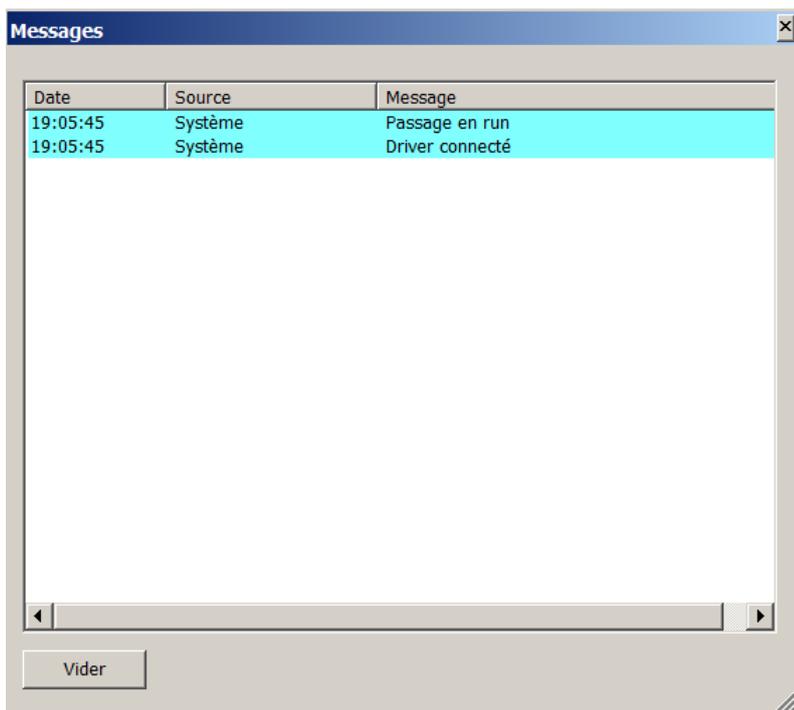
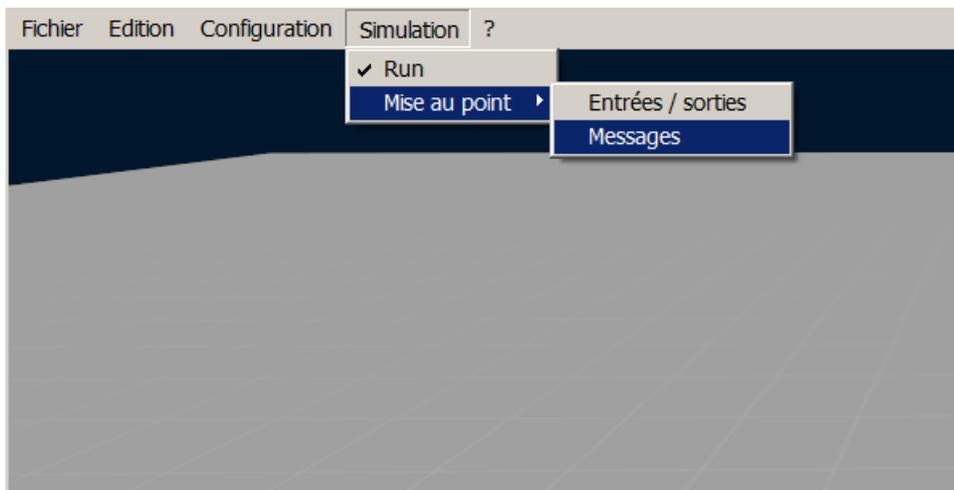
Orange: warning (défaut de conception dans la simulation)

Rouge : erreur fatale (script en erreur, problème de connexion)

Une deuxième fenêtre s'ouvre également en parallèle, invitant l'utilisateur à stopper la simulation (pour corriger le problème) ou poursuivre la simulation.



- A tout moment, en cours de simulation, l'utilisateur peut accéder à la fenêtre des messages, dans le menu Simulation/Mise au point/ Messages.

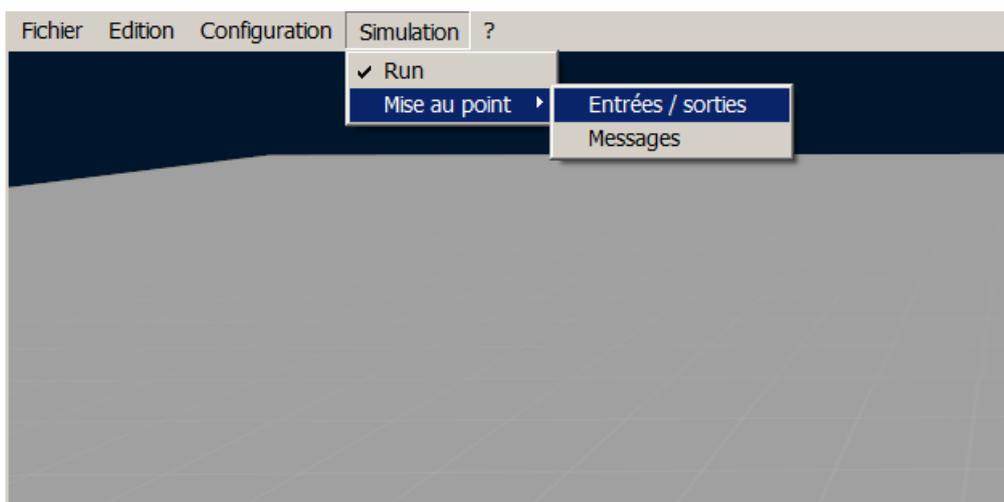


Tester la simulation en forçant ses entrées/sorties

En cours de simulation, il est possible de tester la simulation en forçant et visualisant ses variables d'entrée/sortie, lorsqu'il est connecté ou non à un logiciel/contrôleur externe.

Ceci permet par exemple, de tester la simulation avant que le programme automate ne soit complètement terminé ou disponible, ou encore de simuler des scénarios inattendus (perte de la connexion automate,...).

Dans le menu Simulation/Mise au point/Entrées-sorties, l'utilisateur accède à une fenêtre de forçage et visualisation des entrées/sorties de la simulation.



Nom	A. ▾	Variable externe	Valeur courante	Valeur courante interne	Forçage	Temps d'acquisition	Erreur
yellow light		%Q1.2	0	0		47	Aucune
stop		%I2.1	0	0		47	Aucune
start		%I2.0	0	0		47	Aucune
red light		%Q1.0	1	1		47	Aucune
power		%I1.0	1	1		47	Aucune
manual mode		%I2.4	1	1		47	Aucune
init		%I2.5	0	0		47	Aucune
gripper sensor		%I7.2	0	0		47	Aucune
gripper out request		%Q7.1	0	0		47	Aucune
gripper out position		%I7.1	0	0		47	Aucune
gripper home request		%Q7.0	0	0		16	Aucune
gripper home position		%I7.0	1	1		47	Aucune
green light		%Q1.1	0	0		16	Aucune
grab		%Q7.2	0	0		47	Aucune
emergency 3		%I2.6	0	0		47	Aucune
emergency 2		%I1.1	0	0		16	Aucune
emergency 1		%I2.2	0	0		47	Aucune
conveyor start		%Q3.0	0	0		16	Aucune
conveyor speed		%MW30	200	200		16	Aucune
conveyor detection sensor 2		%I3.1	0	0		16	Aucune
conveyor detection sensor 1		%I3.0	0	0		16	Aucune
auto mode		%I2.3	0	0		16	Aucune
Z axis out request		%Q6.2	0	0		16	Aucune
Z axis out position		%I6.2	0	0		16	Aucune
Z axis mid request		%Q6.1	0	0		16	Aucune
Z axis mid position		%I6.1	0	0		16	Aucune
Z axis home request		%Q6.0	1	1		16	Aucune
Z axis home position		%I6.0	1	1		16	Aucune
X position request		%MW50	315	315		16	Aucune

- La colonne « **Nom** » indique le nom de la variable d'entrée/sortie de la simulation
 - **rouge** = variable d'entrée de la simulation
 - **vert** = variable de sortie de la simulation
- La colonne « **Variable externe** » indique le nom de la variable du logiciel/contrôleur externe raccordé
- La colonne « **Valeur courante** » indique la valeur courante de la variable
- La colonne « **Valeur courante interne** » indique la valeur courante interne (avant éventuelle conversion) de la variable
- La colonne « **Forçage** » permet de forcer la variable
- La colonne « **Temps d'acquisition** » donne le temps instantané nécessaire pour l'échange de la variable entre la simulation et le logiciel/contrôleur externe
- La colonne « **Erreur** » indique une éventuelle erreur de connexion avec le logiciel externe

Mesurer et optimiser les performances d'une simulation

On distingue trois critères pour mesurer et optimiser les performances d'une simulation :

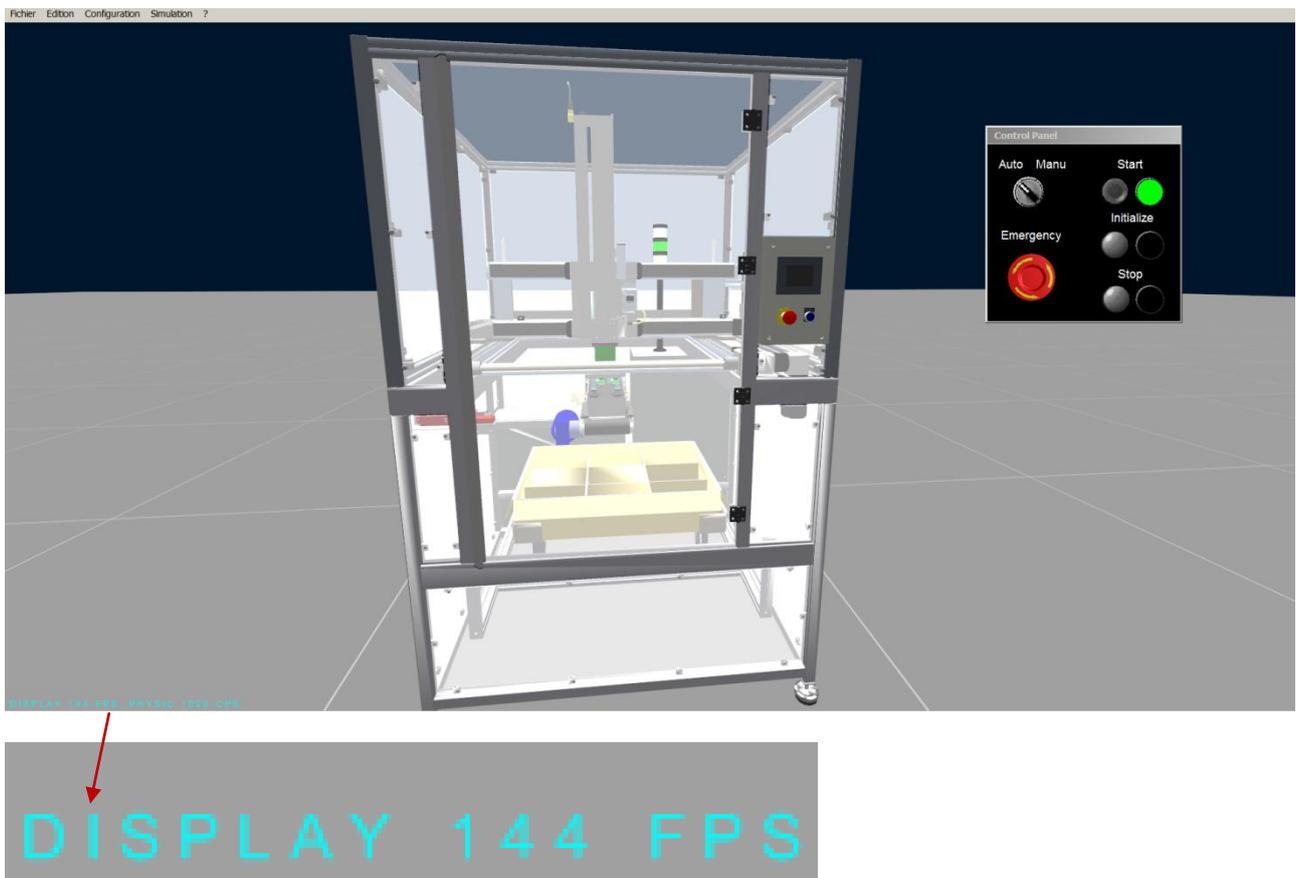
- la fréquence du rafraichissement graphique (vitesse et fluidité du rendu 3D)
- la vitesse de calcul du moteur physique (réalisme des phénomènes physiques simulés)
- les performances du dialogue avec le logiciel/contrôleur externe

Mesurer les performances graphiques

Il est important de savoir mesurer et optimiser les performances graphiques d'une simulation. Ces performances graphiques jouent directement sur la fluidité et la qualité visuelle du rendu 3D.

Les performances graphiques d'une simulation représentent le nombre d'images affichées par seconde dans le rendu 3D (FPS), durant la simulation et hors simulation. On parle encore de fréquence du rafraichissement graphique.

Dans VIRTUAL UNIVERSE PRO, cette information est en permanence affichée en bas à gauche dans la fenêtre du rendu 3D.



Les performances graphiques d'une simulation dépendent fortement des facteurs prédominants suivants :

- La qualité et les performances de la carte graphique installée sur l'ordinateur exécutant la simulation
- La quantité et la complexité des géométries 3D affichées dans le rendu 3D de l'émulateur
- Le nombre de caméras (vues) utilisées dans la simulation
- L'ouverture ou la fermeture de la fenêtre de configuration pendant la simulation

On considère que les performances graphiques d'une simulation sont bonnes lorsque la fréquence du rafraîchissement graphique reste toujours supérieure à **15 FPS, soit 15 images par seconde**.

Optimiser les performances graphiques

Carte graphique

Pour obtenir de bonnes performances graphiques avec les émulateurs 3D VIRTUAL UNIVERSE PRO, il est vivement recommandé d'utiliser un ordinateur récent équipé d'une bonne carte graphique.

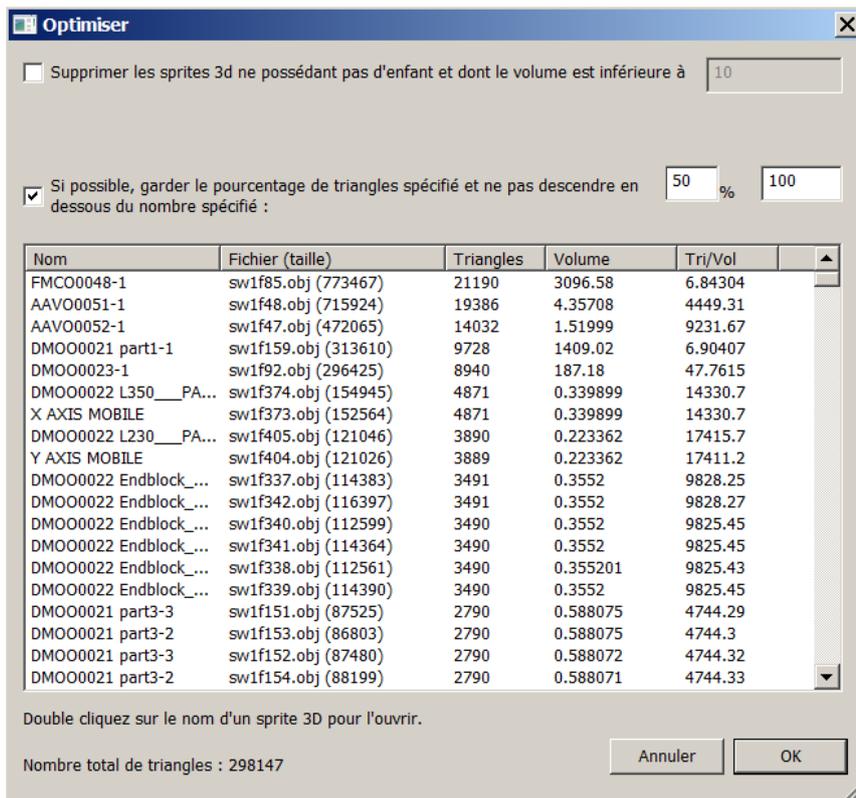
Les cartes graphiques réservées au monde des jeux vidéo sont particulièrement performantes et adaptées pour les émulateurs 3D VIRTUAL UNIVERSE PRO.

Alléger les géométries 3D

Les modèles 3D issus des logiciels de CAO possèdent souvent des géométries 3D très complexes (nombre de triangles) et des petites pièces qui ne sont pas toujours nécessaires aux émulateurs 3D VIRTUAL UNIVERSE PRO et peuvent au contraire contribuer à diminuer fortement les performances graphiques.

Pour alléger les modèles CAO 3D, VIRTUAL UNIVERSE possède un outil d'optimisation des géométries 3D qu'il est recommandé d'utiliser systématiquement.

Pour plus d'informations sur l'usage de cet outil, voir [Alléger les modèles CAO 3D](#)



Ouverture de la fenêtre de configuration

Lorsqu'elle est attachée à la fenêtre de rendu, la fenêtre de configuration peut contribuer à diminuer la fréquence du rafraichissement graphique.

En cours de simulation, il est recommandé de laisser la fenêtre de configuration fermée ou flottante.

Une autre possibilité consiste à utiliser un deuxième écran, sur lequel ouvrir la fenêtre de configuration (et la fenêtre de mise au point), afin de ne garder que la fenêtre de rendu 3D ouverte sur le premier écran. Ainsi, les performances graphiques de la simulation en cours de simulation ne sont pas réduites.

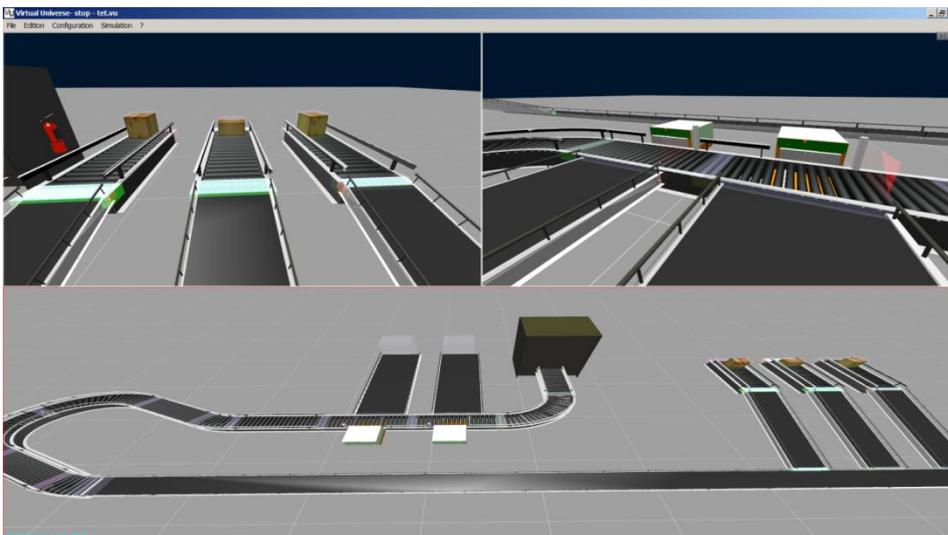
Fenêtre de configuration

Fenêtre de rendu 3D



Usage des multi-caméras

L'usage des multi-caméras dans une simulation peut contribuer à faire baisser ses performances graphiques.



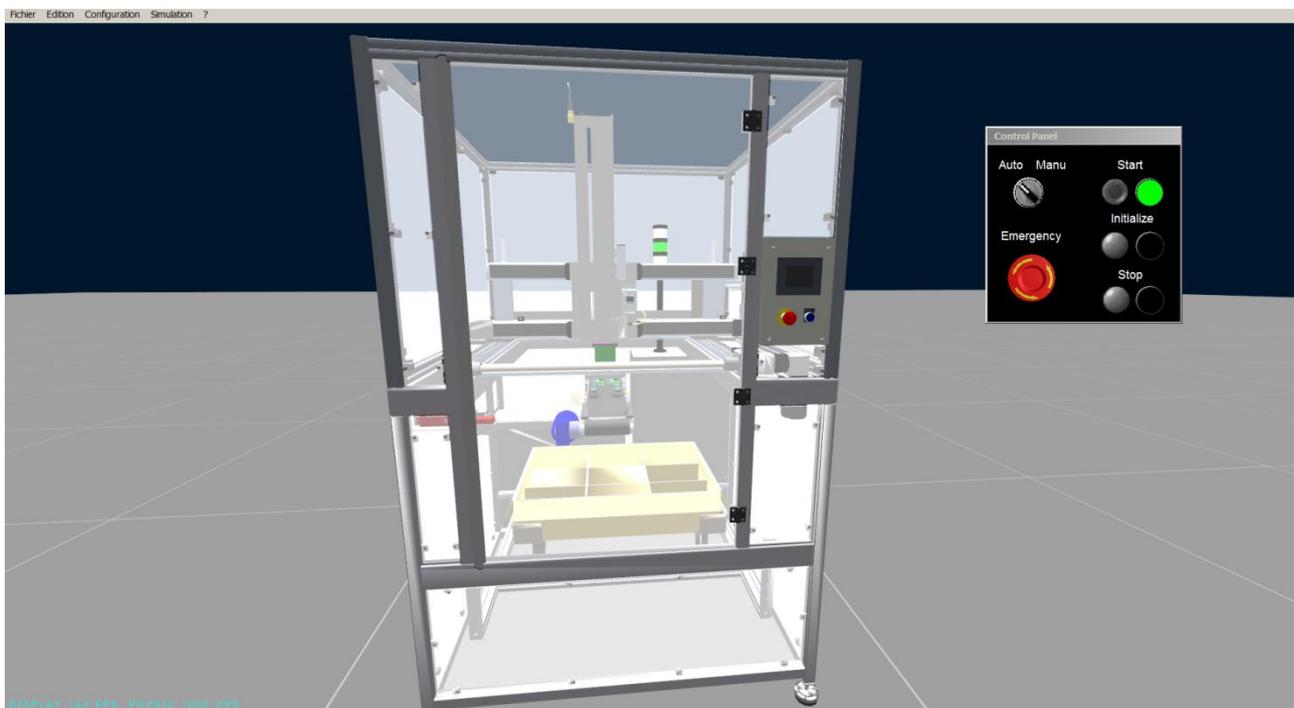
Mesurer les performances du moteur physique

Il peut être utile de savoir mesurer et optimiser les performances du moteur physique utilisé dans VIRTUAL UNIVERSE PRO, pour les émulateurs 3D qui nécessitent la simulation de nombreux phénomènes physiques (gravité des corps ; frictions, collisions).

Les performances du moteur physique exploité dans VIRTUAL UNIVERSE PRO jouent directement sur le réalisme des phénomènes physiques simulés.

Les performances physiques s'expriment en nombre de calcul du moteur physique par seconde (CPS).

Dans VIRTUAL UNIVERSE PRO, cette information est en permanence affichée en bas à gauche dans la fenêtre du rendu 3D.



PHYSIC 1000 CPS

Les performances physiques d'une simulation dépendent fortement des facteurs prédominants suivants :

- la puissance du processeur (CPU) de l'ordinateur utilisé pour exécuter la simulation.
- le nombre et la forme des objets 3D pris en compte par le moteur physique en simulation.

On considère que les performances physiques d'une simulation sont bonnes lorsque le nombre de calculs du moteur physique, en cours de simulation, reste toujours supérieur à **100 CPS, soit 100 calculs par seconde**.

Optimiser les performances du moteur physique

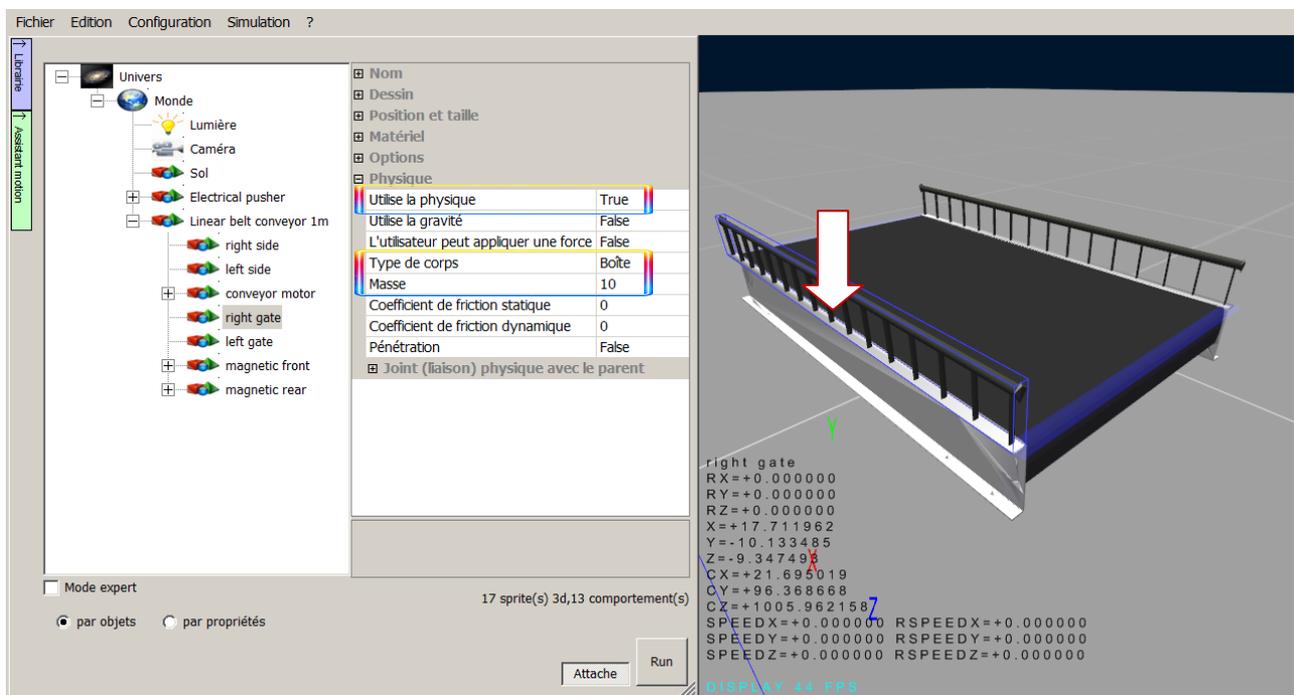
Processeur de l'ordinateur

Pour obtenir de bonnes performances physiques dans les émulateurs 3D VIRTUAL UNIVERSE PRO, il est vivement recommandé d'utiliser un ordinateur récent équipé d'un processeur performant (multi-cœurs).

Forme des sprites 3D prise en compte par le moteur physique

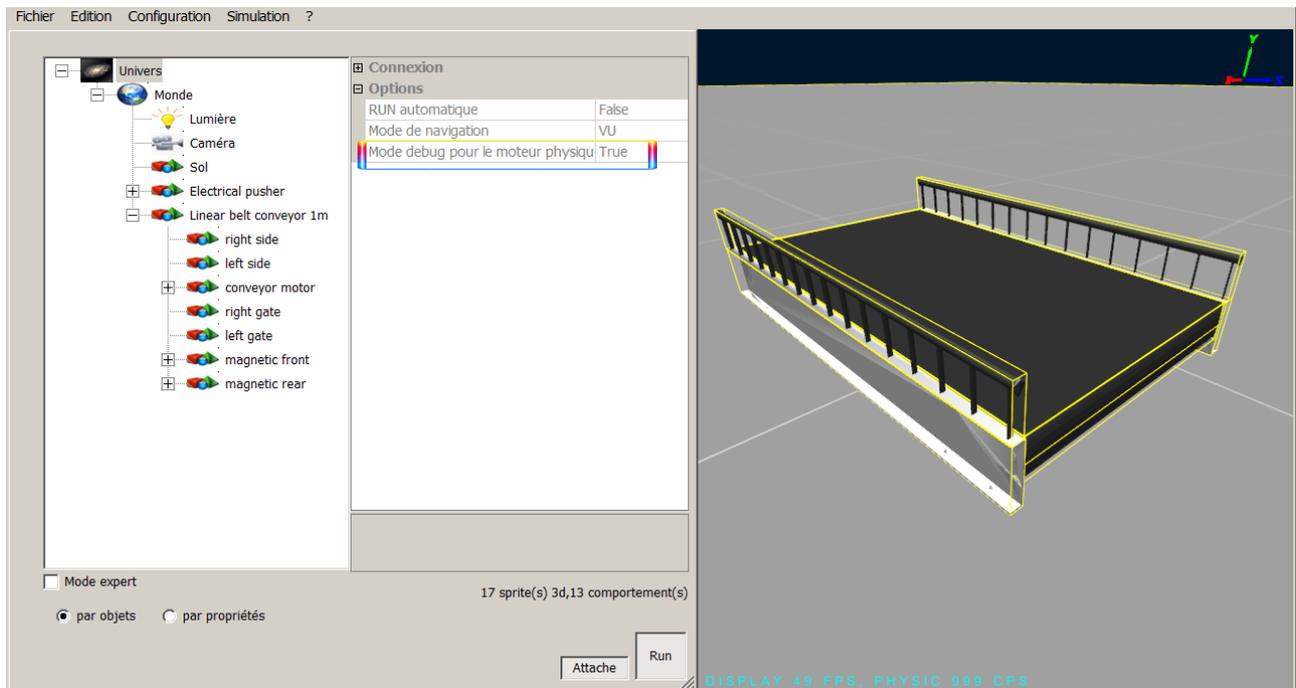
Par défaut, lorsqu'ils sont ajoutés dans le projet, les sprites 3D ne sont pas exposés au moteur physique, leur option « Utilise la physique » est désactivée

Pour optimiser les performances du moteur physique, il est recommandé de ne pas soumettre un trop grand nombre de sprites 3D au moteur physique et de choisir le type de corps « boîte » lorsque cela est possible.



Le type de corps « Quelconque » ou «convexe » est celui qui donne le plus grand réalisme dans le phénomène physique simulé, mais qui sollicite également le plus le moteur physique.

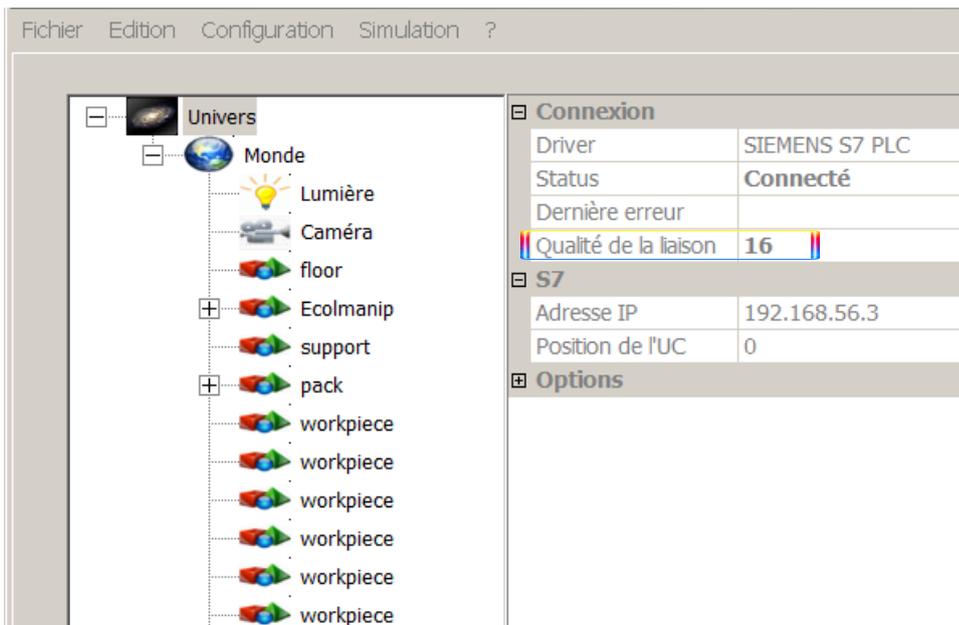
Il est souvent utile dans les phases de mise au point de visualiser les géométries manipulées par le moteur physique en activant l'option « Mode debug pour le moteur physique » dans les propriétés de l'Univers.



Performances du dialogue avec le logiciel/contrôleur externe

Les performances du dialogue avec le logiciel/contrôleur externe sont exprimées par le temps nécessaire (en milliseconde) à l'échange de l'ensemble des variables partagées entre la simulation et le logiciel/contrôleur externe.

Cette information est accessible, en cours de simulation, dans l'onglet Univers/Connexion/Qualité de la liaison.



La fenêtre de mise au point détaille également le temps d'échange pour chacune des variables échangées avec un logiciel externe.

The 'Mise au point' window displays a table with the following columns: Nom, Variable externe, Valeur courante, Valeur courante interne, Forçage, and Erreur. The table lists various variables and their states:

Nom	Variable externe	Valeur courante	Valeur courante interne	Forçage	Erreur
yellow light	%Q1.2	0	0	47	Aucune
stop	%I2.1	0	0	47	Aucune
start	%I2.0	0	0	47	Aucune
red light	%Q1.0	1	1	47	Aucune
power	%I1.0	1	1	47	Aucune
manual mode	%I2.4	1	1	47	Aucune
init	%I2.5	0	0	47	Aucune
gripper sensor	%I7.2	0	0	47	Aucune
gripper out request	%Q7.1	0	0	47	Aucune
gripper out position	%I7.1	0	0	47	Aucune
gripper home request	%Q7.0	0	0	16	Aucune
gripper home position	%I7.0	1	1	47	Aucune
green light	%Q1.1	0	0	16	Aucune
grab	%Q7.2	0	0	47	Aucune
emergency 3	%I2.6	0	0	47	Aucune
emergency 2	%I1.1	0	0	16	Aucune
emergency 1	%I2.2	0	0	47	Aucune
conveyor start	%Q3.0	0	0	16	Aucune
conveyor speed	%MW30	200	200	16	Aucune
conveyor detection sensor 2	%I3.1	0	0	16	Aucune
conveyor detection sensor 1	%I3.0	0	0	16	Aucune
auto mode	%I2.3	0	0	16	Aucune
Z axis out request	%Q6.2	0	0	16	Aucune
Z axis out position	%I6.2	0	0	16	Aucune
Z axis mid request	%Q6.1	0	0	16	Aucune
Z axis mid position	%I6.1	0	0	16	Aucune
Z axis home request	%Q6.0	1	1	16	Aucune
Z axis home position	%I6.0	1	1	16	Aucune
X position request	%MW50	315	315	16	Aucune

Générer des émulateurs 3D indépendants (players)

VIRTUAL UNIVERSE PRO permet de générer des émulateurs 3D indépendants, sous forme de fichiers exécutables (au format .exe) appelés « players ».

Un player embarque toute l'intelligence de simulation du projet de simulation dont il est issu. Il peut être connecté à un logiciel/contrôleur externe (suivant le type de connexion définie dans la simulation d'origine) et propose, en cours de simulation, les mêmes fonctionnalités que le projet de simulation d'origine.

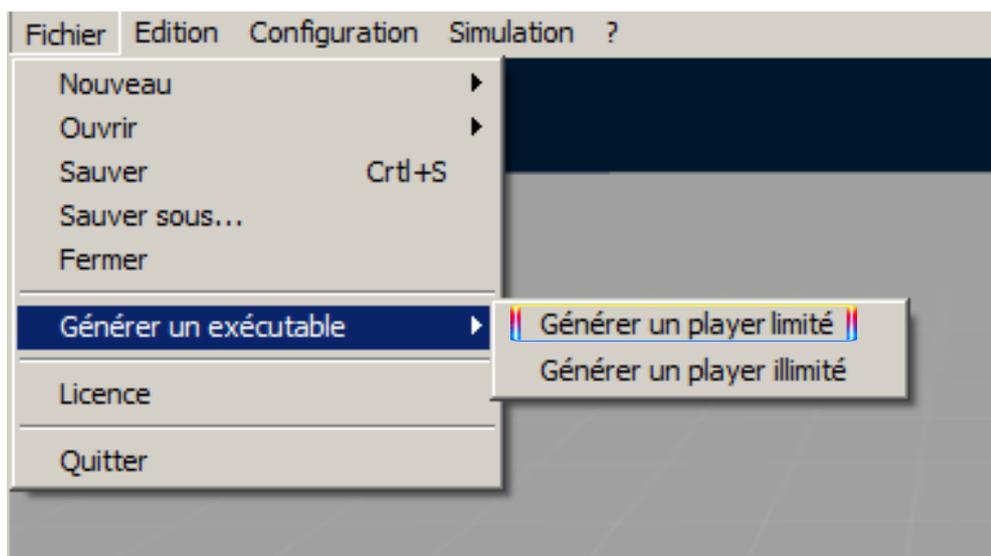
Il existe 2 types de players :

- player limité (à une durée d'utilisation de 2 minutes).
- player illimitée (sans limite d'utilisation).

Players à durée limitée

VIRTUAL UNIVERSE PRO permet de générer des players, limités à une durée d'utilisation de 2 minutes (fermeture du player au bout de 2 minutes après son ouverture).

La génération d'un player à durée limitée est accessible dans le menu Fichier/Générer un exécutable/Générer un player limité.

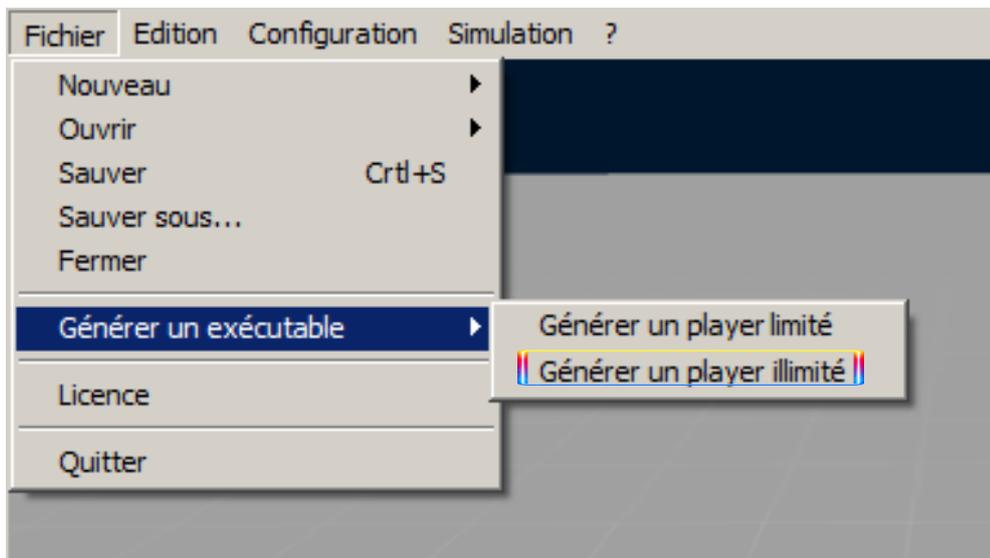


Players à durée illimitée

VIRTUAL UNIVERSE PRO permet de générer des players non limités en termes de durée d'utilisation, mais protégés par un code de sécurité unique.

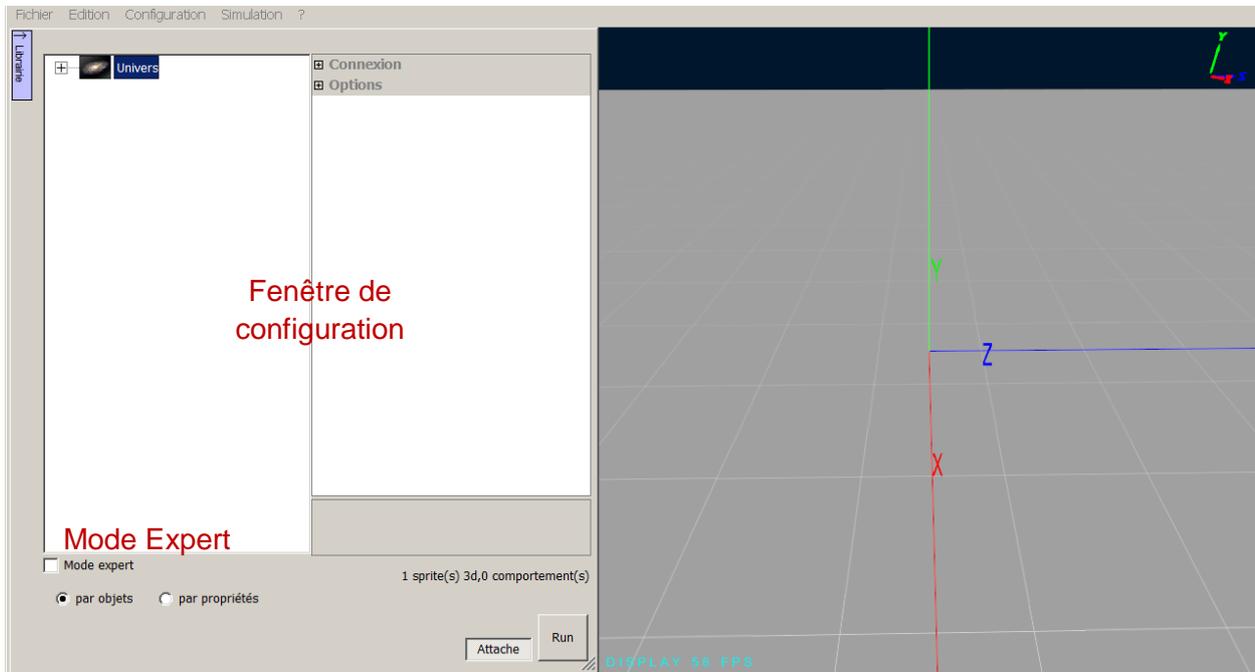
La génération d'un player à durée illimitée est accessible dans le menu Fichier/Générer un exécutable/Générer un player illimité.

Cette génération nécessite l'obtention d'un code de sécurité. Pour plus d'informations, voir [Enregistrer un code de sécurité \(player illimité\)](#).



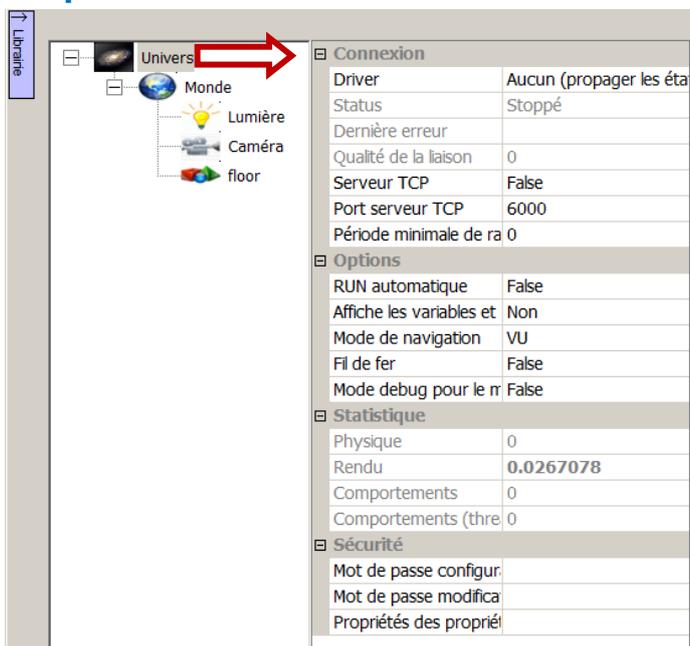
Propriétés détaillées d'une simulation

La fenêtre de configuration permet d'accéder aux propriétés détaillées d'un projet de simulation. Certaines de ces propriétés sont réservées aux utilisateurs avancés et ne sont accessibles qu'en activant le mode Expert.



Propriétés de l'Univers

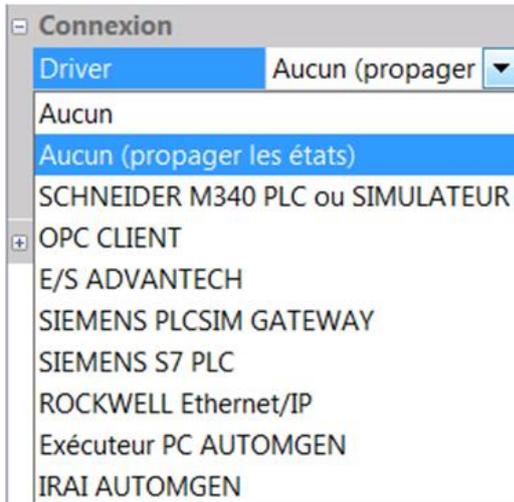
Propriétés détaillées de l'Univers



Propriétés de l'Univers

Connexion

Driver : Détermine la connexion avec un logiciel/contrôleur externe (logiciel avec lequel dialogue VIRTUAL UNIVERSE PRO). Les pilotes disponibles sont les suivants :



Status : Etat actuel de la connexion (Stoppé, Logiciel externe non prêt, Connecté).

Dernière erreur : Dernière erreur de communication.

Qualité de la liaison : Temps nécessaire à l'échange de toutes les données avec le logiciel/contrôleur externe (en millisecondes).

Serveur TCP (mode expert) : Activation d'un serveur TCP permettant une ou plusieurs applications externes de dialoguer avec VIRTUAL UNIVERSE PRO.

Port serveur TCP (mode expert) : Numéro de port du serveur TCP.

Période minimale de rafraîchissement des variables externes (mode expert): Limite minimale (en ms) à l'échange de l'ensemble des variables. 100 : échange de l'ensemble des variables toutes les 100 ms au minimum.

Options

Run automatique : Lance automatiquement la simulation à l'ouverture du projet. Pour désactiver ce lancement automatique, maintenir la touche « Shift » à l'ouverture.

Affiche les variables et les états (mode expert) : Affichera dans la fenêtre de rendu les noms de variables ainsi que les états pour les Comportements faisant référence à une variable du logiciel externe.

Mode de navigation : VU (navigation à la souris) ou IRIS3D (en cas d'absence de souris, ce mode permet une navigation à l'aide de flèches).

Fil de fer (mode expert) : Dessine tous les Sprites 3D en fil de fer.

Mode debug pour le moteur physique : Si vrai, les volumes utilisés par le moteur physique sont affichés dans la fenêtre de rendu (lignes de couleurs jaunes). Ceci est très utile en phase de développement d'un projet utilisant le moteur physique pour visualiser les volumes pris en compte par le moteur physique.

Statistique (mode expert)

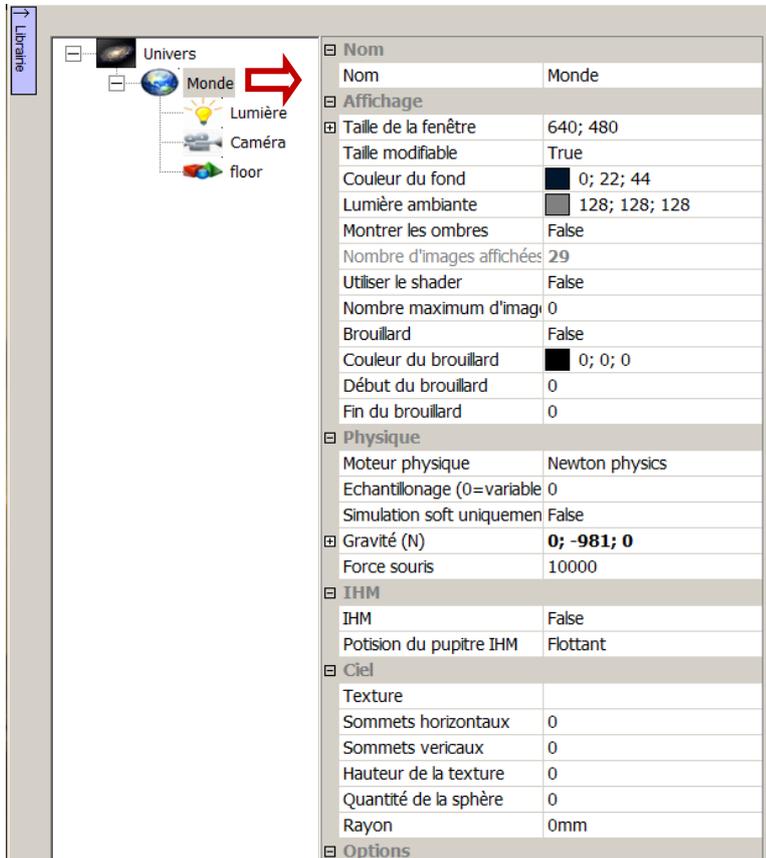
Donne des informations sur le temps nécessaire au rendu, à l'exécution du moteur physique et au traitement des comportements.

Sécurité (mode expert)

Permet de définir un mot de passe qui contrôle l'accès à la fenêtre de configuration ainsi qu'à la modification du projet.

Propriétés du Monde

Propriétés détaillées du Monde



Propriétés du
Monde

Nom (mode expert)

Permet de désigner un monde par son nom.

Affichage

Taille de la fenêtre (mode expert) : Taille de la fenêtre de rendu en pixels en mode plein écran. Le mode plein écran s'active en lançant VIRTUAL UNIVERSE PRO avec le paramètre /fullscreen sur la ligne de commande. Ceci est nécessaire pour une utilisation de VIRTUAL UNIVERSE PRO avec un écran disposant d'une vue 3D couplé avec une paire de lunettes 3D.

Taille modifiable (mode expert) : Autorise ou non la modification de la taille de fenêtre par l'utilisateur.

Couleur du fond : Couleur du fond de la fenêtre de rendu 3D.

[Lumière ambiante](#) : Détermine la couleur et l'intensité de la lumière ambiante (lumière illuminant l'ensemble des objets quel que soit leurs positions et leurs orientations).

[Montrer les ombres](#) : Si vrai, gère l'affichage des ombres, nécessite que les propriétés des objets relatives aux ombres soit également positionnées. L'affichage des ombres peut réduire de façon significative les performances du rendu.

[Nombre d'images affichées par seconde](#) : fréquence du rafraîchissement graphique

[Utilise le shader \(mode expert\)](#) : Utilisation d'un Shader.

[Nombre maximum d'images par seconde \(mode expert\)](#) : Si différent de 0, impose une limite maximale au nombre d'images par seconde affichées lors du rendu 3D. Permet de préserver le temps processeur.

[Brouillard, couleur du brouillard, début du brouillard, fin du brouillard \(mode expert\)](#) : Affiche un effet de brouillard.

Physique

[Moteur physique \(mode expert\)](#): Détermine le moteur physique utilisé : Newton Physics par défaut.

[Echantillonnage \(0=variable\) \(mode expert\)](#) : Période d'échantillonnage du moteur physique en secondes, 0 indique un échantillonnage variable.

[Simulation soft uniquement \(mode expert\)](#) : Force une simulation physique logicielle pour le moteur physique. N'utilise pas l'accélérateur matériel pour la simulation physique.

[Gravité \(N\) \(mode expert\)](#) : Fixe une valeur à la gravité en Newtons.

[Force souris](#) : Valeur définissant la force d'attraction appliquée par la souris sur les objets lorsque la simulation est active (mode RUN).

IHM

[IHM](#) : Active un pupitre d'interface Homme – Machine.

[Position du pupitre IHM](#) : Détermine la position du pupitre IHM (flottant ou ancré à la fenêtre de rendu).

Ciel (mode expert)

Permet de définir une texture utilisée comme ciel.

Options

[Coefficient de taille globale \(mode expert\)](#) : Modifie les valeurs de position, de taille et de vitesse. Par défaut, le coefficient de taille globale est de 100 pour chacun des axes, ceci correspond à un affichage des unités de mesure en millimètres.

[Unités de mesure longueur et position](#) : Définit les unités utilisées pour les propriétés de longueur et position (par défaut, le millimètre).

Fonctionnalités au niveau Monde



Ajouter

Ajout d'un Sprite 3D, d'une Lumière ou d'une Caméra comme « enfant » du Monde.

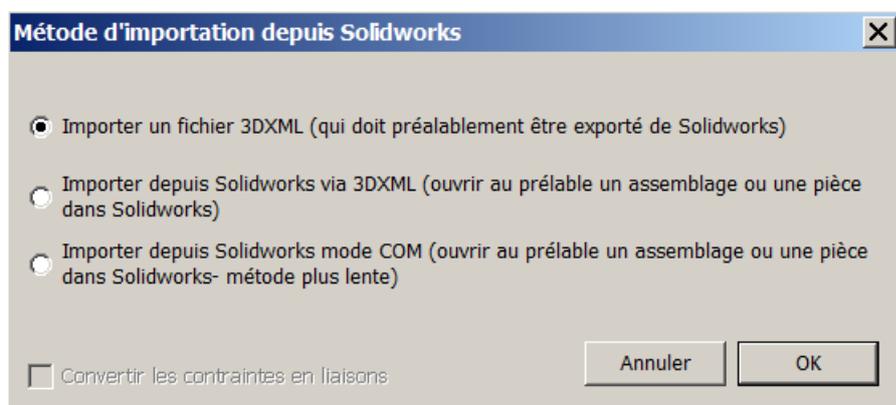
Importer

Un objet : Permet d'importer un objet VIRTUAL UNIVERSE PRO (fichier .VUO) dans le projet de simulation. Un objet VIRTUAL UNIVERSE PRO est une ressource de simulation que l'on souhaite réutiliser dans un projet de simulation. Un objet peut être limité à un simple sprite ou un simple comportement, ou peut représenter une ressource 3D intelligente plus complexe (assemblages de sprites et de comportements). Les ressources 3D intelligentes présentes dans la bibliothèque de démonstration VIRTUAL UNIVERSE PRO sont des objets.

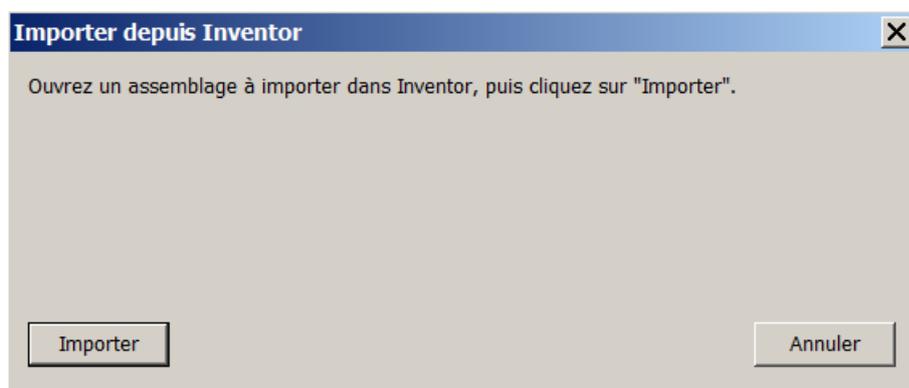
Un fichier 3D : Permet d'importer des fichiers 3D au format standard (fichiers .3DS, VRML, STL, .OBJ, etc..) dans le projet de simulation VIRTUAL UNIVERSE PRO.

Une primitive : Donne accès à la bibliothèque de formes 3D primitives disponibles dans VIRTUAL UNIVERSE PRO. Une primitive est un objet (fichier .vuo) représentant une forme 3D primitive. VIRTUAL UNIVERSE PRO fournit dans sa bibliothèque une liste de formes primitives qu'il est possible de réutiliser dans un projet de simulation. Les fichiers objet primitive sont situés dans le répertoire library/primitive du répertoire d'installation de VIRTUAL UNIVERSE PRO.

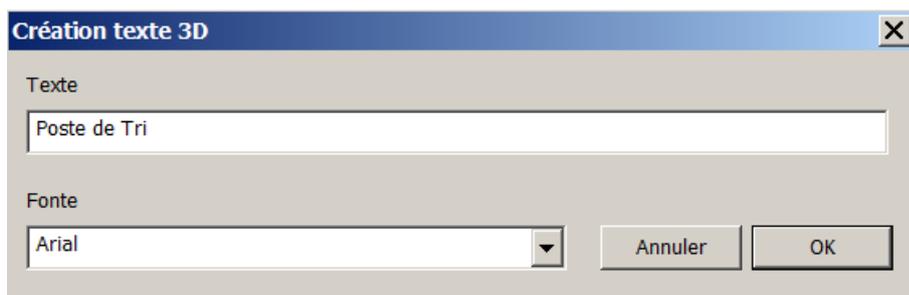
[Depuis SolidWorks](#) : Donne accès à l'outil d'importation des données 3D issus du logiciel de CAO SolidWorks. Pour plus d'information sur la procédure d'importation des données SolidWorks, voir [Importer des modèles 3D issus de SolidWorks](#).



[Depuis Inventor](#) : Donne accès à l'outil d'importation des données 3D issus du logiciel de CAO Inventor. Pour plus d'information sur la procédure d'importation des données Inventor, voir [Importer des modèles 3D issus d'Autodesk Inventor](#).



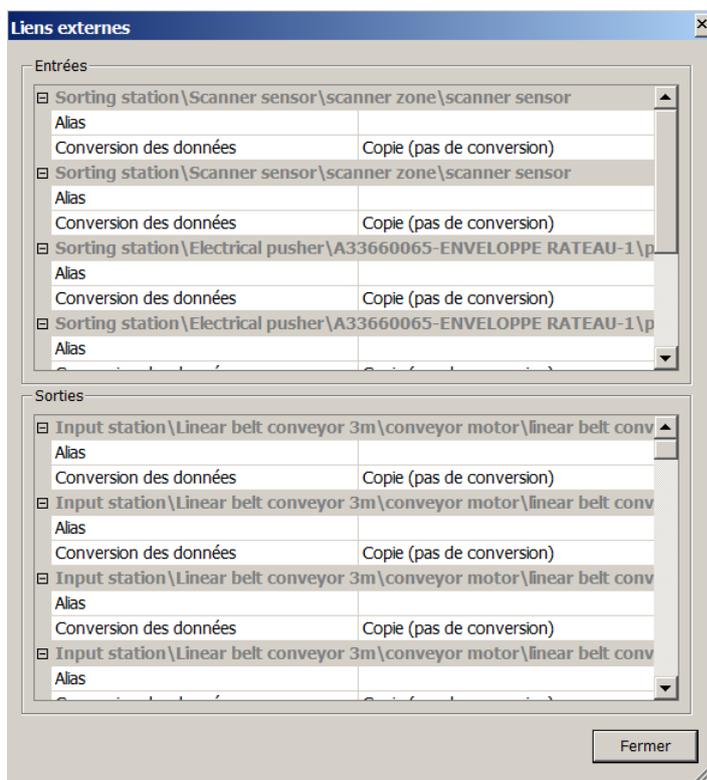
[Un texte 3D](#): Permet d'ajouter un sprite 3D ayant la forme d'un texte.



Poste de Ti

Liens

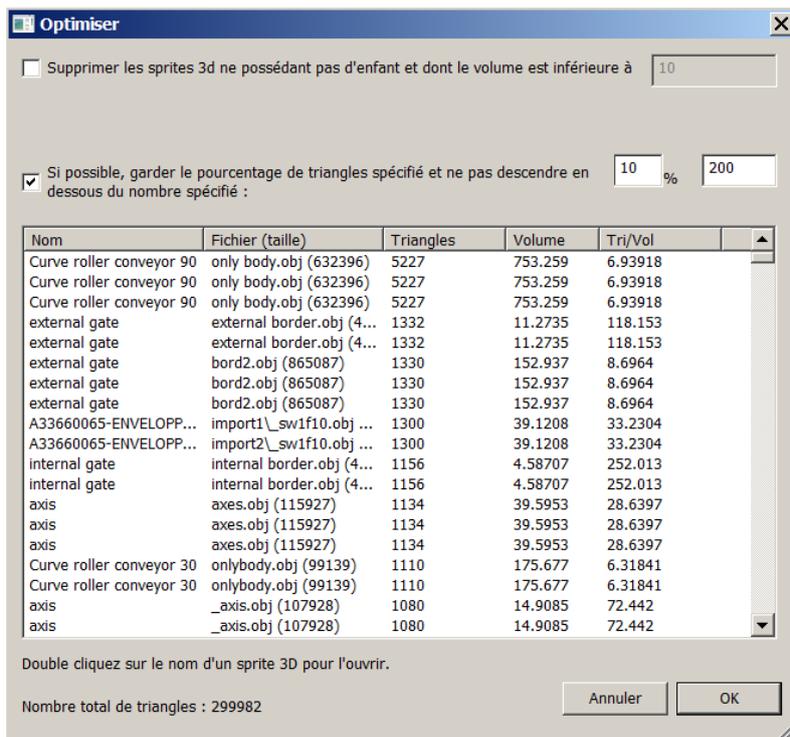
Donne accès à la fenêtre des Liens externes, recensant l'ensemble des comportements déclarés comme « Liens externes » dans le projet de simulation (niveau Monde). Pour plus d'informations concernant les entrées/sorties d'une simulation, voir [Définir la liste des entrées/sorties d'une simulation](#).



Optimiser

Ouvre l'outil d'optimisation des géométries de VIRTUAL UNIVERSE PRO, permettant de simplifier les géométries de tous les sprites 3D du projet d'émulateur.

La simplification des géométries 3D est utilisée pour optimiser les performances du rendu 3D. Pour plus d'informations sur l'utilisation de cet outil d'optimisation, voir [Alléger les modèles CAO 3D](#).



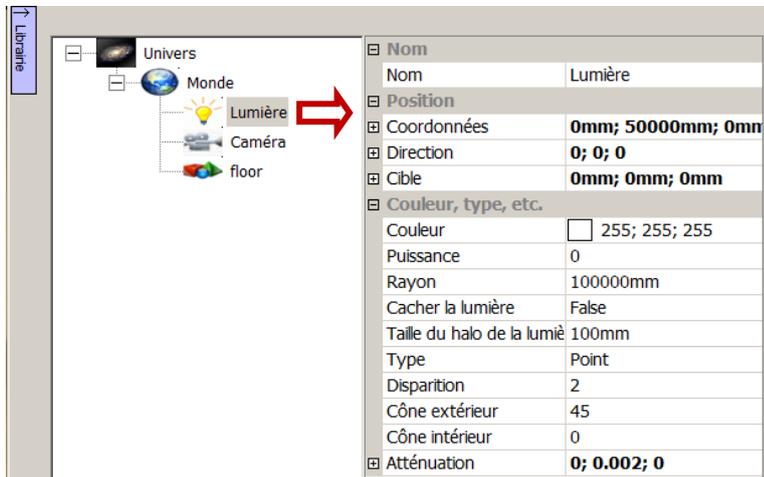
Pour plus d'information sur la mesure et l'optimisation des performances graphiques d'un émulateur3D, voir [Mesurer les performances graphiques](#).

Inclure dans le rendu / Exclure du rendu :

Rend l'ensemble des sprites 3D du monde visibles/invisibles dans le rendu 3D.

Propriétés des Lumières

Propriétés détaillées des Lumières



Propriétés
d'une Lumière

Nom

Permet de désigner une lumière par son nom.

Position

Coordonnées : Permet de définir les coordonnées de la lumière.

Direction : Permet de définir la direction de la Lumière (ce paramètre n'est utilisée que pour les lumières de type Spot ou Directionnel).

Cible (mode expert) : Utilisé uniquement pour les ombres du shader, ce paramètre permet de définir les coordonnées X,Y et Z de la cible.

Couleur, type, etc.

Couleur : Définit la couleur de la Lumière.

Puissance : Facteur multiplicatif de la puissance de la Lumière. Une valeur de 0 correspond à un éclairage par défaut, soit un facteur multiplicatif de 1.

Rayon : Rayon de la Lumière (distance jusqu'à laquelle la Lumière fait effet).

Cacher la Lumière : Cache le halo et le spot.

Taille du halo : Modifie la taille du halo Lumineux mais pas la puissance ni le Rayon de la Lumière.

Type : Type de Lumière : point (éclaire dans toutes les directions), spot ou directionnelle.

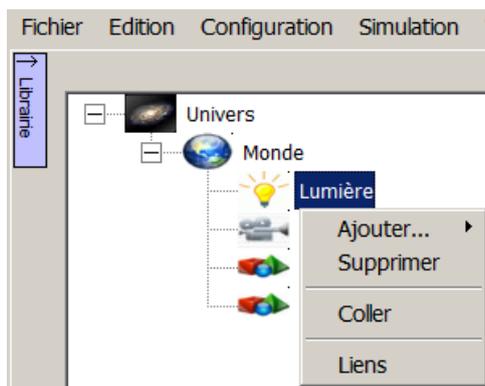
Disparition : Rayon au-delà duquel la lumière disparaît.

Cône extérieur : Angle du cône extérieur pour les spots.

Cône intérieur : Angle du cône intérieur pour les spots.

Atténuation : Baisse d'intensité de l'éclairement en fonction de la distance.

Fonctionnalités au niveau Lumière



Ajouter

Ajout d'un comportement lié à la Lumière.

Supprimer

Supprime définitivement la Lumière.

Coller

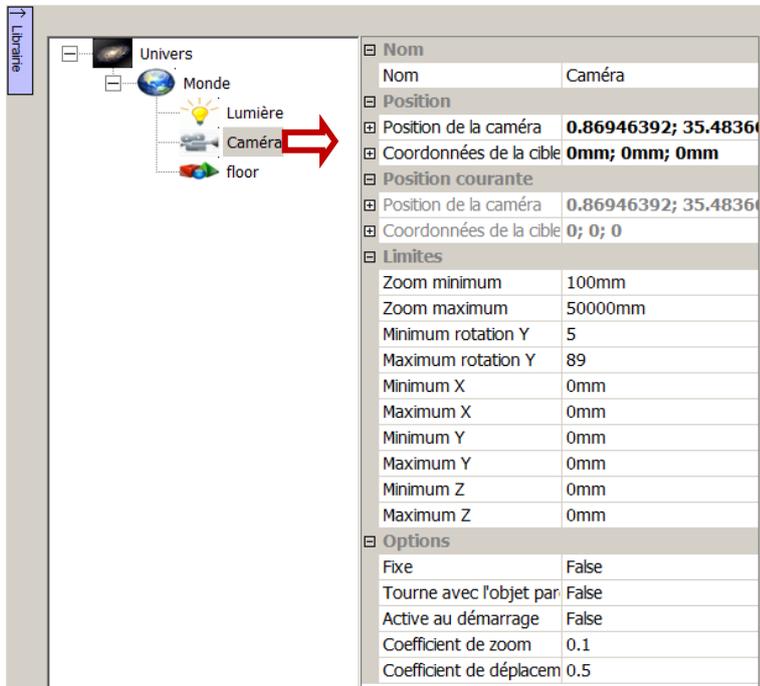
Ajout les comportements copiés depuis une autre lumière.

Liens

Permet d'accéder à la fenêtre de Liens externes de la Lumière. Ceci est utile pour conditionner l'allumage de la Lumière par une variable externe.

Propriétés des Caméras

Propriétés générales des Caméras



Propriétés d'une
Caméra

Nom

Permet de désigner une caméra par son nom.

Position

Détermine la position initiale de la Caméra par les coordonnées de la cible (ce que regarde la Caméra) ainsi qu'une rotation sur les axes X et Y et un zoom.

Position courante

Idem mais pour la position courante. La position courante peut être recopiée dans la position initiale en cliquant sur les flèches descendantes apparaissant à droite des éléments de positions initiales et en choisissant « Copier depuis les valeurs courantes ».

Limites

Permet de limiter les mouvements de la caméra.

Options

Fixe : L'utilisateur ne peut pas déplacer la caméra.

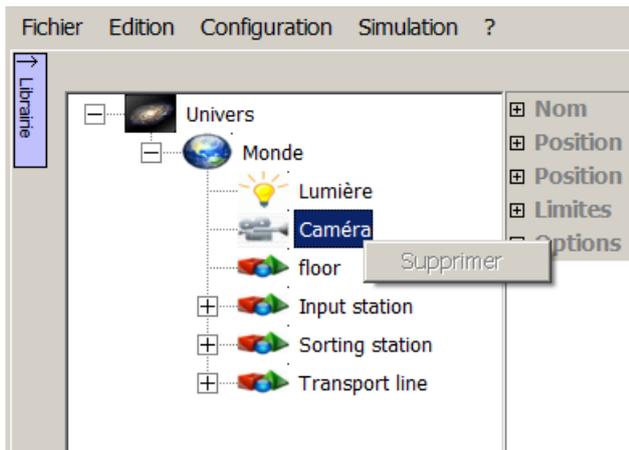
Tourne avec l'objet parent : Si vrai, la caméra tourne en même temps que son parent.

Active au démarrage : En mode multi-caméras, définit la caméra comme active.

Coefficient de zoom : Augmente ou diminue la vitesse du Zoom (0 = valeur par défaut).

Coefficient de déplacement : Augmente ou diminue la vitesse de déplacement (0 = valeur par défaut).

Fonctionnalités au niveau Caméra

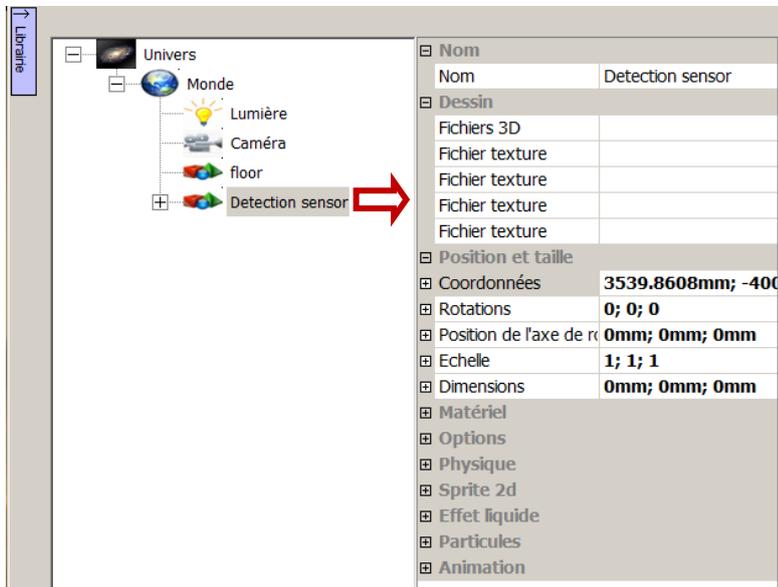


Supprimer

Suppression de la Caméra.

Propriétés des Sprites

Propriétés détaillées des sprites



Propriétés
d'un Sprite

Nom

Permet de désigner un Sprite 3D par son nom.

Dessin

Détermine le fichier 3D utilisé pour définir la géométrie du Sprite 3D ainsi que d'éventuels fichiers de texture.

Position et taille

Définit la position, la rotation (ainsi que l'axe) et l'échelle (mode expert) initiale du Sprite 3D. Les rotations sont exprimées en degrés (de -180 à +180 degrés).

Position et taille (valeurs courantes) – mode RUN

Idem mais pour les valeurs courantes, avec en plus : la translation et la rotation relative (par rapport au Sprite 3D parent), ainsi que la position du centre de l'objet et la rotation absolue par rapport au Monde.

Matériel

Ces propriétés regroupent les caractéristiques du matériel utilisé pour afficher l'objet. Ces caractéristiques sont directement liées au moteur de rendu Irrlicht.

Transparence : Définit la transparence de l'objet depuis 0 (aucune) jusqu'à 1 (totalement transparent).

[Dessine les deux faces \(mode expert\)](#) : Permet de dessiner les deux faces des surfaces de l'objet.

Matériel (valeurs courantes) – mode RUN

Idem pour les valeurs courantes.

Options

[Non sélectionnable](#) : Si vrai, le Sprite 3D n'influe pas sur la navigation lorsqu'il est survolé par le curseur de la souris.

[Sélectionne le parent](#) : Si vrai, sélectionne le parent à la place de l'objet si cliqué.

[Magnétique](#) : Objet magnétique pour l'aide au positionnement et à la connexion automatique des sprites 3D.

Physique

[Utilise la physique](#) : Si vrai, le Sprite 3D sera pris en compte par le moteur physique, dans le cas contraire, l'objet sera complètement ignoré par le moteur physique. En d'autres termes, l'objet sera uniquement affiché dans le monde 3D mais n'aura aucune interaction physique avec les autres objets.

[Utilise la gravité](#) : Si vrai, le Sprite 3D sera soumis à la gravité. Sa masse devra également être non nulle.

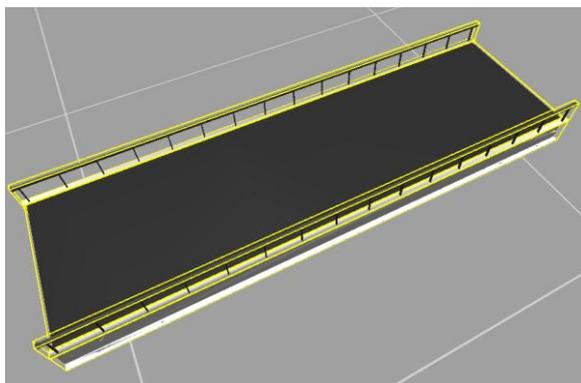
[L'utilisateur peut appliquer une force à l'objet](#) : Si vrai, l'utilisateur peut, en mode RUN, agir sur l'objet en maintenant le bouton droit de la souris enfoncé lorsque le curseur est sur le Sprite 3D et en déplaçant le curseur.

[Type de corps](#) : Détermine pour le moteur physique le type de géométrie du Sprite 3D :

- Quelconque (convexe) : une forme convexe déduite de la géométrie du Sprite 3D,
- Quelconque : une forme quelconque (disponible si le moteur physique est Physx),
- Boîte : un parallélépipède rectangle,
- Sphère : une sphère,
- Capsule : une capsule,
- Tissu : un tissu (moteur NVidia Physx seulement),
- Objet mou : un objet mou (moteur NVidia Physx seulement),
- Fluide : un fluide (moteur Nivida Physx seulement).

Attention ! Le type « Quelconque », lorsqu'il est utilisé avec un Sprite 3D complexe (possédant de très nombreuses faces), peut consommer beaucoup de ressources pour la simulation physique. Privilégiez, lorsque ceci est possible un des autres types (comme le type Boîte).

Il est possible et souvent utile dans les phases de mise au point de visualiser les géométries manipulées par le moteur physique en activant l'option « Mode debug pour le moteur physique » dans les propriétés de l'Univers.



Masse : La masse de l'objet. Une masse de 0 fige l'objet.

Moment d'inertie (mode expert): Détermine la quantité d'énergie nécessaire pour faire tourner l'objet sur chacun des axes.

Amortissement linéaire/angulaire (mode expert) : Friction visqueuse linéaire/angulaire.

Ajuster automatiquement le centre de masse (mode expert): Si vrai, le centre de masse de l'objet est automatiquement recalculé en fonction de la géométrie du Sprite 3D. Sinon, le centre de masse est le point de coordonnées 0/0/0 du Sprite 3D.

Centre de gravité (mode expert) : Définit le centre de gravité de l'objet.

Coefficients : Déterminent la friction, l'élasticité et la souplesse des objets. Une valeur de 0 utilise les paramètres par défaut du moteur physique. Le coefficient utilisé par le moteur physique entre un objet A et un objet B est une combinaison (le produit) des coefficients de l'objet A et de l'objet B.

Pénétration : Si vrai, les collisions de l'objet ne sont pas gérées. Dans le cas d'objets liés par des joints (voir ci-après) les collisions sont automatiquement désactivées entre deux objets liés par un joint.

Joint physique avec le parent

Joint (liaison) : Détermine le type de la liaison avec l'objet parent :

- Pivot,
- Glissière,
- Rotule,
- Pivot glissant,
- Poulie (disponible avec le moteur NVidia Physx uniquement).
- Fixe.

Position du pivot : Détermine la position x/y/z de la liaison avec l'objet parent pour les liaisons pivot.

Ligne d'action : Détermine la ligne d'action du joint pour les joints Glissière (axe de translation) et Pivot (axe de rotation).

Limites : Détermine les limites minimales et maximales du joint. Si ces deux valeurs sont égales, alors le joint n'a pas de limite (rotation ou translation sans borne).

Puissance du joint : Détermine la rigidité du joint.

Force du joint (mode expert) : Donne la valeur de la force supportée par le joint.

Force de cassure du joint (mode expert) : Force au-delà de laquelle le joint sera automatiquement détruit. Si 0, cette fonction est désactivée. Ceci permet de simuler la destruction d'une liaison entre deux objets.

Position du pivot ou point d'ancrage (objet parent ou autre objet) : Origine du pivot ou point d'ancrage.

Poulies (mode expert) : Options relatives à la création d'une poulie.

Sprite 2d (mode expert)

Fichier graphique 2D affiché dans le monde 3D.

Effet liquide (mode expert)

Utilisé pour simuler un effet de liquide.

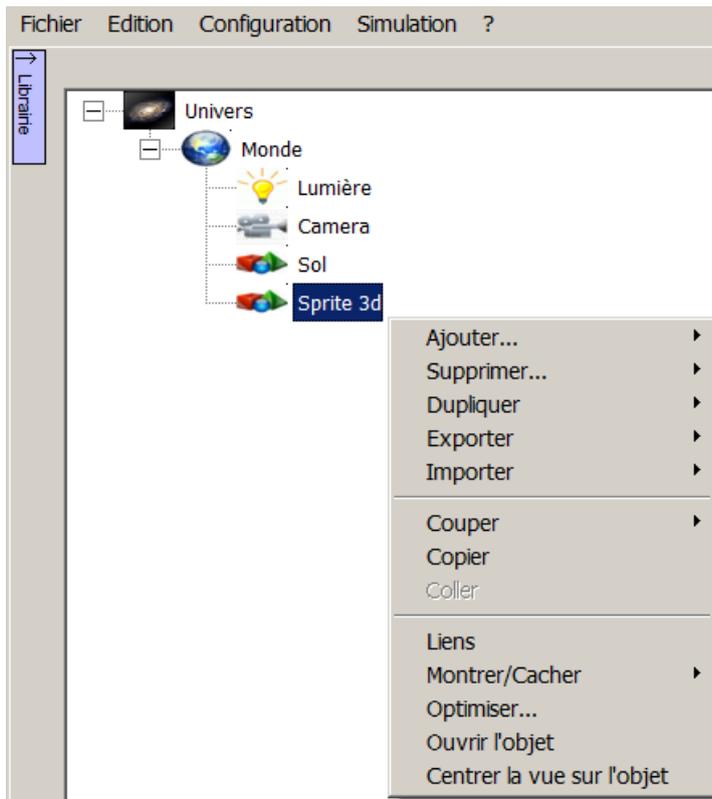
Particules (mode expert)

Affichage de particules (simulation de petits objets se déplaçant : simulation de liquide, feu...).

Animation (mode expert)

Gère l'animation incluse dans les fichiers 3D (seulement disponible pour les fichiers .x).

Fonctionnalités au niveau Sprite



Ajouter

Ajout d'un Sprite 3D, d'une Lumière, d'une Caméra ou d'un comportement comme enfant du Sprite 3D.

Supprimer

Suppression du Sprite 3D. Possibilité de supprimer un groupe de Sprites 3D avec « Ce sprite 3D et le nombre de sprites 3D suivant ».

Dupliquer

Duplication en ligne ou sous forme de chaînes de ressources 3D. Associée à la fonction « Magnétique », cette option est particulièrement utile pour la construction d'une ligne de convoyage.

Exporter

Permet d'exporter un Sprite (avec ses éventuels enfants et comportements) sous la forme d'un objet VIRTUAL UNIVERSE PRO réutilisable (fichier .VUO), comme une ressource 3D intelligente. Par défaut, VIRTUAL UNIVERSE PRO propose d'enregistrer cet objet dans le répertoire de la bibliothèque VIRTUAL UNIVERSE PRO (library), mais il est possible de sélectionner un autre emplacement sur l'ordinateur.

Il est également possible d'exporter uniquement les comportements ou les enfants du sprite sélectionné.

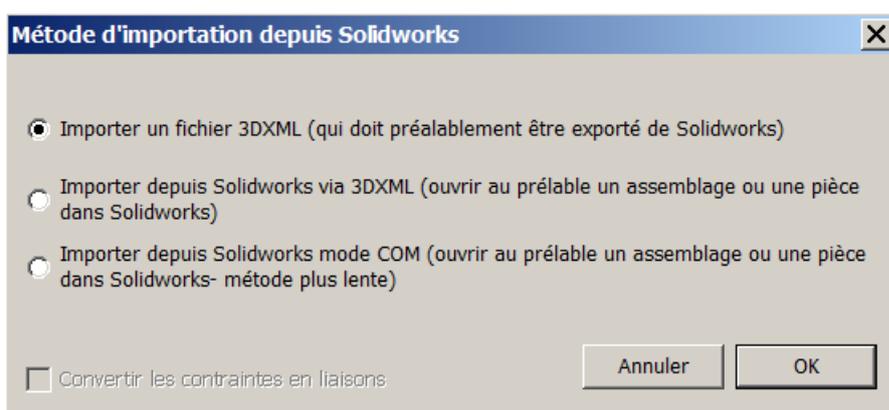
Importer

Un objet : Permet d'importer un objet VIRTUAL UNIVERSE PRO (fichier .VUO) dans le projet de simulation. Un objet VIRTUAL UNIVERSE PRO est une ressource de simulation que l'on souhaite réutiliser dans un projet de simulation. Un objet peut être limité à un simple sprite ou un simple comportement, ou peut représenter une ressource 3D intelligente plus complexe (assemblages de sprites et de comportements). Les ressources 3D intelligentes présentes dans la bibliothèque de démonstration VIRTUAL UNIVERSE PRO sont des objets.

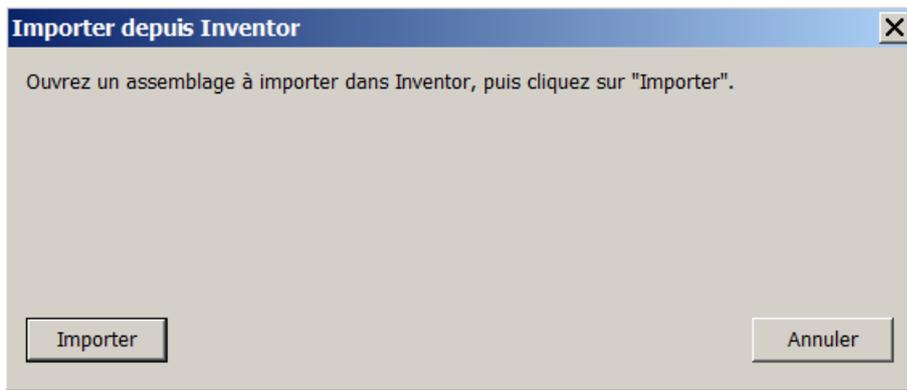
Un fichier 3D : Permet d'importer des fichiers 3D au format standard (fichiers .3DS, VRML, STL, .OBJ, etc..) dans le projet de simulation VIRTUAL UNIVERSE PRO.

Une primitive : Donne accès à la bibliothèque de formes 3D primitives disponibles dans VIRTUAL UNIVERSE PRO. Une primitive est un objet (fichier .vuo) représentant une forme 3D primitive. VIRTUAL UNIVERSE PRO fournit dans sa bibliothèque une liste de formes primitives qu'il est possible de réutiliser dans un projet de simulation. Les fichiers objet primitive sont situés dans le répertoire library/primitive du répertoire d'installation de VIRTUAL UNIVERSE PRO.

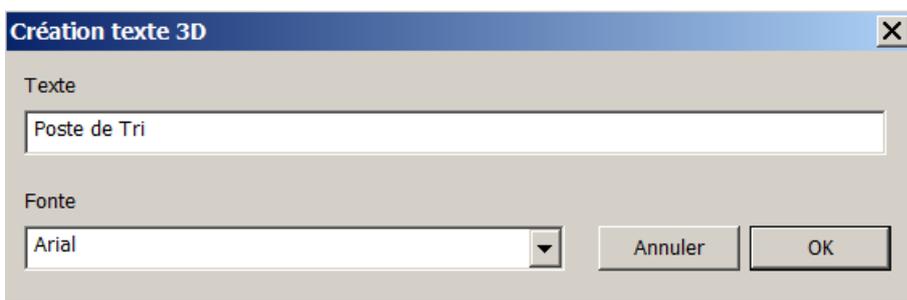
Depuis SolidWorks : Donne accès à l'outil d'importation des données 3D issus du logiciel de CAO SolidWorks. Pour plus d'information sur la procédure d'importation des données SolidWorks, voir [Importer des modèles 3D issus de SolidWorks](#).



Depuis Inventor: Donne accès à l'outil d'importation des données 3D issus du logiciel de CAO Inventor. Pour plus d'information sur la procédure d'importation des données Inventor, voir [Importer des modèles 3D issus d'Autodesk Inventor](#).

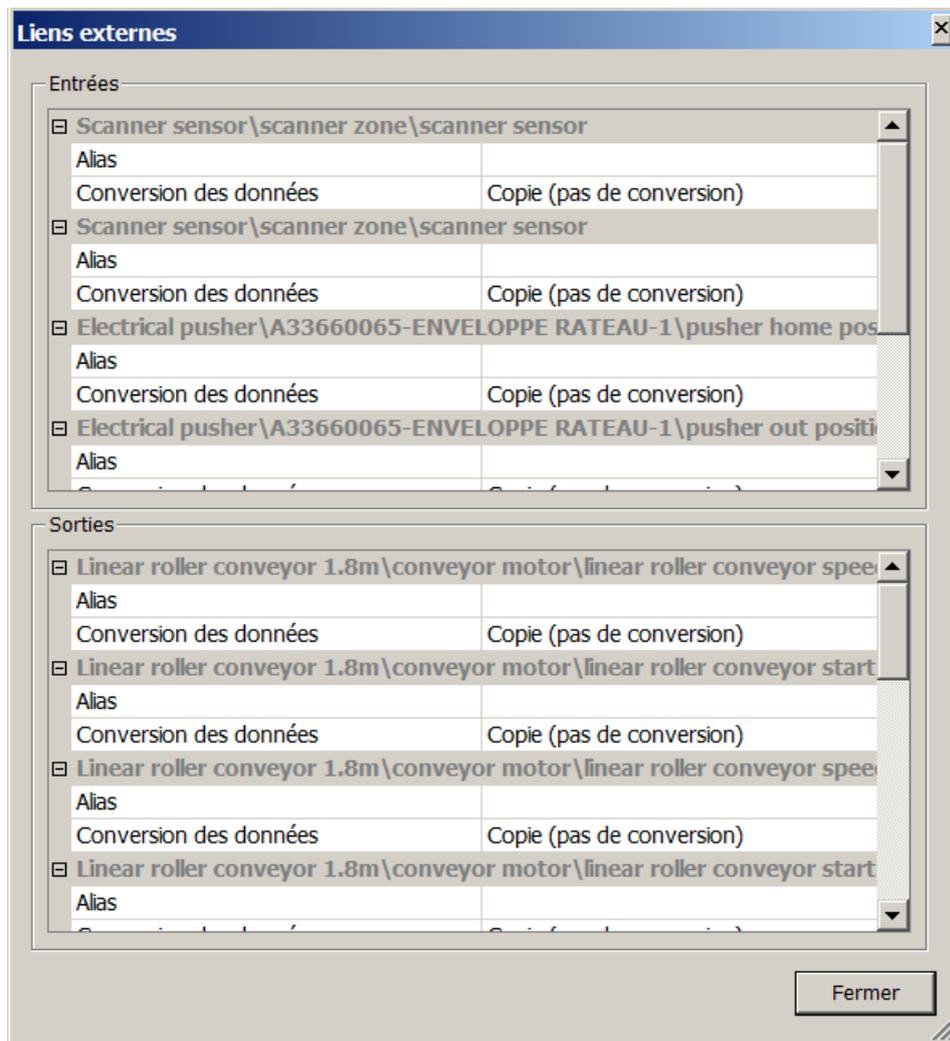


Un texte 3D: Permet d'ajouter un sprite 3D ayant la forme d'un texte.



Liens

Donne accès à la fenêtre des Liens externes, recensant l'ensemble des comportements déclarés comme « Liens externes » au niveau du sprite sélectionné.



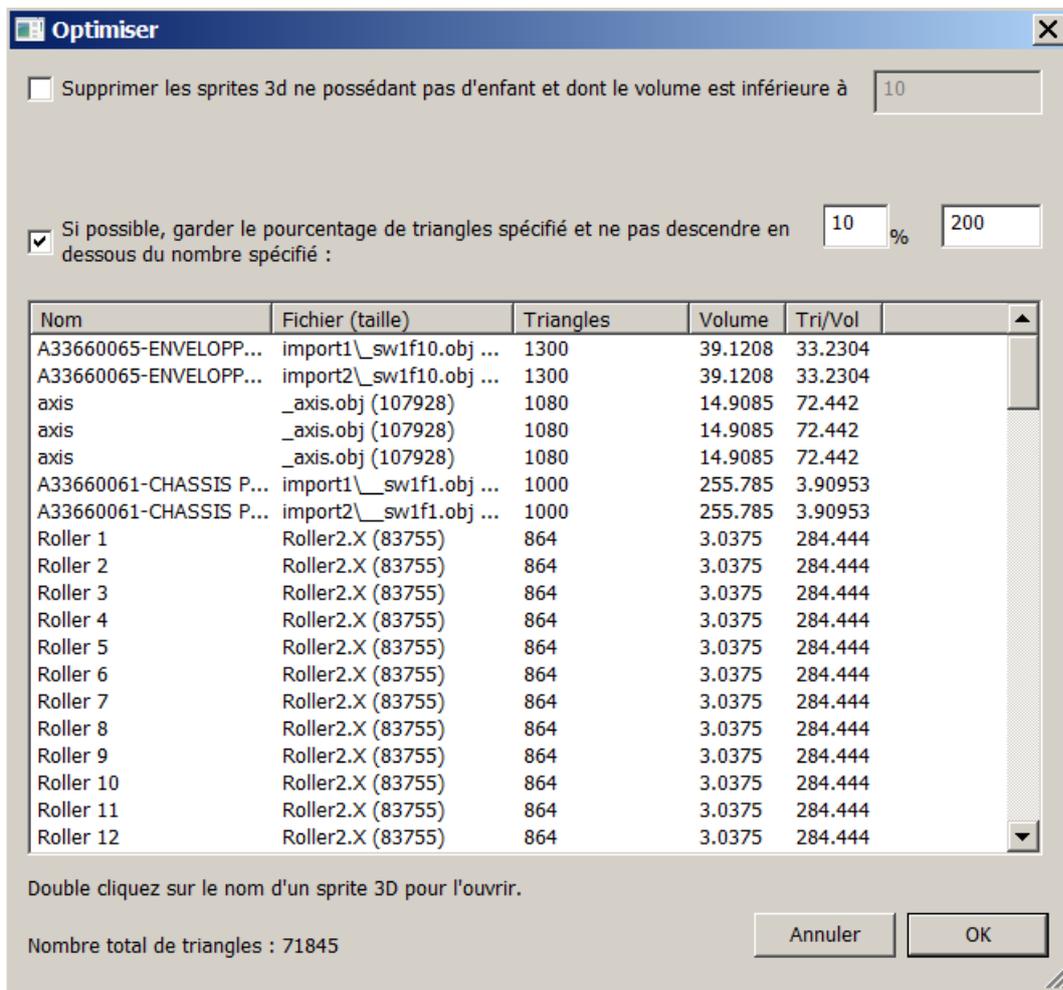
Montrer/Cacher

Permet de cacher, de montrer, d'exclure ou d'inclure dans le rendu une ressource 3D.

Optimiser

Ouvre l'outil d'optimisation des géométries de VIRTUAL UNIVERSE PRO, permettant de simplifier les géométries du sprite 3D sélectionné (incluant ses sprites enfants).

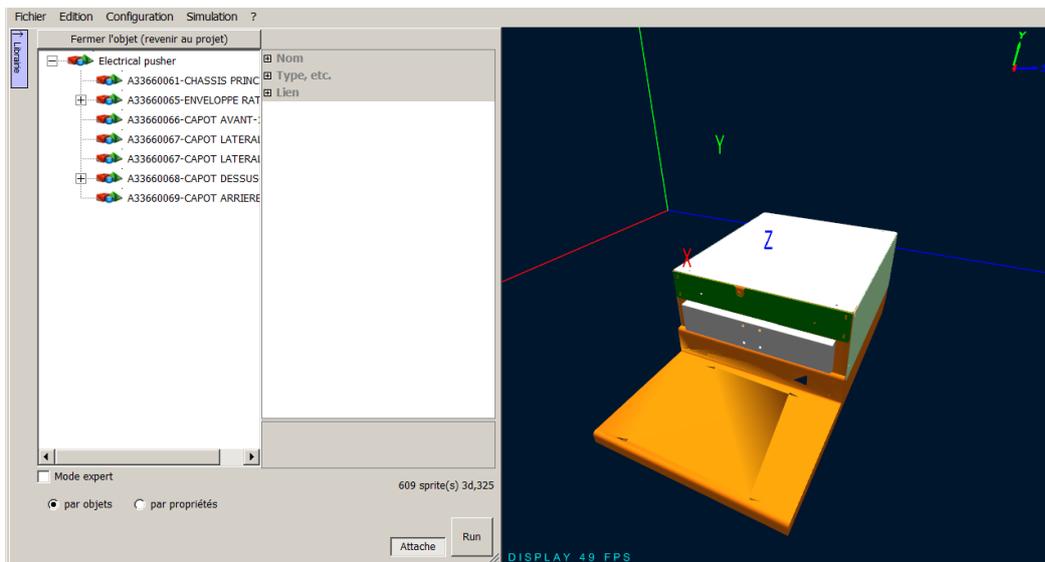
Pour plus d'informations sur l'utilisation de cet outil d'optimisation, voir [Alléger les modèles CAO 3D](#)



Pour plus d'information sur la mesure et l'optimisation des performances graphiques d'une simulation, voir [Mesurer les performances graphiques](#)

Ouvrir l'objet

Permet d'isoler et de visualiser uniquement la ressource sélectionnée dans le rendu 3D et dans l'arborescence du projet.

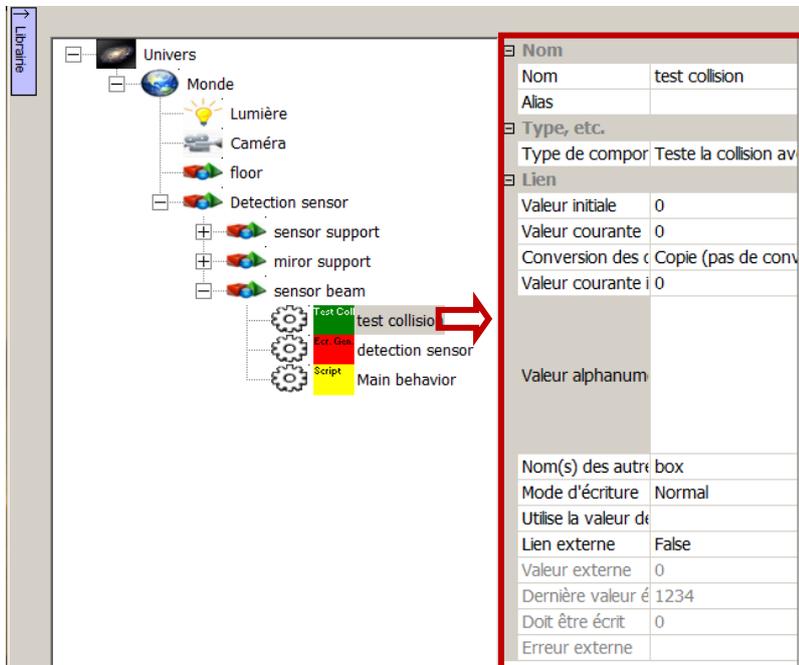


Centrer la vue sur l'objet

Centre la vue sur le sprite 3D sélectionné.

Propriétés des Comportements

Propriétés détaillées des comportements



Propriétés d'un
Comportement

Nom

Nom du comportement.

Type, etc.

Type du comportement (voir détails plus loin).

Lien

Valeur initiale : Sera recopiée dans la valeur courante au passage en mode RUN de la simulation. Peut être utilisé pour activer de façon permanente un Comportement. Par exemple, un script pourra être exécuté de façon inconditionnelle dès le lancement de la simulation en mettant cette propriété à 1.

Valeur courante : Valeur courante de la variable du comportement, cette valeur peut être copiée depuis ou copiée vers le logiciel externe.

Conversion des données : Détermine la façon dont les données sont converties entre la valeur courante et la valeur courante interne :

- Copie (pas de conversion),
- Booléen (faux si 0, vrai autrement),
- Non booléen (vrai si 0, faux autrement),
- Depuis le monde réel,
- Vers le monde réel.

Valeur courante interne : Valeur courante interne du comportement, cette valeur est utilisée pour piloter la simulation ou cette valeur provient de la simulation.

Valeur alphanumérique (mode expert) : Permet de stocker une chaîne de caractères.

Mode d'écriture : Détermine le mode d'écriture des données vers le logiciel externe :

- Normal : écriture des données à chaque cycle d'échange.
- Seulement sur changement : écriture des données seulement sur changement de valeur.
- Sécurisé : lecture des données depuis le logiciel externe et écriture des données que si elles sont différentes. Quand les données seront écrites, elles seront relues pour vérifier que l'écriture a fonctionné.

Utilise la valeur de ce comportement : Si non vide, cette zone donne le nom d'un Comportement dont la valeur sera lue et recopiée dans la valeur interne courante.

Lien externe : Si vrai, le comportement sera listé dans la liste des liens externes.

Valeur externe (mode expert) : Utilisé par le mode « Seulement sur changement ».

Dernière valeur écrite vers le logiciel externe (mode expert) : Utilisé par le mode sécurisé.

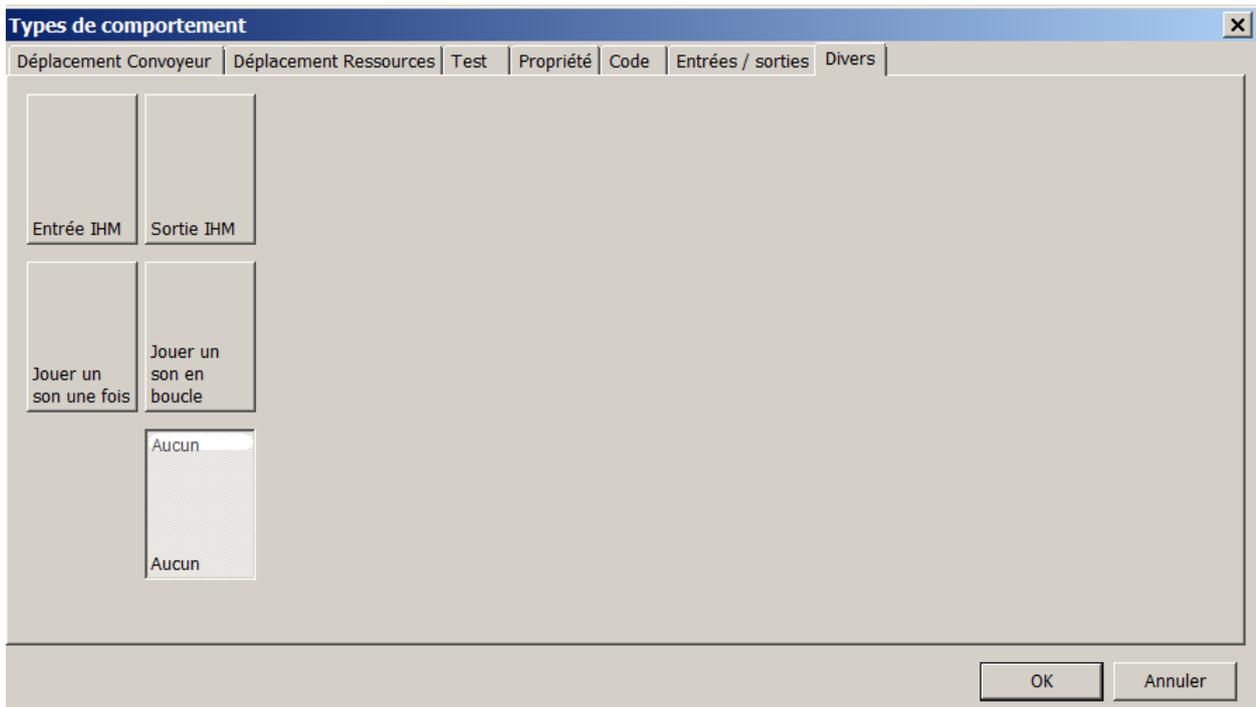
Doit être écrit (mode expert) : Utilisé par le mode sécurisé.

Erreur externe (mode expert) : Message d'erreur si une erreur s'est produite pendant l'accès à une variable externe. Vide si pas d'erreur.

Types de comportements

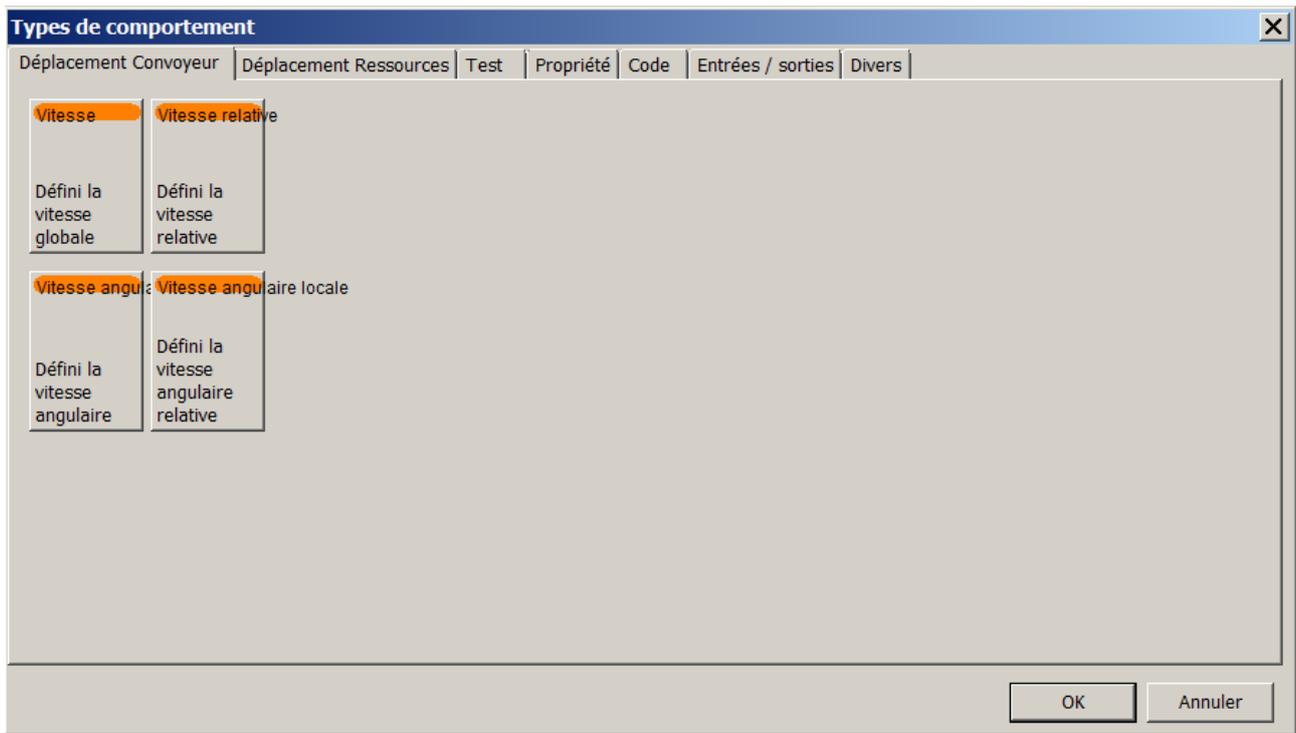
VIRTUAL UNIVERSE PRO propose une grande variété de comportements à appliquer aux sprites 3D.

Ces comportements sont classés en 7 catégories présentées dans la fenêtre « Types de comportement ».



Déplacement Convoyeur

Ces comportements sont destinés à asservir la vitesse de déplacement du tapis d'un convoyeur (en translation ou en rotation, selon la forme du convoyeur).



Vitesse

Vitesse :

- Définit la vitesse linéaire du sprite relativement aux axes du repère du Monde (repère absolu). Exprimée en mètre/s ou millimètre/s selon les unités choisies dans les propriétés du monde.

Important ! Ce comportement exploite le moteur physique de VIRTUAL UNIVERSE PRO. Il nécessite que l'option Physique soit activée pour le sprite et qu'une liaison glissière soit définie entre ce sprite (sprite déplacé) et le sprite parent.



Vitesse relative

Vitesse relative :

- Définit la vitesse linéaire du sprite relativement aux axes du repère local au sprite parent (repère local). Exprimée en mètre/s ou millimètre/s selon les unités choisies dans les propriétés du monde.

Important ! Ce comportement exploite le moteur physique de VIRTUAL UNIVERSE PRO. Il nécessite que l'option Physique soit activée pour le sprite et qu'une liaison glissière soit définie entre ce sprite (sprite déplacé) et le sprite parent.

Exemple

Ce comportement est utilisé dans les convoyeurs de type « Linear belt conveyor » et « Linear roller conveyor » présents en bibliothèque de démo VIRTUAL UNIVERSE PRO, pour gérer la vitesse de déplacement du tapis du convoyeur.



Vitesse angulaire :

- Définit la vitesse angulaire du sprite relativement aux axes du repère du Monde (repère absolu). Exprimée en degré/s.

Important ! Ce comportement exploite le moteur physique de VIRTUAL UNIVERSE PRO. Il nécessite que l'option Physique soit activée pour le sprite et qu'une liaison pivot soit définie entre ce sprite (sprite déplacé) et le sprite parent.



Vitesse angulaire relative :

- Définit la vitesse angulaire du sprite relativement aux axes du repère local au sprite parent (repère local). Exprimée en degré/s.

Important ! Ce comportement exploite le moteur physique de VIRTUAL UNIVERSE PRO. Il nécessite que l'option Physique soit activée pour le sprite et qu'une liaison pivot soit définie entre ce sprite (sprite déplacé) et le sprite parent.

Exemple

Ce comportement est utilisé dans les convoyeurs de type « Curve roller conveyor » présents en bibliothèque de démo VIRTUAL UNIVERSE PRO, pour gérer la vitesse de déplacement du tapis du convoyeur.

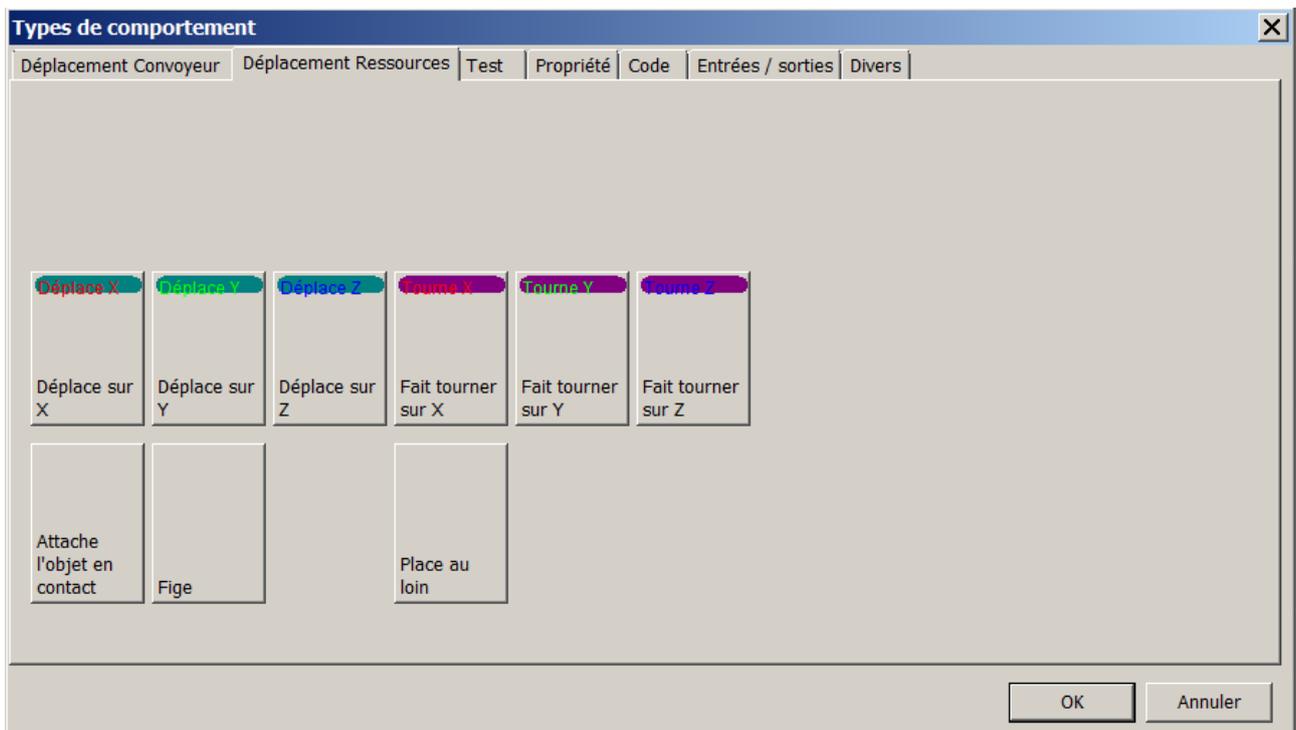
Déplacement Ressources

Ces comportements sont destinés à commander le déplacement des ressources 3D et objets 3D (translation, rotation, position).

Ce sont ces comportements qui sont le plus largement utilisés pour gérer le déplacement des ressources 3D dans les émulateurs 3D VIRTUAL UNIVERSE PRO.

La plupart de ces comportements peuvent être créés et paramétrés automatiquement à l'aide du Motion Assistant. Pour plus de détails, voir [Définir des profils de mouvement avec le Motion Assistant](#).

Ces comportements n'exploitent pas le moteur physique de VIRTUAL UNIVERSE PRO et ne nécessitent pas que des liaisons soient définies entre les sprites enfant et parent.



Déplace sur X:

- Déplace le sprite en translation relativement à l'axe X du repère local au sprite parent (repère local).

La propriété « Mode pour déplacement et rotation » permet de choisir le mode de pilotage pour ce déplacement (Temps, Vitesse, Position).

La vitesse est exprimée en mètre/s ou millimètre/s selon les unités choisies dans les propriétés du monde.

Important ! Ce comportement n'exploite pas le moteur physique de VIRTUAL UNIVERSE PRO. Toutefois, l'option Physique peut être activée sur le sprite, pour prendre en compte les collisions potentielles de ce sprite avec les autres sprites, lors de son déplacement.

Exemple

Le comportement « Déplace sur X » en mode de pilotage « Temps » est utilisé dans le poussoir électrique « Electrical pusher » présent en bibliothèque, pour piloter la rentrée/sortie du tiroir.



Déplace sur Y:

- Déplace le sprite en translation relativement à l'axe Y du repère local au sprite parent (repère local).

La propriété « Mode pour déplacement et rotation » permet de choisir le mode de pilotage pour ce déplacement (Temps, Vitesse, Position).

La vitesse est exprimée en mètre/s ou millimètre/s selon les unités choisies dans les propriétés du monde.

Important ! Ce comportement n'exploite pas le moteur physique de VIRTUAL UNIVERSE PRO. Toutefois, l'option Physique peut être activée sur le sprite, pour prendre en compte les collisions potentielles de ce sprite avec les autres sprites, lors de son déplacement.



Déplace sur Z:

- Déplace le sprite en translation relativement à l'axe Z du repère local au sprite parent (repère local).

La propriété « Mode pour déplacement et rotation » permet de choisir le mode de pilotage pour ce déplacement (Temps, Vitesse, Position).

La vitesse est exprimée en mètre/s ou millimètre/s selon les unités choisies dans les propriétés du monde.

Important ! Ce comportement n'exploite pas le moteur physique de VIRTUAL UNIVERSE PRO. Toutefois, l'option Physique peut être activée sur le sprite, pour prendre en compte les collisions potentielles de ce sprite avec les autres sprites, lors de son déplacement.



Fait tourner sur X:

- Déplace le sprite en rotation relativement à l'axe X du repère local au sprite parent (repère local).

La propriété « Mode pour déplacement et rotation » permet de choisir le mode de pilotage pour ce déplacement (Temps, Vitesse, Position).

La vitesse est exprimée en degré/s.

Important ! Ce comportement n'exploite pas directement le moteur physique de VIRTUAL UNIVERSE PRO. Toutefois, l'option Physique peut être activée sur le sprite, pour prendre en compte les collisions potentielles de ce sprite avec les autres sprites, lors de son déplacement.



Fait tourner sur Y:

- Déplace le sprite en rotation relativement à l'axe Y du repère local au sprite parent (repère local).

La propriété « Mode pour déplacement et rotation » permet de choisir le mode de pilotage pour ce déplacement (Temps, Vitesse, Position).

La vitesse est exprimée en degré/s.

Important ! Ce comportement n'exploite pas le moteur physique de VIRTUAL UNIVERSE PRO. Toutefois, l'option Physique peut être activée sur le sprite, pour prendre en compte les collisions potentielles de ce sprite avec les autres sprites, lors de son déplacement.



Fait tourner sur Z:

- Déplace le sprite en rotation relativement à l'axe Z du repère local au sprite parent (repère local).

La propriété « Mode pour déplacement et rotation » permet de choisir le mode de pilotage pour ce déplacement (Temps, Vitesse, Position).

La vitesse est exprimée en degré/s.

Important ! Ce comportement n'exploite pas le moteur physique de VIRTUAL UNIVERSE PRO. Toutefois, l'option Physique peut être activée sur le sprite, pour prendre en compte les collisions potentielles de ce sprite avec les autres sprites, lors de son déplacement.



Attache l'objet en contact :

- Attache le sprite 3D sélectionné avec d'autres sprites dont le nom est renseigné dans le champ Lien/Nom(s).

Important ! Ce comportement doit toujours être appliqué au sprite en déplacement, venant au contact avec les autres sprites. Le nom des sprites à attacher doit être renseigné.



Fige :

- Immobilise l'objet.

Important ! Ce comportement exploite le moteur physique de VIRTUAL UNIVERSE PRO. Il nécessite que l'option Physique soit activée pour le sprite.

Exemple

Ce comportement est utilisé dans les convoyeurs présents en bibliothèque, pour figer le tapis du convoyeur et le ramener à sa position initiale, lorsqu'il est soumis à un déplacement.

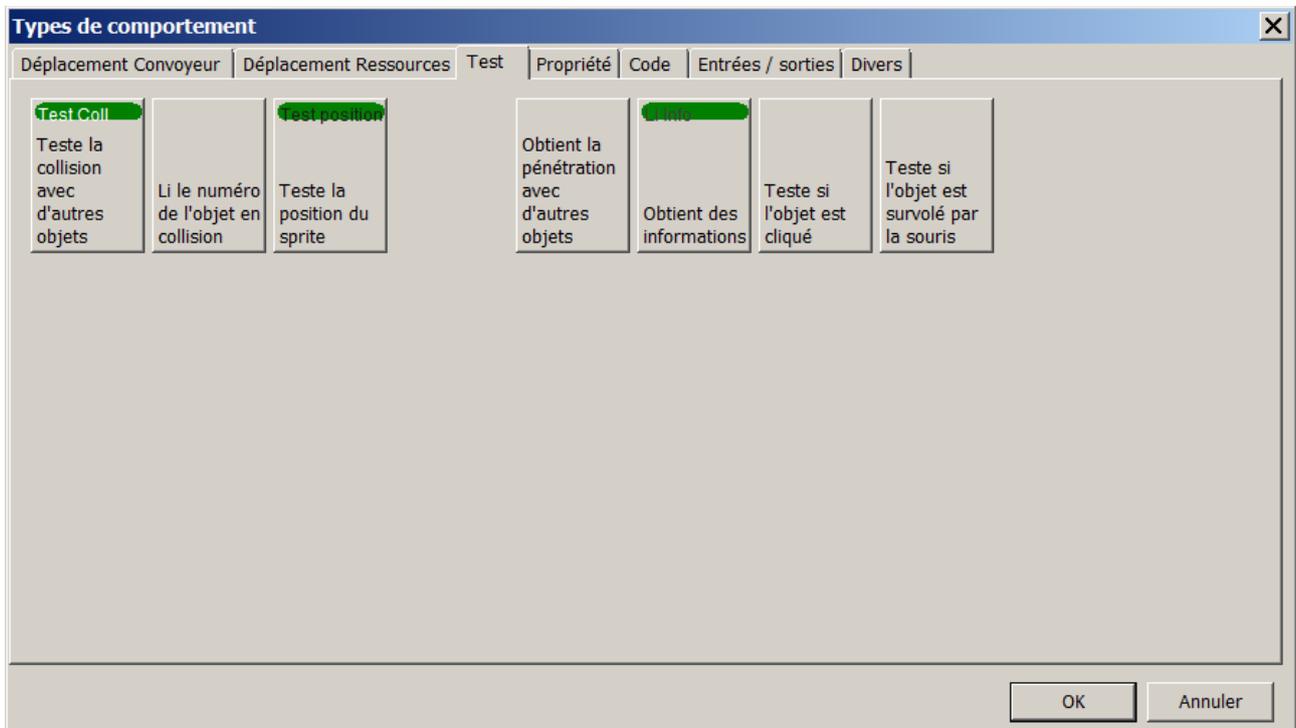


Place au loin :

- Place l'objet 3D au loin en le déplaçant à une position lointaine (non visible).

Test

Ces comportements sont destinés à récupérer des informations sur les objets 3D en cours de simulation (position ou identifiant d'un objet, détection d'une pièce par un capteur, détection des collisions...)



[Teste la collision avec d'autres objets :](#)

- Permet de détecter la collision du sprite avec d'autres sprites 3D du monde. La valeur courante interne donne le nombre de triangles en collision.

Important ! Le paramètre « Nom(s) des autres sprites 3D » permet de limiter l'action de ce comportement à un groupe de sprites 3D. Le nom des sprites à surveiller doit être renseigné.



[Lis le numéro de l'objet en collision :](#)

- Permet d'obtenir l'identifiant du sprite 3D en collision avec le sprite.

Exemple

Le capteur scanner présent dans la bibliothèque de démo VIRTUAL UNIVERSE PRO exploite ce comportement.



[Teste la position du Sprite :](#)

- Permet de tester si la position du sprite est bien comprise entre deux bornes « Position min » et « Position max ». Ce comportement permet de modéliser un capteur de position situé sur un actionneur.

Important ! Ce comportement exploite le moteur physique de VIRTUAL UNIVERSE PRO. Il nécessite que l'option Physique soit activée pour le sprite et qu'une liaison soit définie entre le sprite et le sprite parent.



Obtient la pénétration avec d'autres objets :

- Donne la profondeur de pénétration entre le sprite associé au Comportement et les autres Sprites 3D. Le paramètre « Nom(s) des autres Sprites 3D » permet de limiter l'action de ce Comportement à un groupe de Sprites 3D.

Important ! Ce comportement exploite le moteur physique de VIRTUAL UNIVERSE PRO. Il nécessite que l'option Physique soit activée pour le sprite. Une utilisation fréquente de ce comportement peut diminuer les performances du moteur physique.

Exemple

Ce comportement permet de modéliser un capteur de proximité.



Obtient des informations :

- Permet d'accéder aux valeurs dynamiques d'un Sprite 3D. Le paramètre « Sélectionne l'information à lire depuis le Sprite 3D » détermine l'information reçue.



Teste si l'objet est cliqué :

- Prend la valeur 1 si l'utilisateur clique sur l'objet associé (avec le bouton gauche de la souris).

Important ! Ce comportement ne fonctionne que si le sprite est sélectionnable à la souris (option Non sélectionnable = faux)

Exemple

La ressource 3D « Electrical Pusher » présent en bibliothèque utilise ce Comportement.

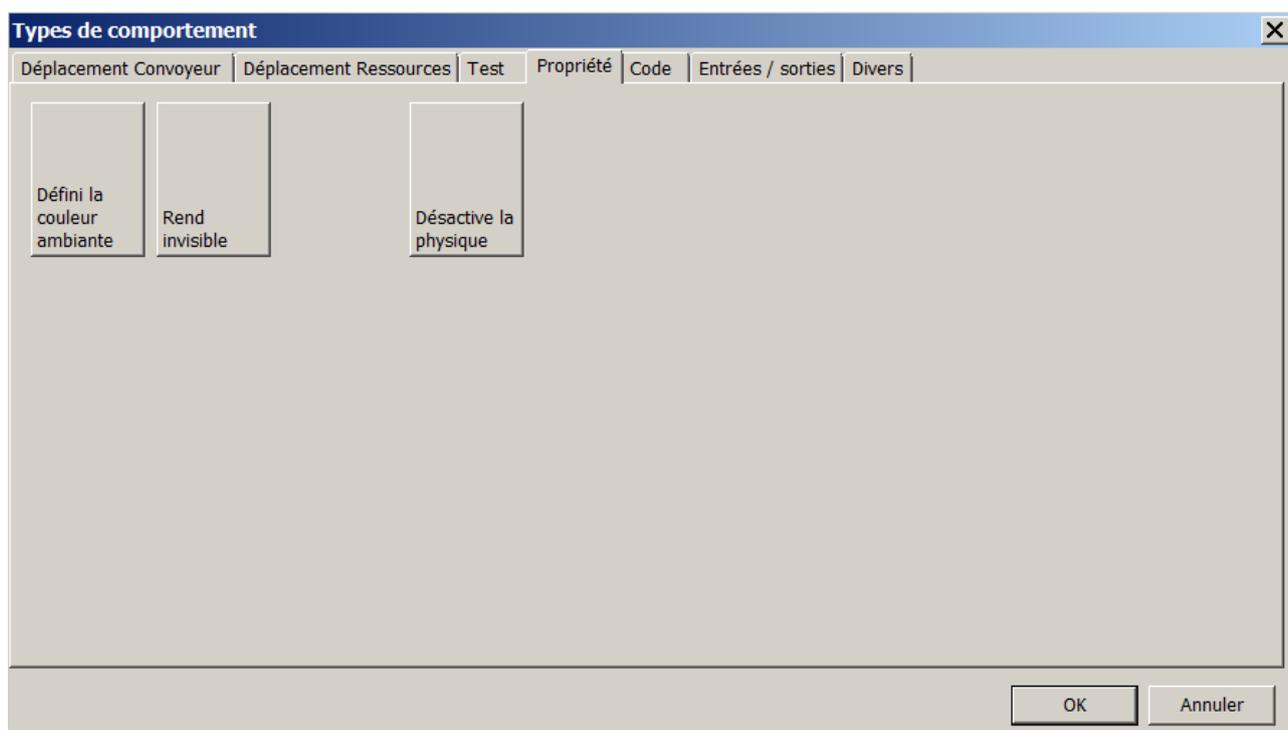


Teste si l'objet est survolé par la souris :

- Prend la valeur 1 si l'utilisateur survole l'objet à la souris.

Propriété

Ces comportements permettent de modifier certaines propriétés.



Défini la couleur ambiante :

- Permet de définir la couleur ambiante d'un Sprite 3D.

Important ! Ce comportement colorie les matériaux du sprite. La colorisation de ces matériaux doit avoir été permise au préalable dans les options du sprite.



Rend invisible :

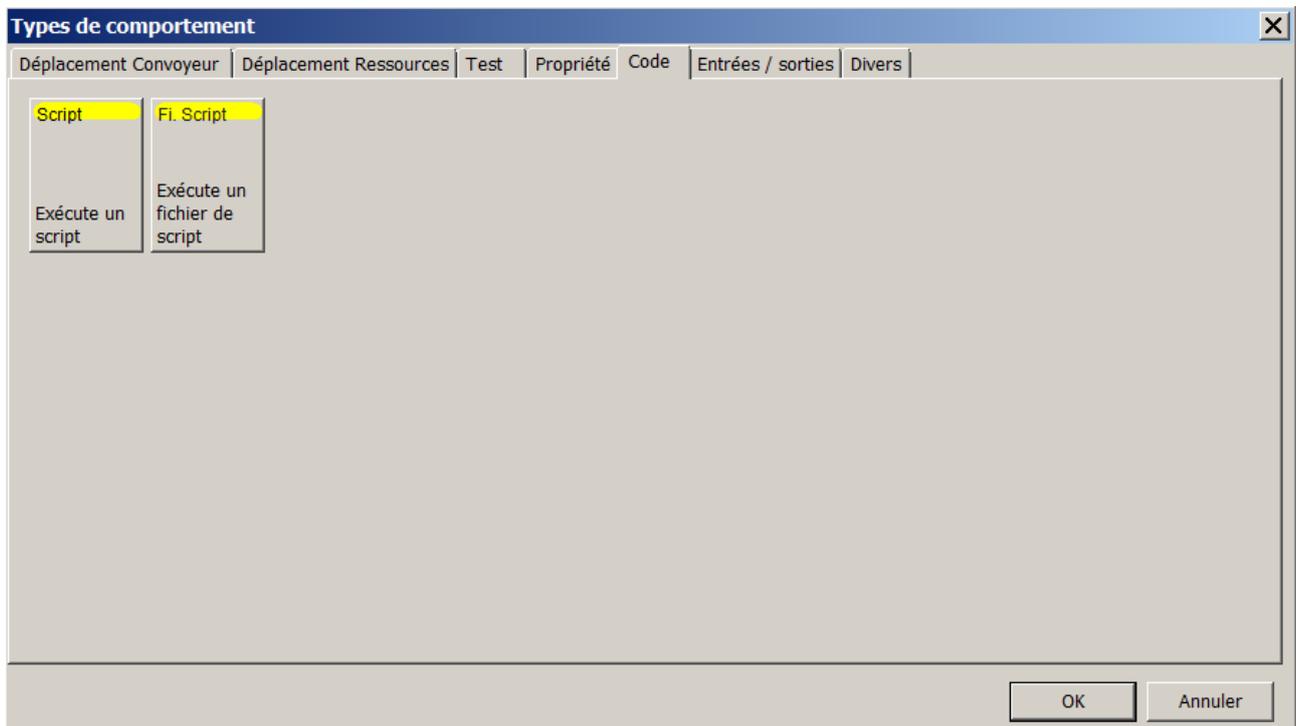
- Rend l'objet invisible si activé (à une valeur 1).



Désactive la physique :

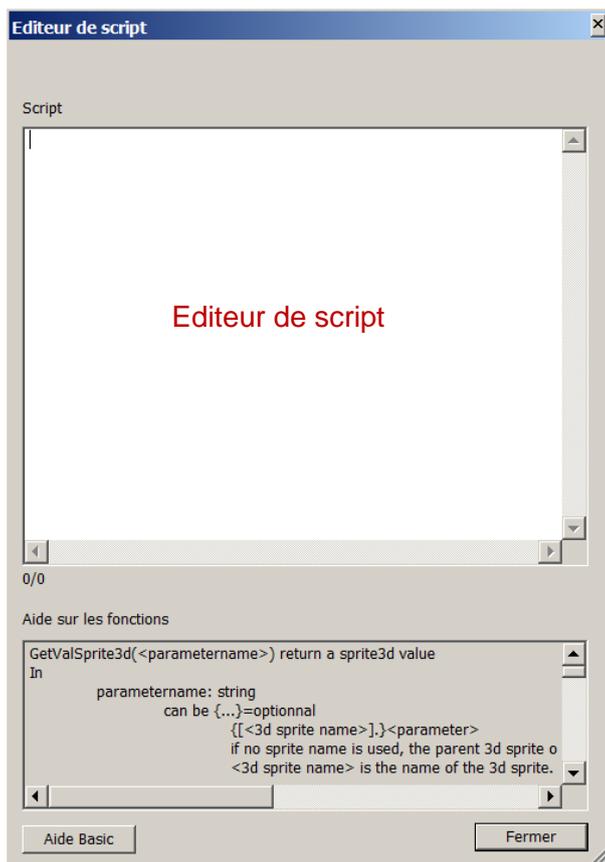
- Désactive la physique pour le Sprite 3D si activé (à une valeur 1).

Code



Exécute un script :

- Exécute le script présent dans l'onglet « Code » si la valeur courante du comportement est différente de 0.



Le langage utilisé dans l'éditeur de script est le Beebasic. Pour plus d'informations, vous pouvez vous référer au fichier d'aide basic_api.chm présent dans le répertoire d'installation de VIRTUAL UNIVERSE PRO et à l'aide disponible sous l'éditeur de script.

Voici la liste des principales fonctions utilisées par l'éditeur de script :

- **Getbehavior**(<paramètre>) retourne une valeur associée à un Comportement.

<paramètre> désigne le paramètre. Il peut désigner un Sprite 3D par son nom. Si ce n'est pas le cas, c'est le Sprite 3D parent du Comportement qui est utilisé. La syntaxe est [<nom du Sprite 3D>].<nom du paramètre>.
- **SetBehavior**(<paramètre>, <valeur>) écrit la valeur d'un Comportement.
- **GetValSprite3d**(<paramètre>) : retourne une valeur associée à un Sprite 3D.
- **SetValSprite3d**(<paramètre>, <valeur>) : modifie une valeur associée à un Sprite 3D.
- **Getuniverse**("time") : retourne le « temps » actuel.

Le nom pour la référence aux Comportements doit respecter la syntaxe suivante :

- Un nom sans chemin d'accès : cherchera le premier Comportement dont le nom commence par ce texte dans l'ensemble Comportements du Monde courant ?
- ..\<nom> : un Comportement nommé frère du Comportement courant,
- <nom de sprite>\<nom de Comportement> : un Comportement nommé enfant d'un Sprite 3D,
- Il est aussi possible de référencer le type de valeur de la manière suivante : [<nom du Comportement>].<type de valeur>. Les types valeurs sont : « internalvalue », « currentvalue », « values ».



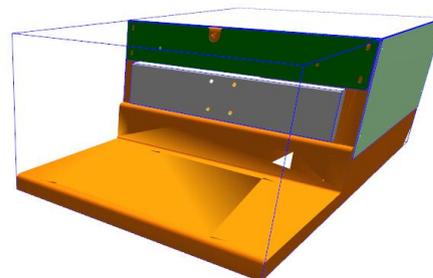
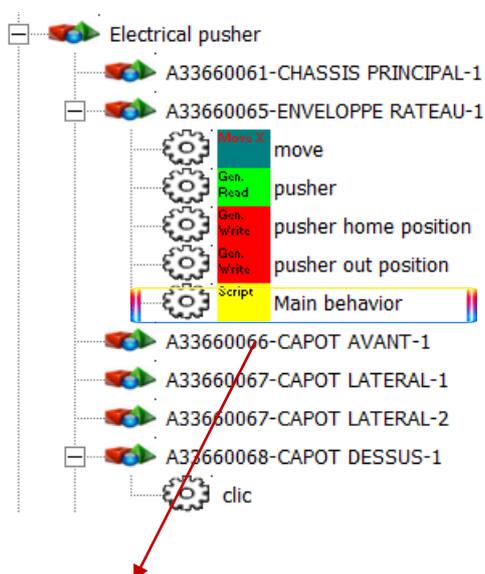
Exécute un fichier de script :

- Exécute un script contenu dans un fichier sir la valeur courante du Comportement est différente de 0.
Cette solution est particulièrement utile dans le cas d'un script d'une longueur trop importante pour l'éditeur intégré à VIRTUAL UNIVERSE PRO.

Exemple 1

Cas de la ressource « Electrical Pusher » présente au sein de la bibliothèque de démonstration VIRTUAL UNIVERSE PRO.

Le script « Main behavior » permet de piloter la rentrée/sortie du tiroir par action du comportement « move » et sur ordre de la commande d'entrée « pusher ». Il retourne également la position du tiroir (variables externes « pusher home position » et « pusher out position »)



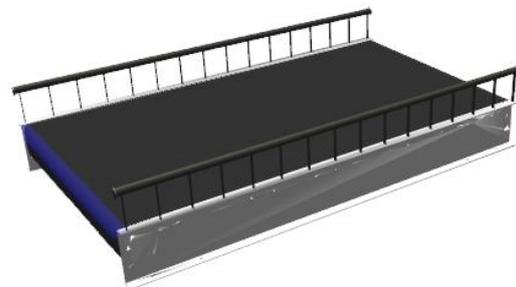
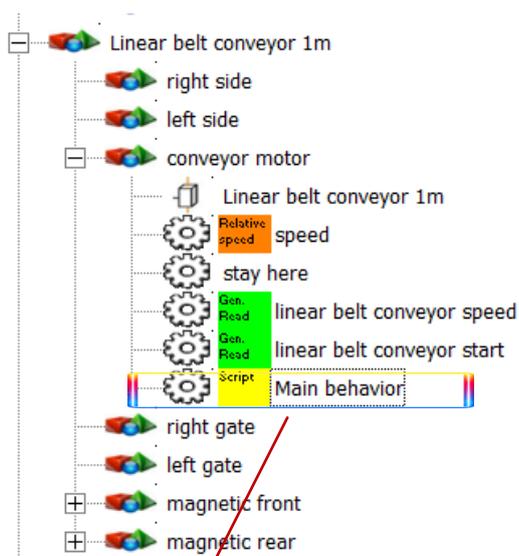
Main behavior

```
myloop:
curpos=GetValSprite3d("POSX")
if getbehavior("../pusher.currentvalue")=true or getbehavior("../A33660068-CAPOT DESSUS-1\clac.currentvalue")=true then
setbehavior("../move.values",1)
else
setbehavior("../move.values",0)
endif
if curpos>=5 then
setbehavior("../pusher home position.values",1)
else
setbehavior("../pusher home position.values",0)
endif
if curpos<=-1.2 then
setbehavior("../pusher out position.values",1)
else
setbehavior("../pusher out position.values",0)
endif
goto myloop
```

Exemple 2

Cas de la ressource « Linear belt conveyor » présente au sein de la bibliothèque de démonstration VIRTUAL UNIVERSE PRO.

Le script « Main behavior » permet de piloter la vitesse du tapis de convoyeur par action du comportement « speed » et sur ordre des 2 commandes d'entrée «linear belt conveyor speed » et « linear belt conveyor start ».



Main behavior

```

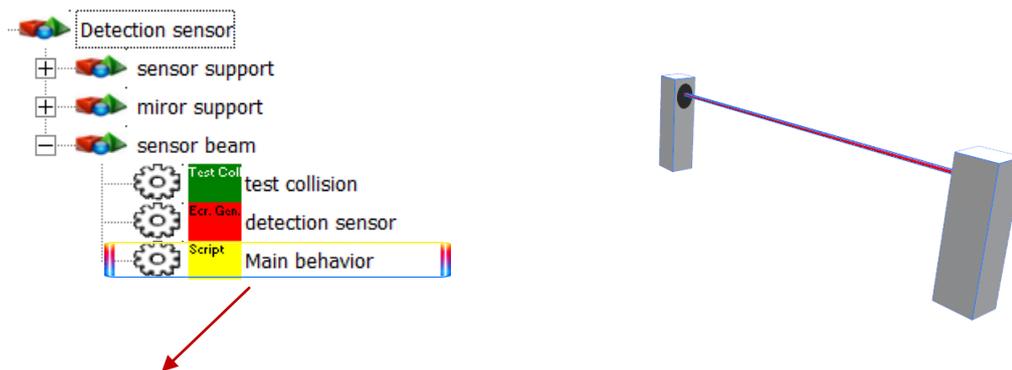
myloop:
conveyor_speed=getbehavior("../linear belt conveyor speed")
conveyor_start=getbehavior("../linear belt conveyor start")
if conveyor_start=true then
setbehavior("../speed.values",conveyor_speed)
else
setbehavior("../speed.values",0)
endif
goto myloop

```

Exemple 3

Cas de la ressource « Detection sensor présente au sein de la bibliothèque de démonstration VIRTUAL UNIVERSE PRO.

Le script « Main behavior » permet de retourner la détection de collision obtenue par le comportement « test collision » vers la variable externe « detection sensor ».



Main behavior

```

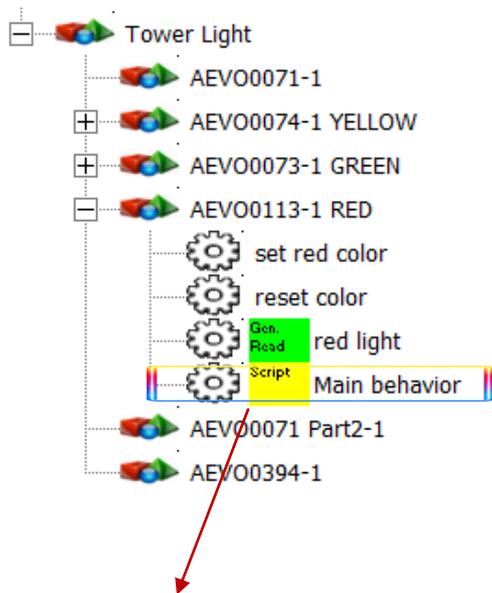
collision=getbehavior("../test collision")
setbehavior("../detection sensor.values", collision)

```

Exemple 4

Cas de la ressource « Tower light » incluse dans la ressource « Electrical enclosure » présente au sein de la bibliothèque de démonstration VIRTUAL UNIVERSE PRO.

Le script « Main behavior » de la lampe RED permet de faire clignoter la lampe (période 0.5s) sur réception du signal d'entrée « red light ».



Main behavior

Sub wait(t)

t1=getuniverse("time")

miloop:

t2=getuniverse("time")

*if(t2-t1<t*1000) then*

goto miloop

endif

End Sub

Sub flash(t)

wait(t)

setbehavior("[..\set red color].values",1)

setbehavior("[..\reset color].values",0)

wait(t)

setbehavior("[..\set red color].values",0)

setbehavior("[..\reset color].values",1)

End Sub

myloop:

signal=getbehavior("[..\red light].currentvalue")

if signal=true then

while (signal=true)

signal=getbehavior("[..\red light].currentvalue")

flash(0.5)

wend

else

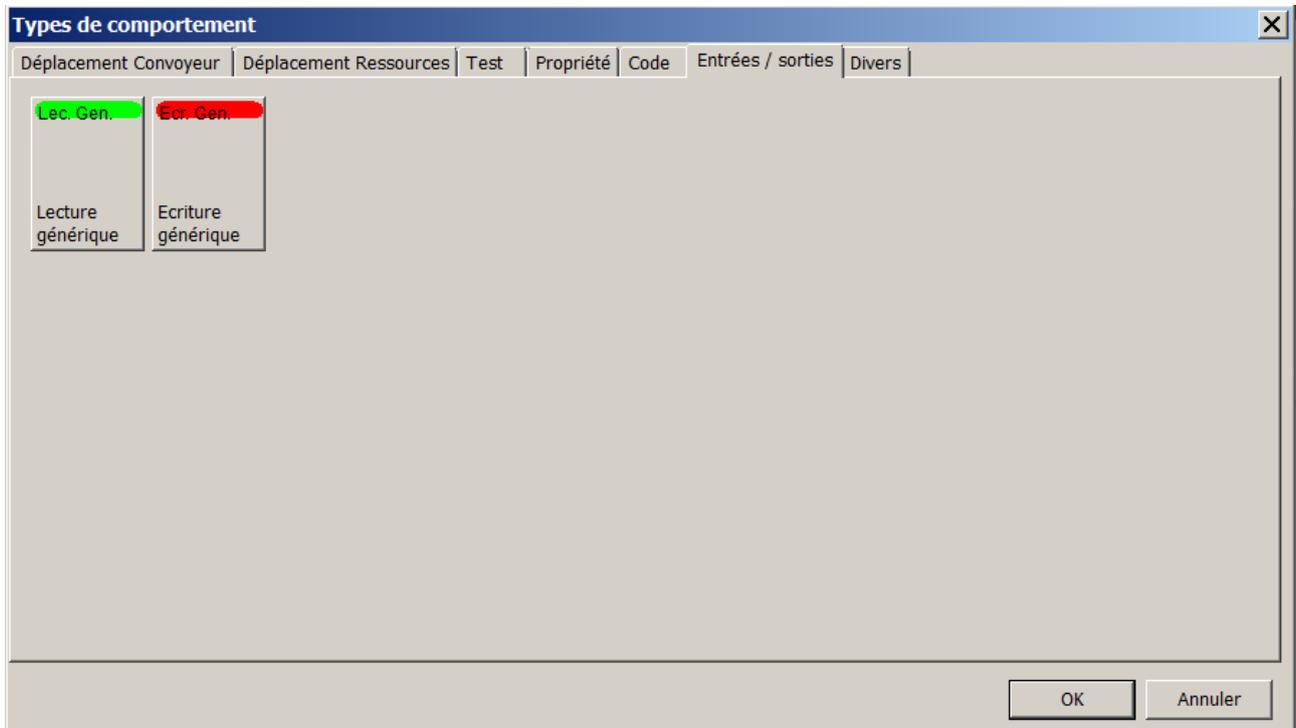
setbehavior("[..\set red color].values",0)

setbehavior("[..\reset color].values",1)

endif

goto myloop

Entrées / sorties



Lecture générique :

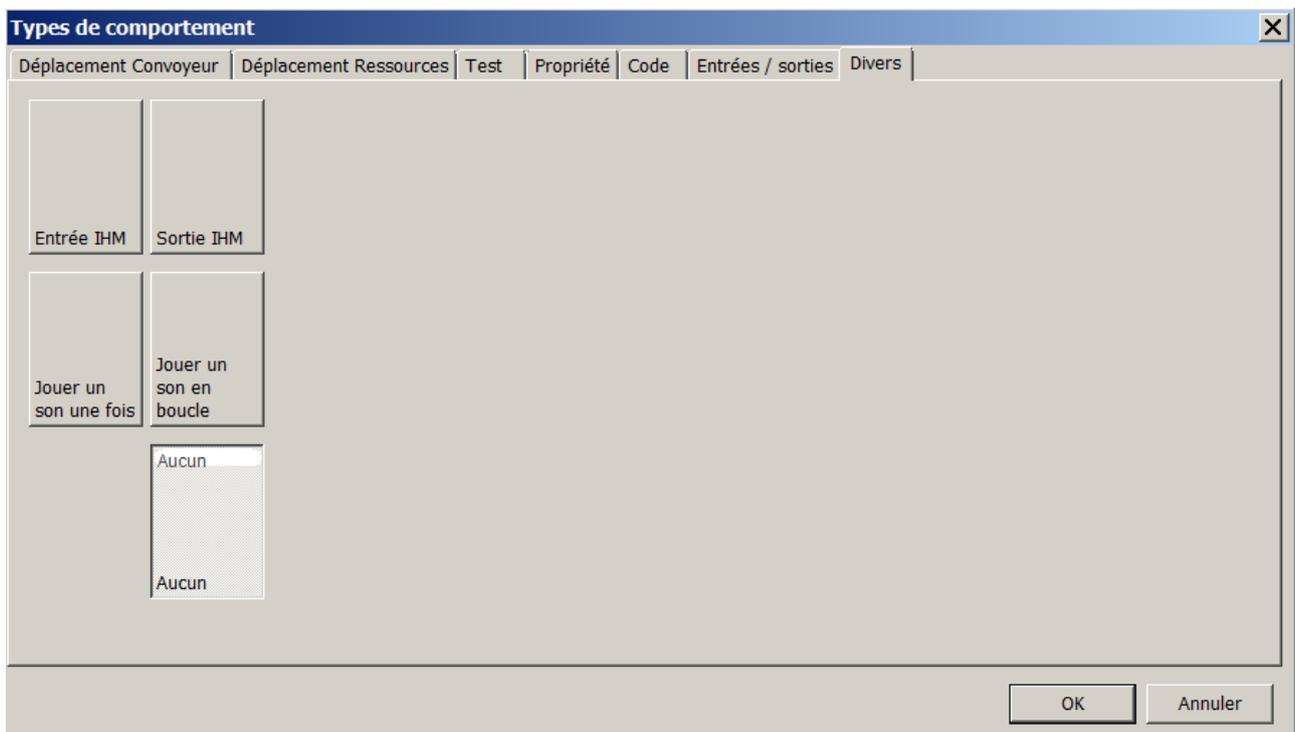
- Variable d'entrée de VIRTUAL UNIVERSE PRO et de sortie pour le logiciel externe. Ce Comportement ne fera que lire une variable provenant du logiciel externe.



Ecriture générique :

- Variable de sortie de VIRTUAL UNIVERSE PRO et d'entrée pour le logiciel externe. Ce Comportement ne fera qu'écrire une variable dans le logiciel externe.

Divers



[Entrée IHM :](#)

- Crée un lien avec un élément IHM en entrée (un bouton poussoir, par exemple)



[Sortie IHM :](#)

- Crée un lien avec un élément IHM en sortie (un voyant, par exemple).



[Jouer un son une fois :](#)

- Permet de jouer un fichier de son une seule fois si la valeur courante du Comportement est différente de 0. Le son 3D sera perçu comme provenant du Sprite 3D parent. De plus, la valeur courante du Comportement peut moduler le volume ou la vitesse du son joué. Ce Comportement permet, par exemple, de reproduire le bruit d'un moteur en fonction de sa vitesse de rotation.



[Jouer un son en boucle :](#)

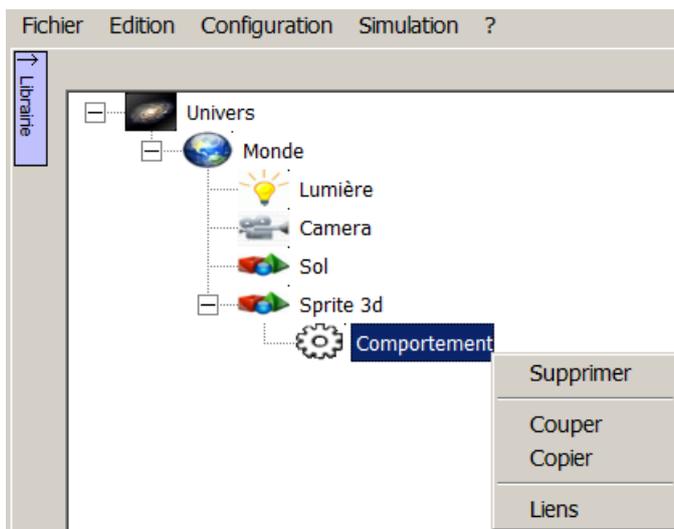
- Idem mais le son est joué en boucle.



[Aucun :](#)

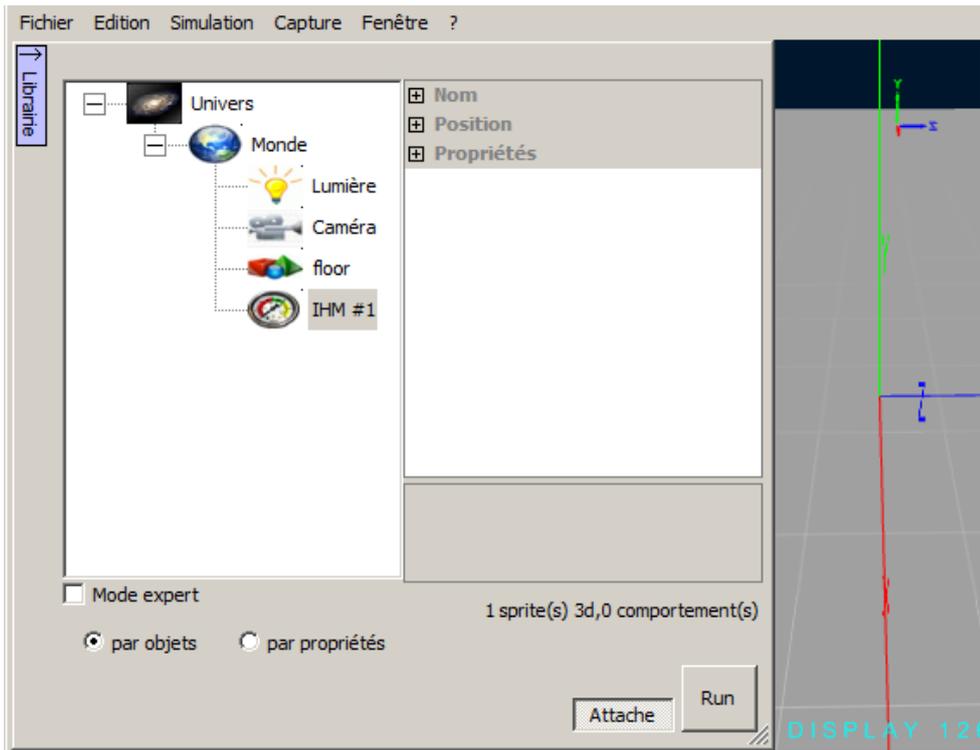
- Type par défaut, représente un signal interne à VIRTUAL UNIVERSE PRO. Le Comportement « Aucun » est inerte.

Fonctionnalités au niveau Comportement

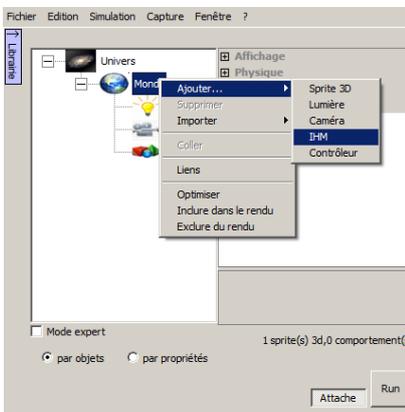


Liens : Permet d'accéder à la fenêtre de Liens externes du Comportement

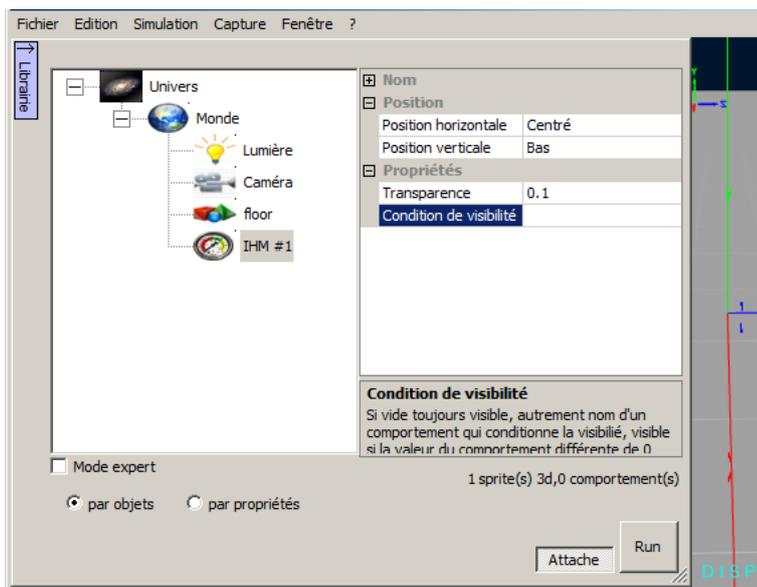
Propriétés des IHMs



Le menu qui s'ouvre lorsqu'on clique avec le bouton droit de la souris sur le monde permet l'ajout d'un IHM :



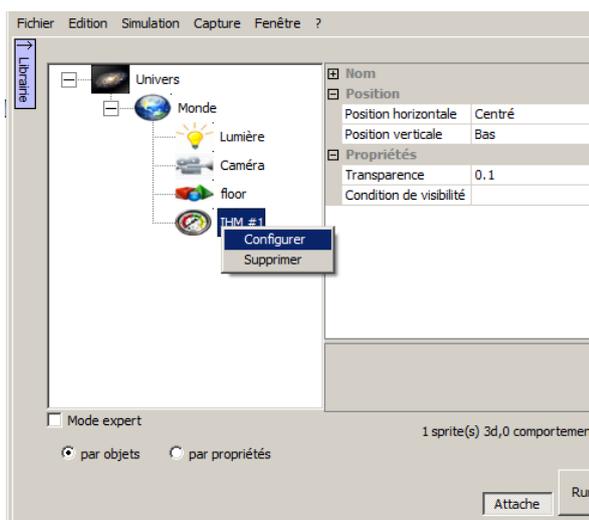
Propriétés détaillées des IHM



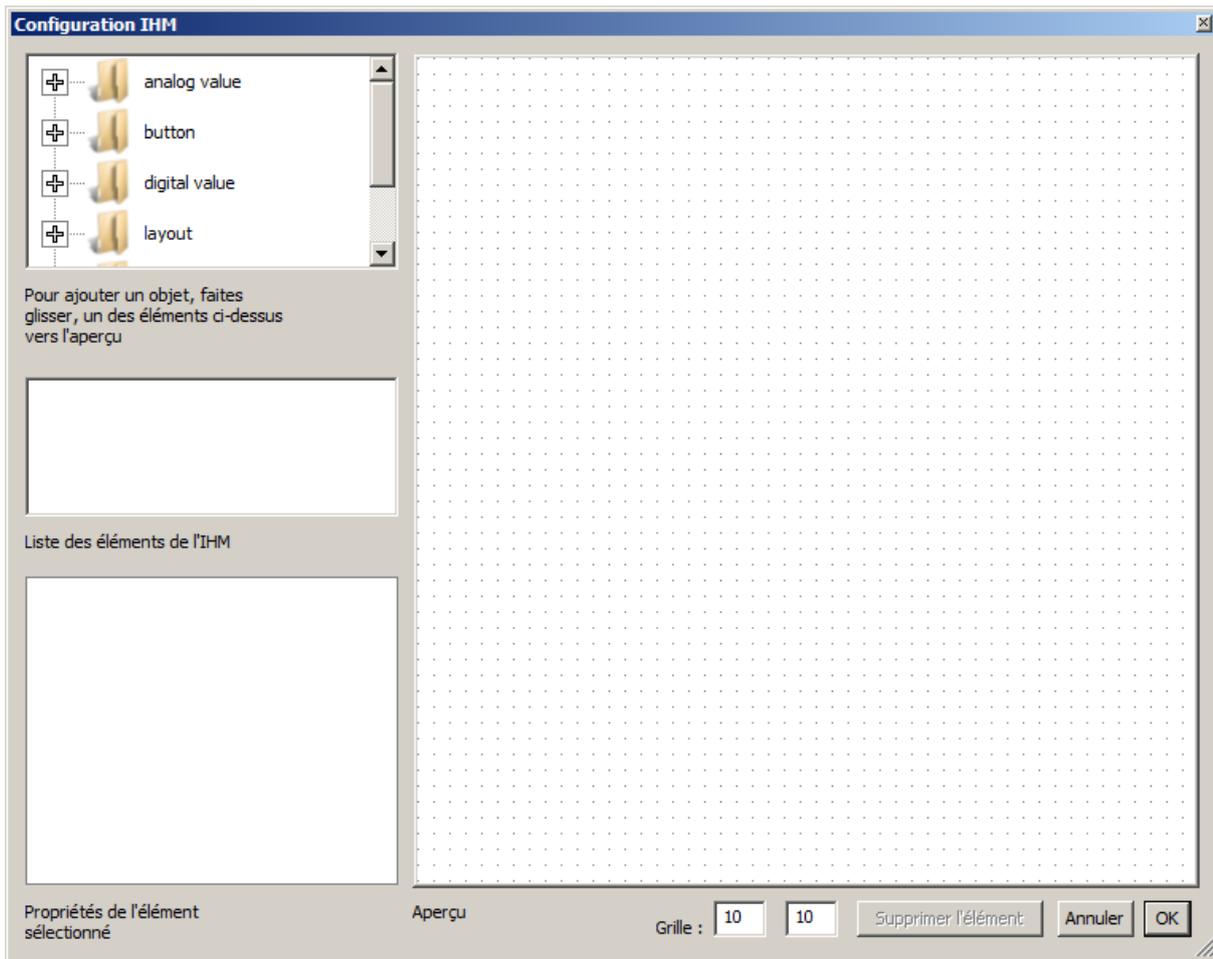
Les paramètres positions permettent de définir où apparaîtra l'IHM dans le fenêtre de rendu.

Les paramètres "propriétés" permettent de définir la transparence de l'IHM ainsi qu'une éventuelle condition (état d'un comportement) pour l'affichage. Si la zone "condition" contient le nom valide d'un comportement, alors l'IHM sera affiché si la valeur du comportement est différente de 0 et cachée dans le cas contraire. Cette condition permet d'afficher de façon conditionnelle certains IHM pour, par exemple, créer un système de menus.

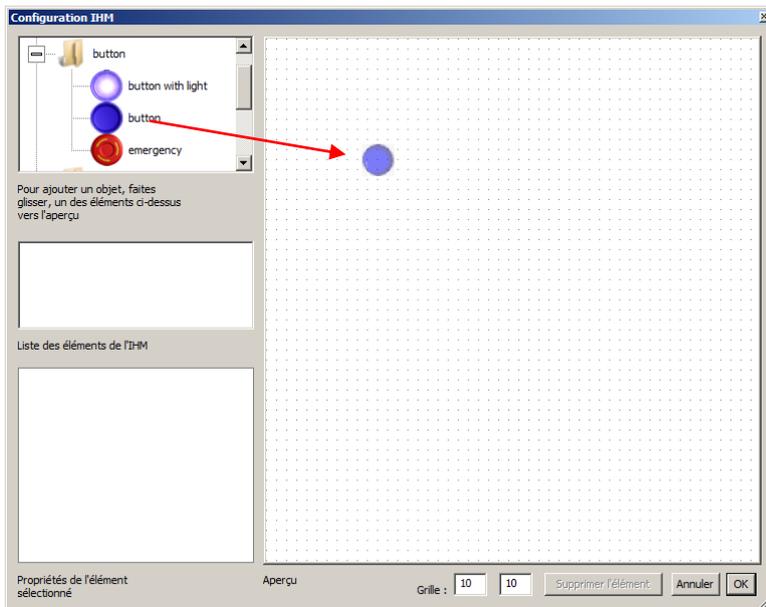
Création ou modification d'un IHM

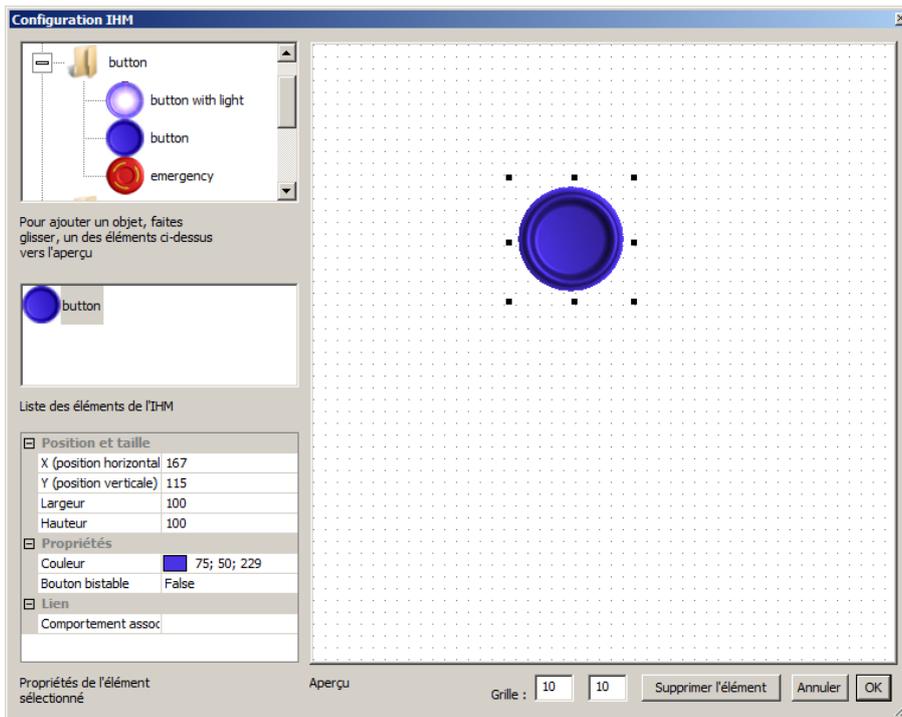


Le menu contextuel ou un double clic sur un élément IHM permettent d'accéder à la fenêtre de configuration :



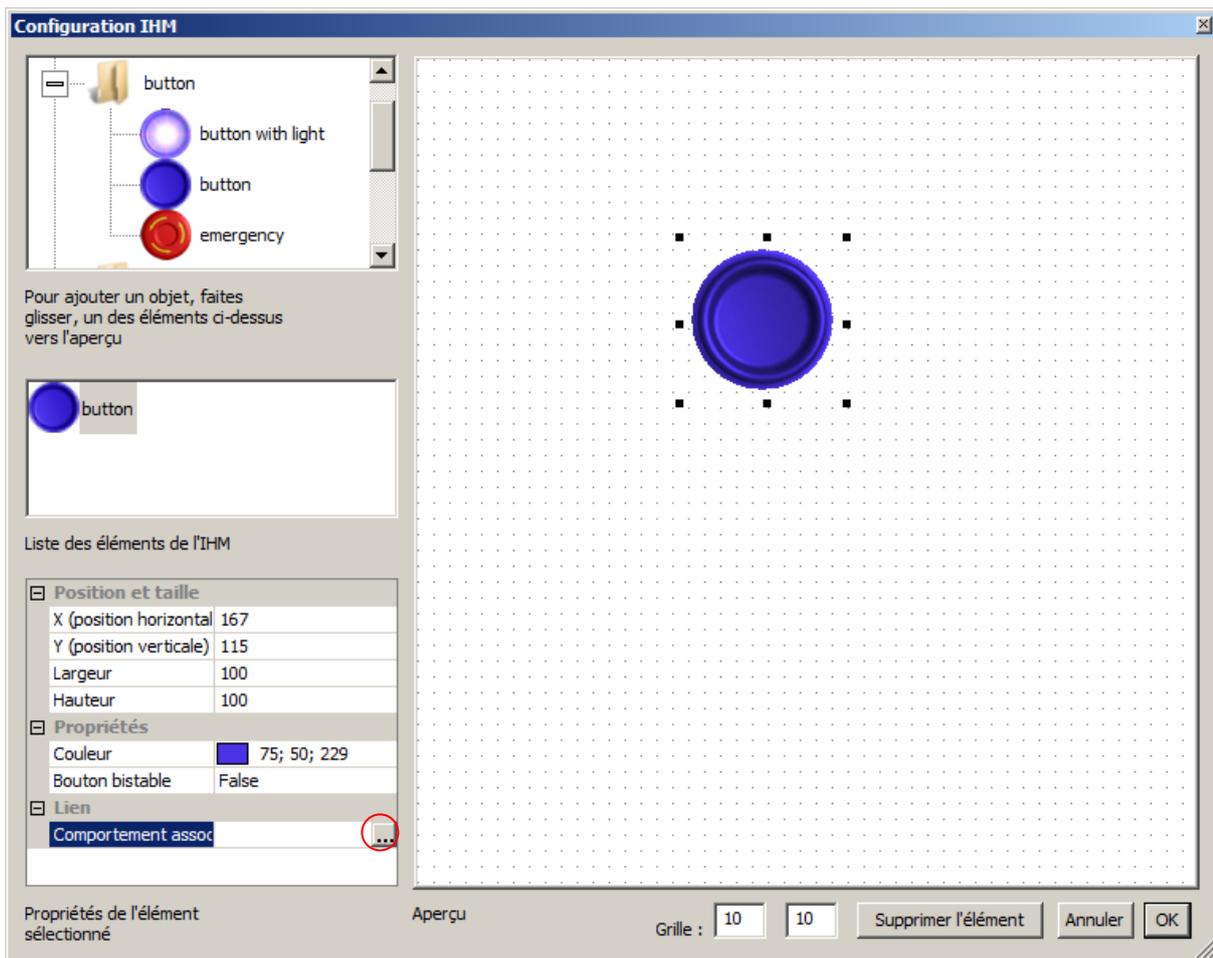
Pour définir les éléments d'un IHM, saisissez les éléments disponibles dans la partie supérieure gauche et faites les glisser vers la zone de droite.

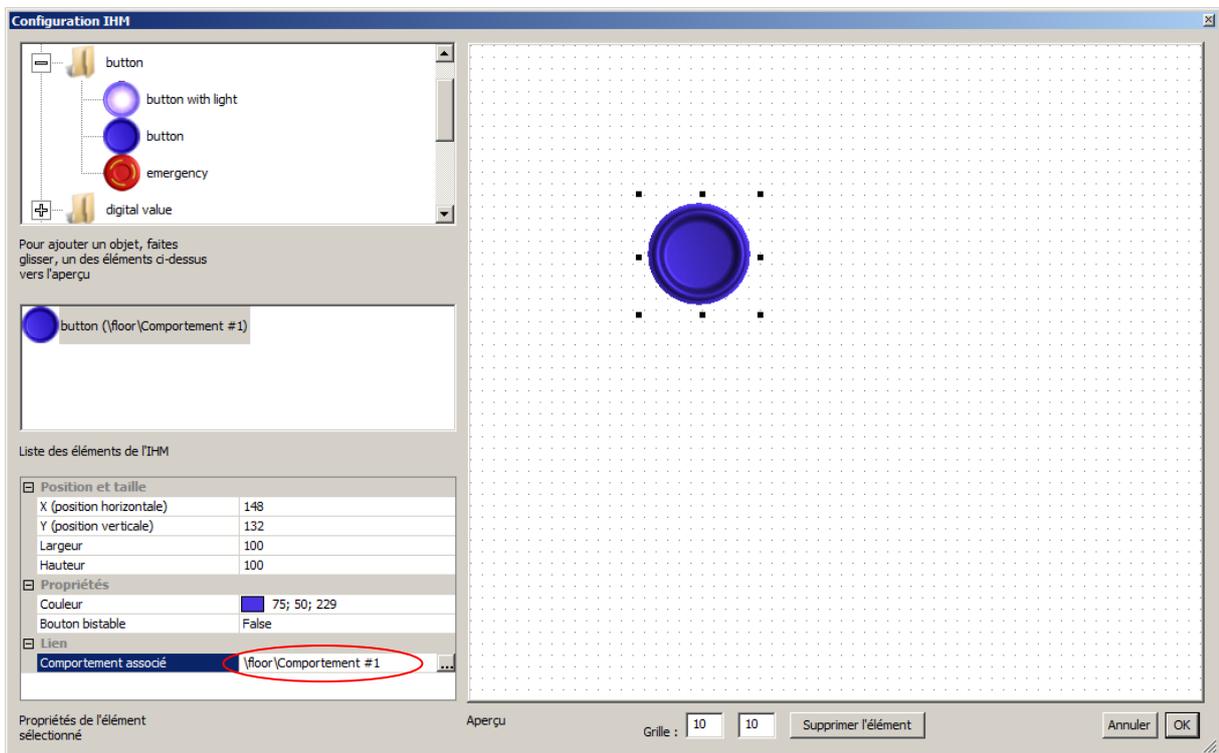
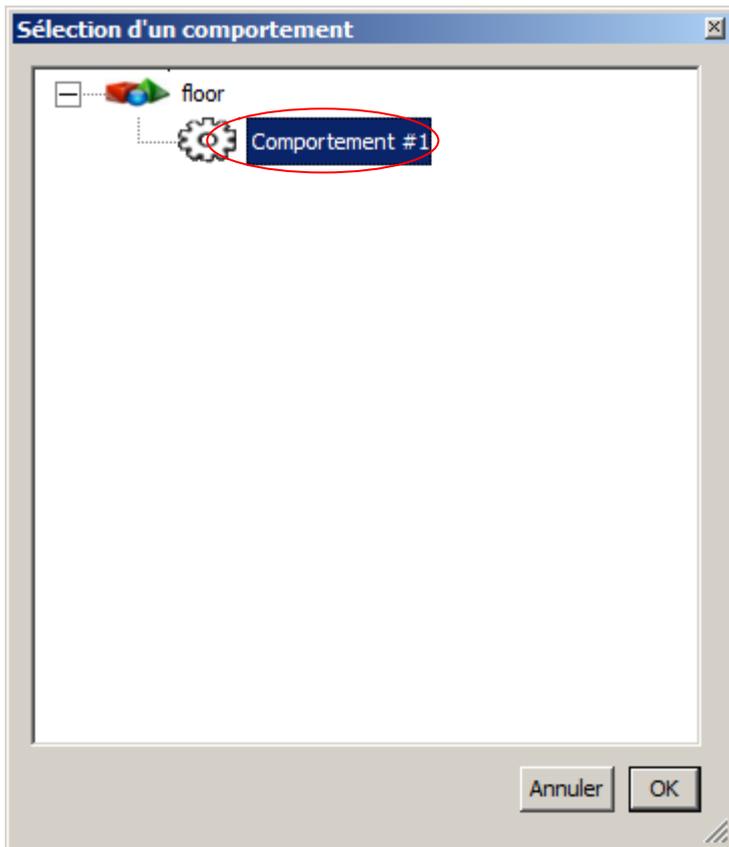




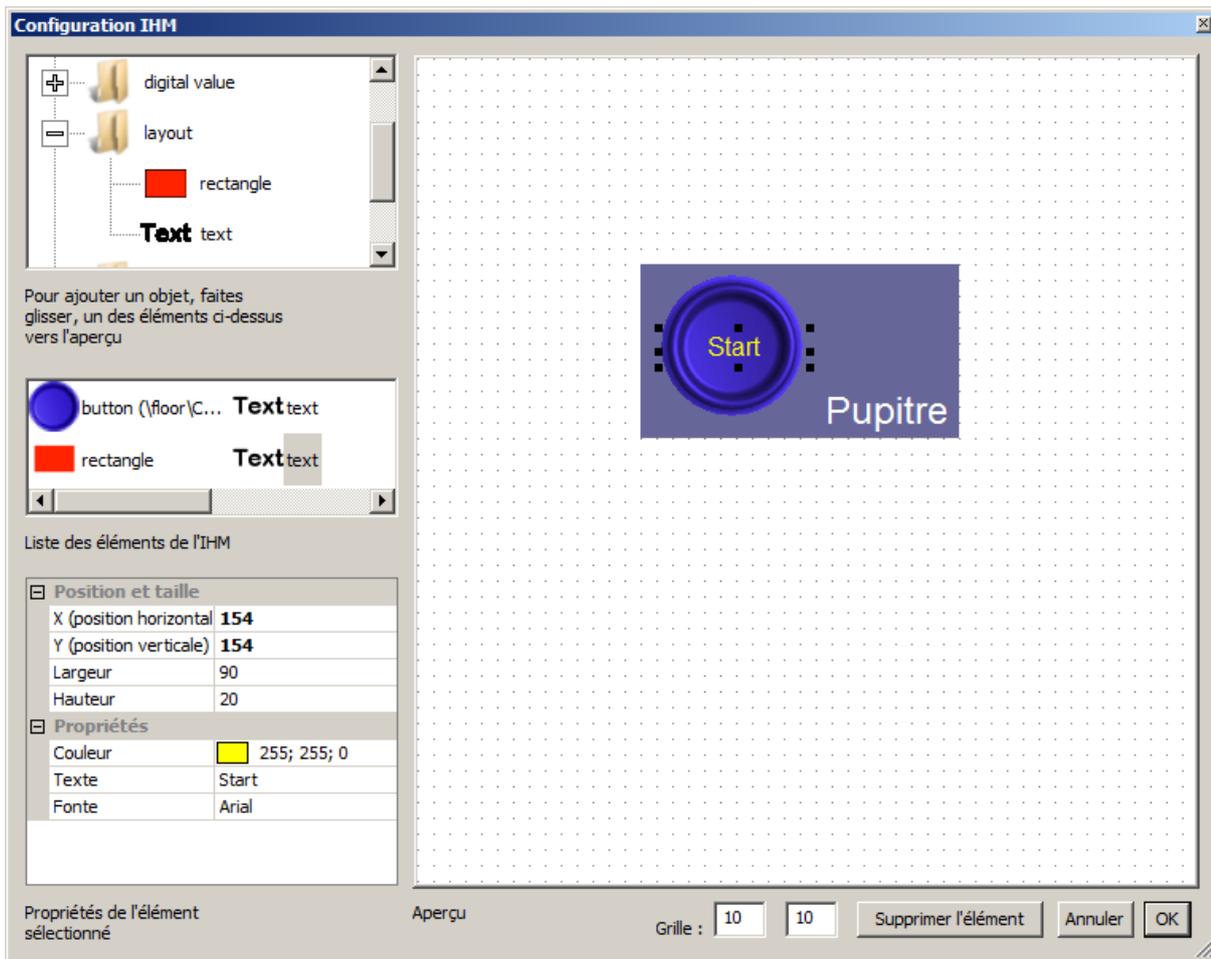
Lorsqu'un élément IHM est sélectionné, ses propriétés sont accessibles en bas à gauche.

La rubrique "Liens" permet de définir la relation entre l'élément et un comportement :





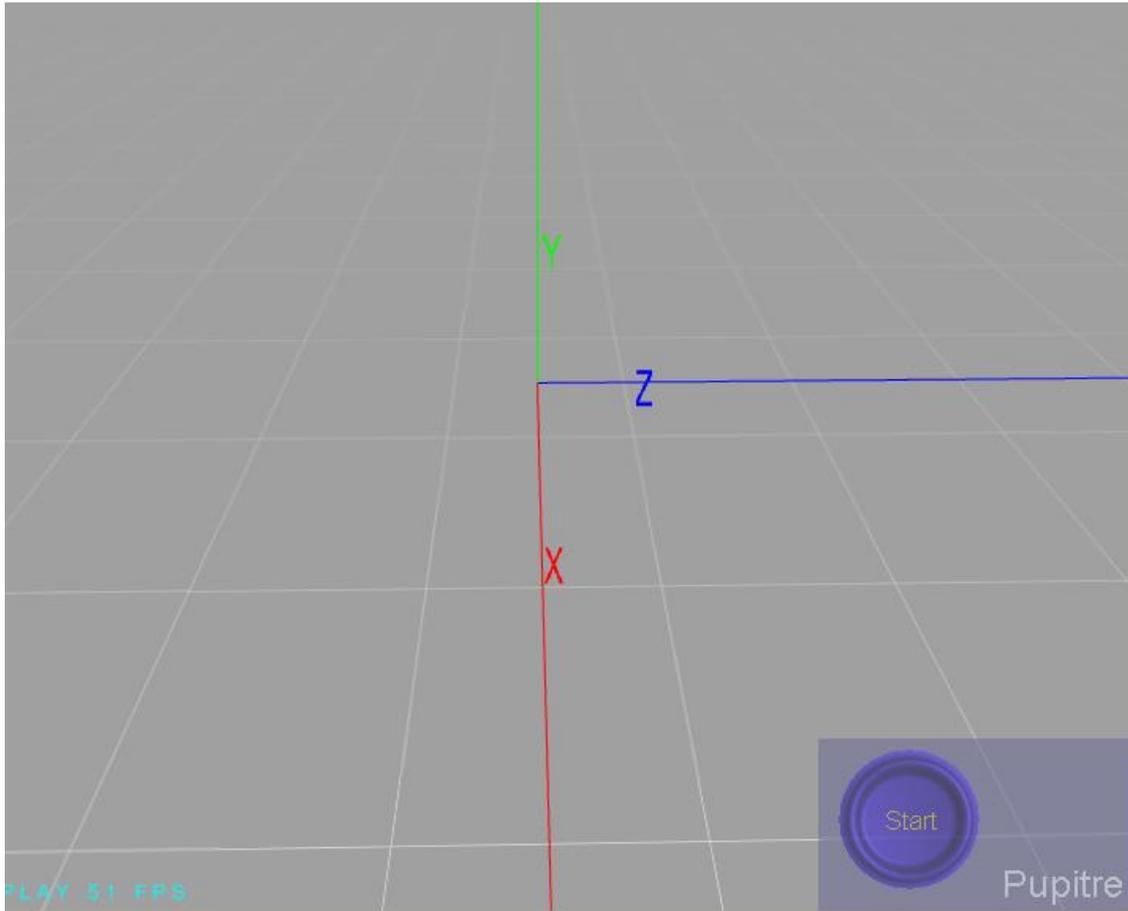
Des éléments de dessins (textes, rectangles colorés) peuvent également être utilisés pour matérialiser le fond d'un pupitre ainsi que des indications sur les éléments IHM :



Remarques :

- les valeurs de la grille vous permettent de positionner facilement les éléments, des valeurs inférieures à 2 désactivent la grille,
- si plusieurs éléments se trouvent à un même emplacement, cliquer plusieurs fois avec le bouton gauche sur ceux-ci permet de sélectionner alternativement l'un d'entre eux.

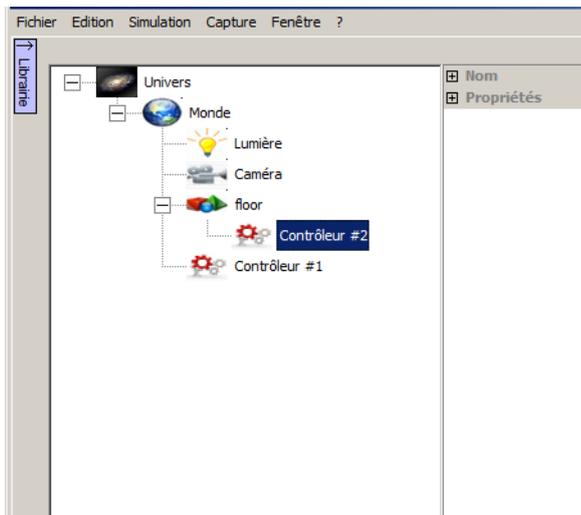
Exemple d'IHM affiché dans la fenêtre de rendu :



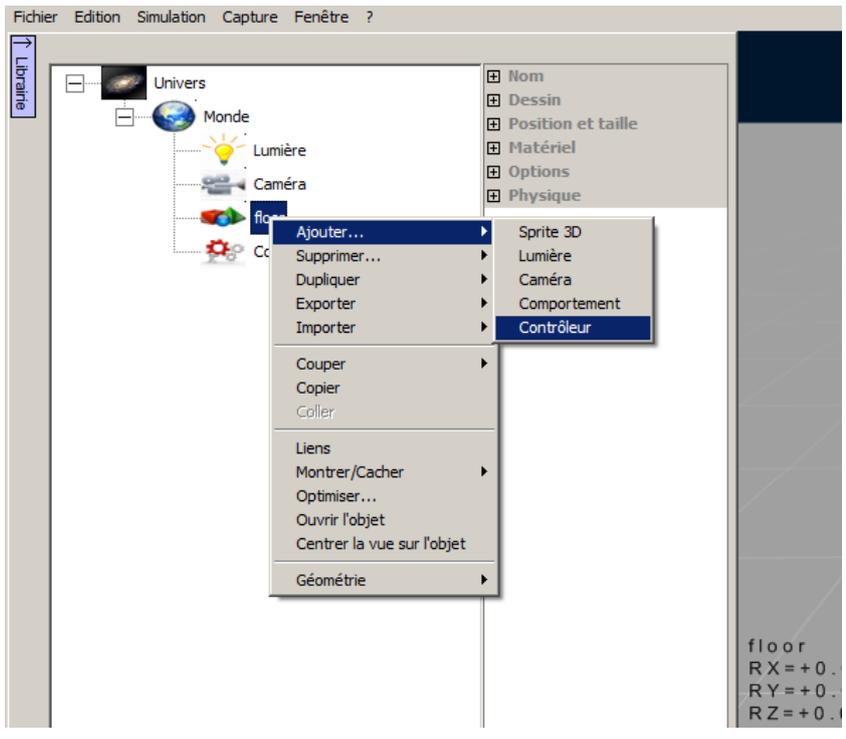
Propriétés des Contrôleurs

Fonctions de programmation

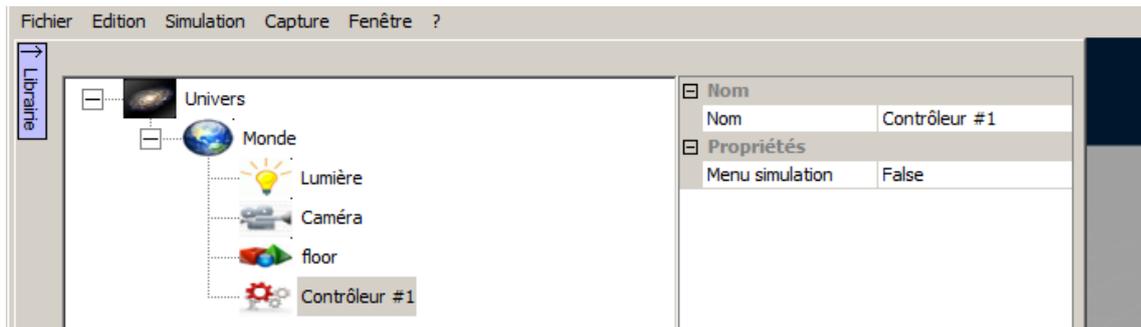
Les fonctions de programmation apparaissent dans les projets comme un ou plusieurs éléments "Contrôleur" enfants de l'élément "Monde" ou des éléments "Sprite 3d". Chaque contrôleur ainsi créé pourra contenir une ou plusieurs pages de programme (zones non limitées en taille) écrits en langage Ladder ou Fbd/SFC (blocs de fonctions et Grafcet). Les contrôleurs peuvent lire et écrire les valeurs des comportements du projet. Chaque contrôleur peut également lire et écrire des variables locales à chaque contrôleur. Les contrôleurs exécutent leurs pages de programmes en mode RUN.



L'ajout d'un contrôleur est réalisé par un clic droit sur l'élément "Monde" ou un élément "Sprite 3d" :

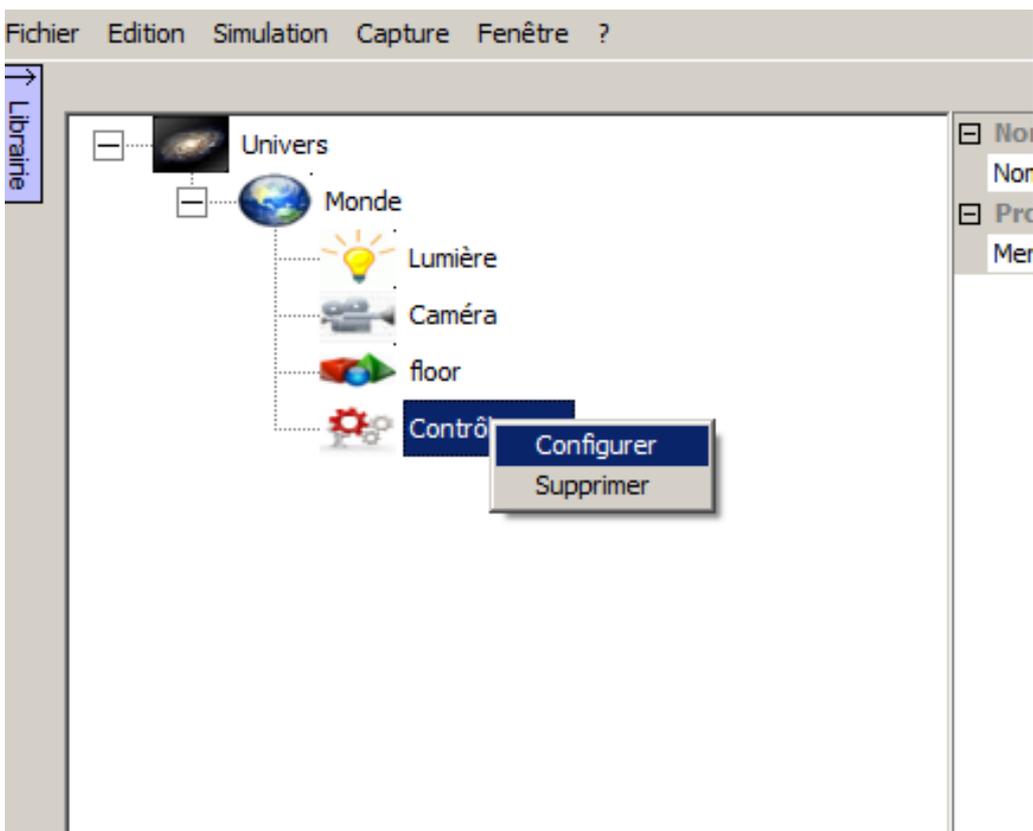


Propriétés détaillées des contrôleurs

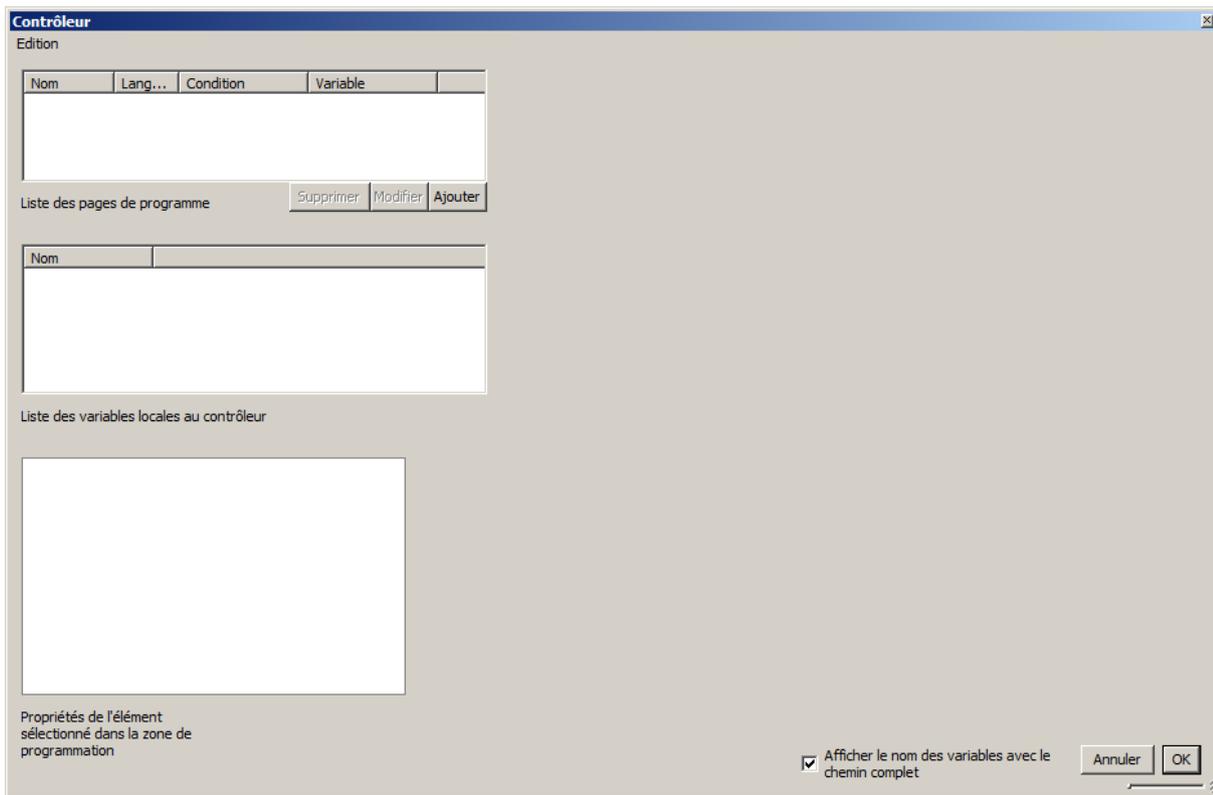


Le paramètre "Menu simulation" définit la visibilité du contrôleur (et des programmes qu'il contient) dans le menu "Simulation / Mise au point / Programme simulation" en mode RUN.

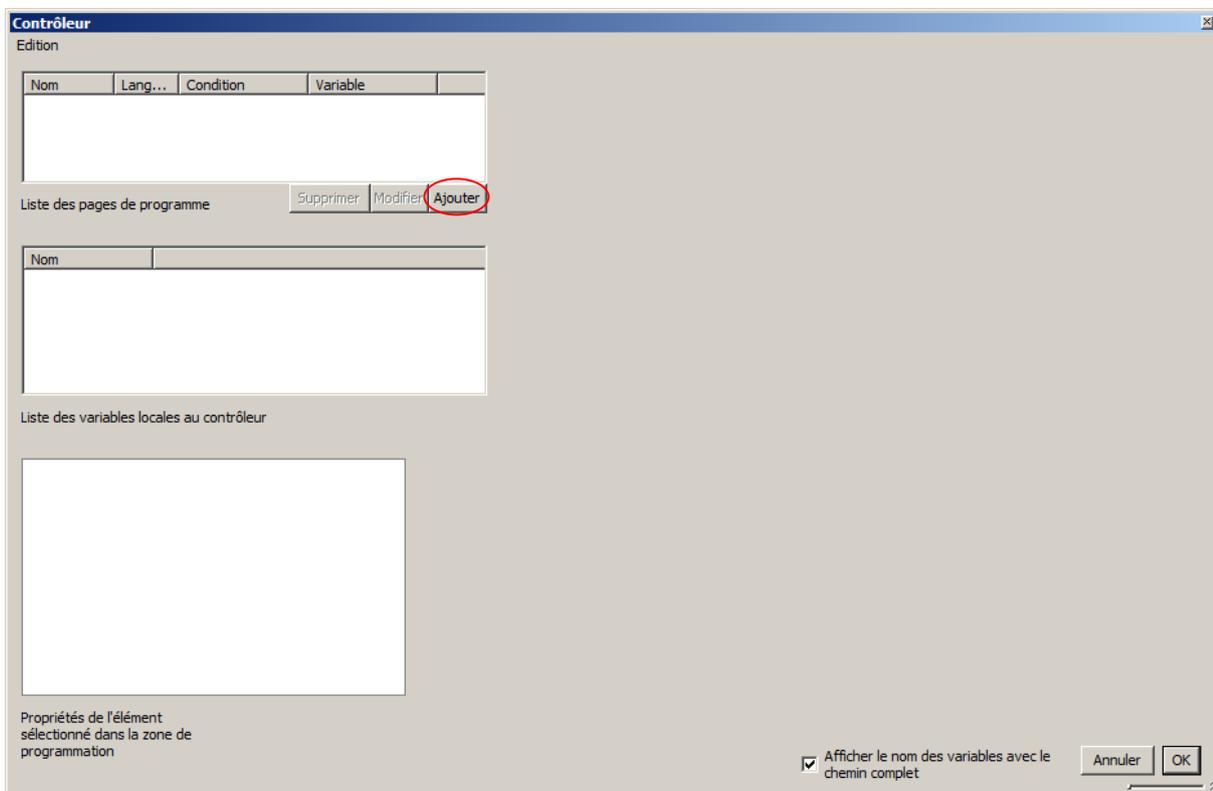
Programmation d'un contrôleur

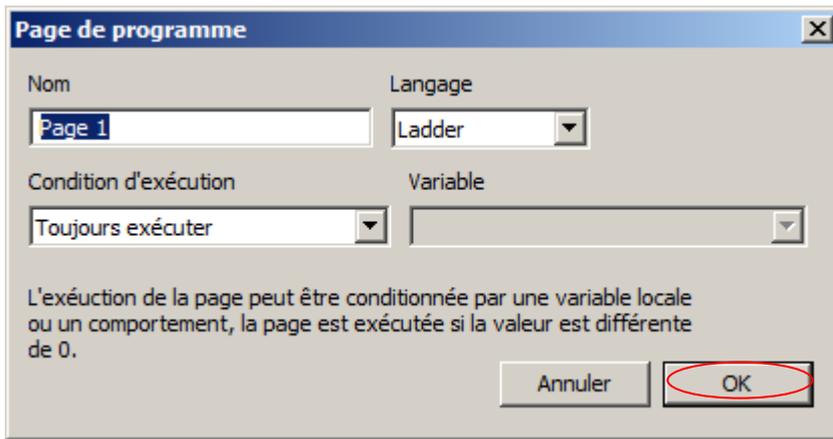


En mode STOP, le menu contextuel ou le double clic sur un élément "Contrôleur" permet d'ouvrir la fenêtre de configuration :



Pour créer une page de programme, cliquez sur "Ajouter" :

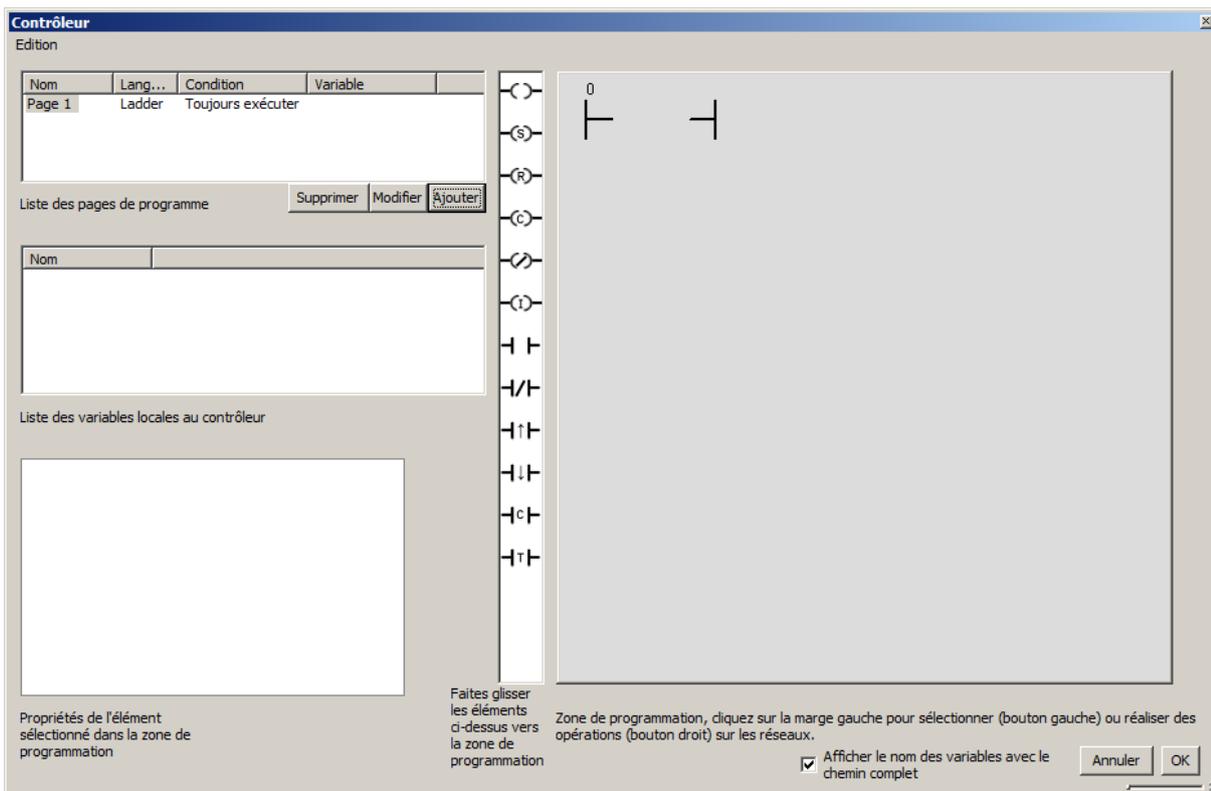




Le nom de la page, le langage utilisé ainsi qu'une éventuelle condition d'exécution peuvent être renseignés.

Remarques :

- le langage choisi ne pourra plus être modifié pour la page ainsi créée,
- la condition qui peut être l'état d'un comportement ou d'une variable locale permet de valider ou non l'exécution de chaque page de programme. Si la valeur de la variable locale ou du comportement est 0, la page n'est pas exécutée, elle l'est dans le cas contraire.



Éléments communs

L'ensemble des langages utilisent trois types de variables :

- comportements (l'état des comportements du projet), pour référencer un comportement, un chemin complet ou relatif peut être utilisé. Le chemin complet donne, en partant du monde le nom des sprites 3d ou des lumières parents. Par exemple : "\floor\robot\moteur" : comportement "moteur" enfant du sprite 3d "robot" lui même enfant du sprite 3d "floor". Les chemins relatifs, utilisables dans le cas d'un contrôleur enfant d'un sprite 3d spécifie le chemin depuis ce sprite 3d. Par exemple : ".\moteur" comportement "moteur" enfant du sprite 3d parent du contrôleur. L'utilisation de chemin relatifs permet de créer des ensembles sprites 3d + contrôleur duplicables, le contrôleur référence en effet dans ce cas les comportements enfants du sprite 3d qui est son parent. Par ailleurs, la case à cocher "Afficher le nom des variables avec le chemin complet" permet d'afficher ou non un format court (sans le chemin complet) pour le nom des variables rendant ainsi plus concis l'affichage des programmes.

- variables locales (variables locales à chaque contrôleur et communes à toutes les pages d'un même contrôleur), les variables locales sont initialisées à la valeur 0 au passage en RUN,

- variables systèmes :

- BLINK500MS : variable qui change d'état faux/vrai toutes les 500MS, typiquement utilisé pour faire clignoter des voyants,

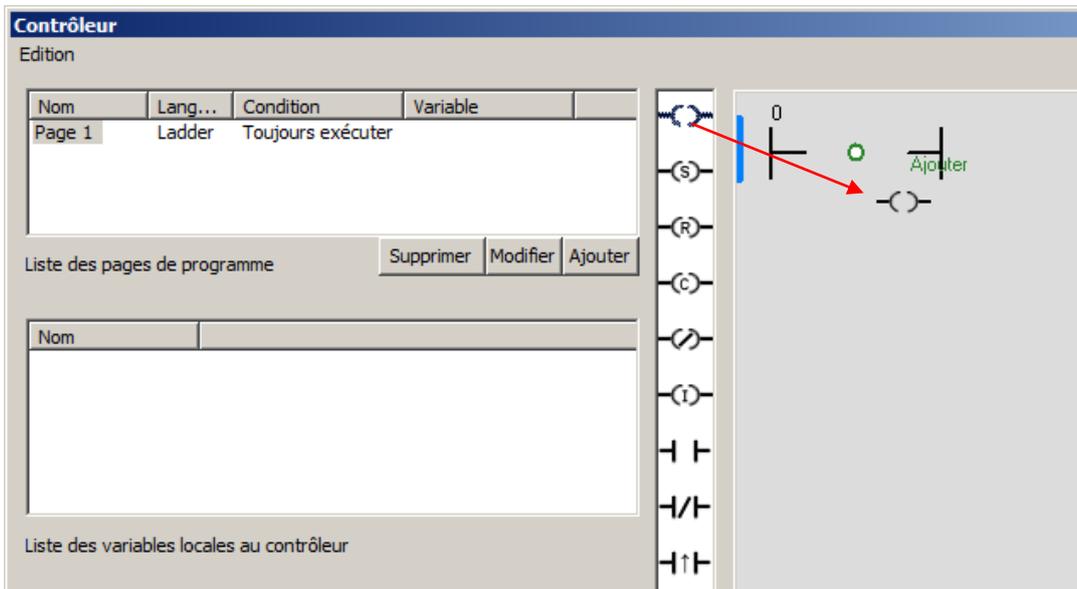
- FIRSTCYCLE : variable vraie uniquement au premier cycle d'exécution du programme, typiquement utilisée pour effectuer des initialisation. Cette variable est vraie uniquement si la page est exécutée au passage en RUN (non conditionnée par une condition fausse au passage en RUN).

- ELLAPSEDTIME : temps écoulé depuis la dernière scrutation en seconde.

Toutes ces variables sont de types numériques réels sur 8 octets. Par convention, pour les traitements booléens, elles sont considérées comme fausse si égale à 0 et comme vraie si différente de 0.

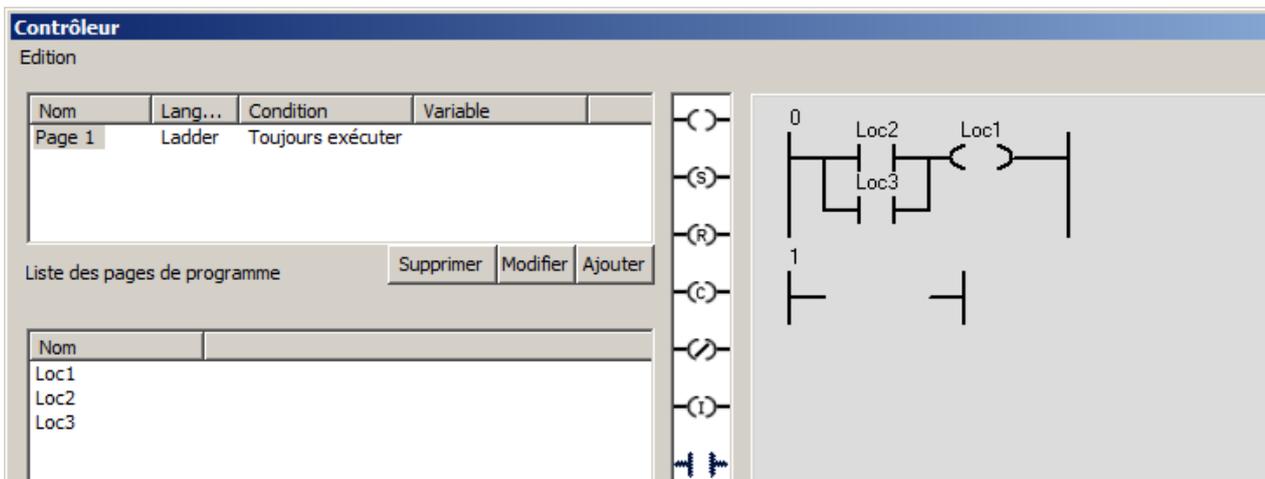
Langage Ladder

Création des programmes :

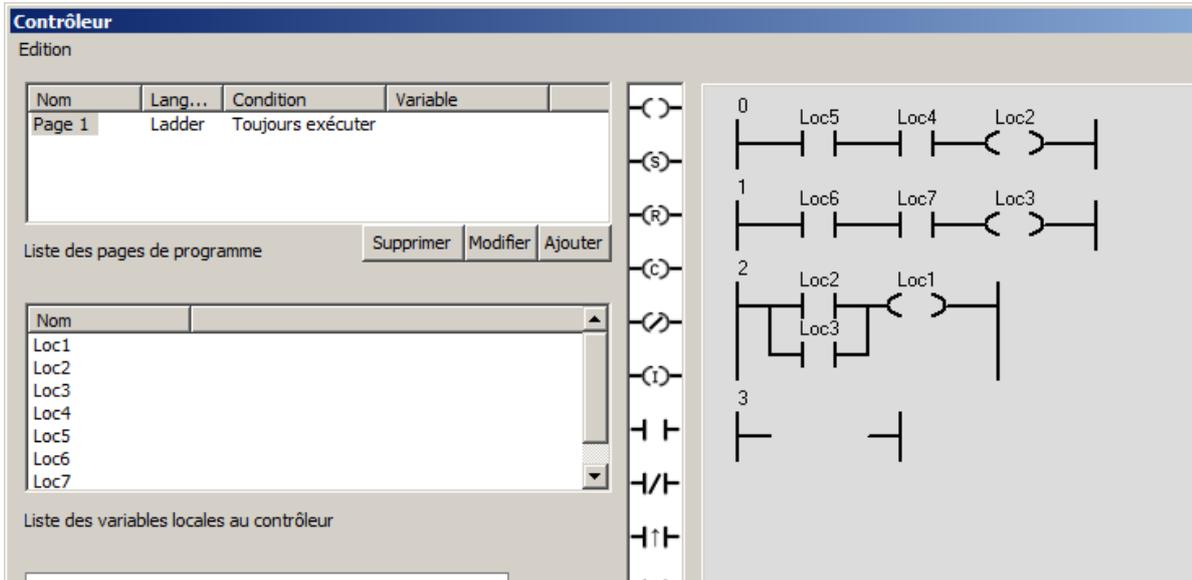


La création des programmes est réalisée en faisant glisser les éléments vers la zone de programmation. Le premier élément pouvant être déposé est un des éléments bobines disponibles. Une nouvelle variable locale est automatiquement associée à chaque nouvel élément créé.

Les contacts peuvent ensuite être déposés pour former un réseau.

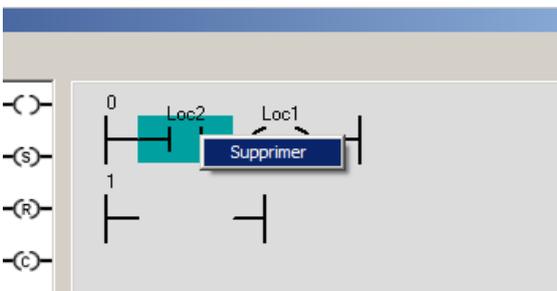


Seuls des réseaux simples peuvent être créés, si des réseaux plus complexes sont nécessaires, il doivent être décomposés en réseaux simples en utilisant des variables locales intermédiaires. Exemple :



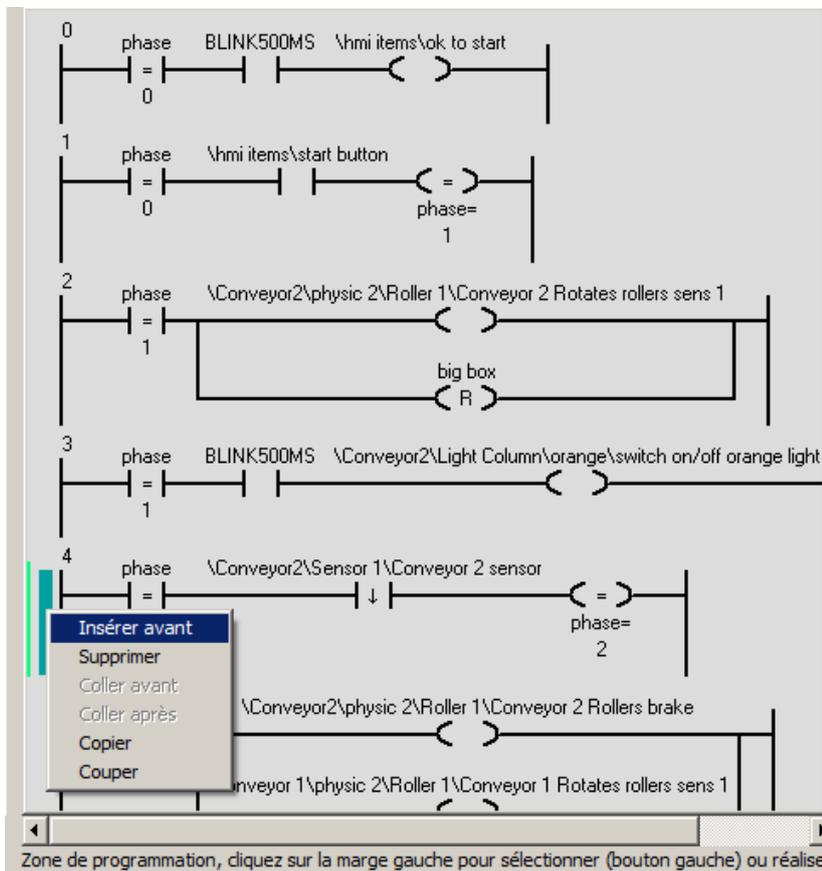
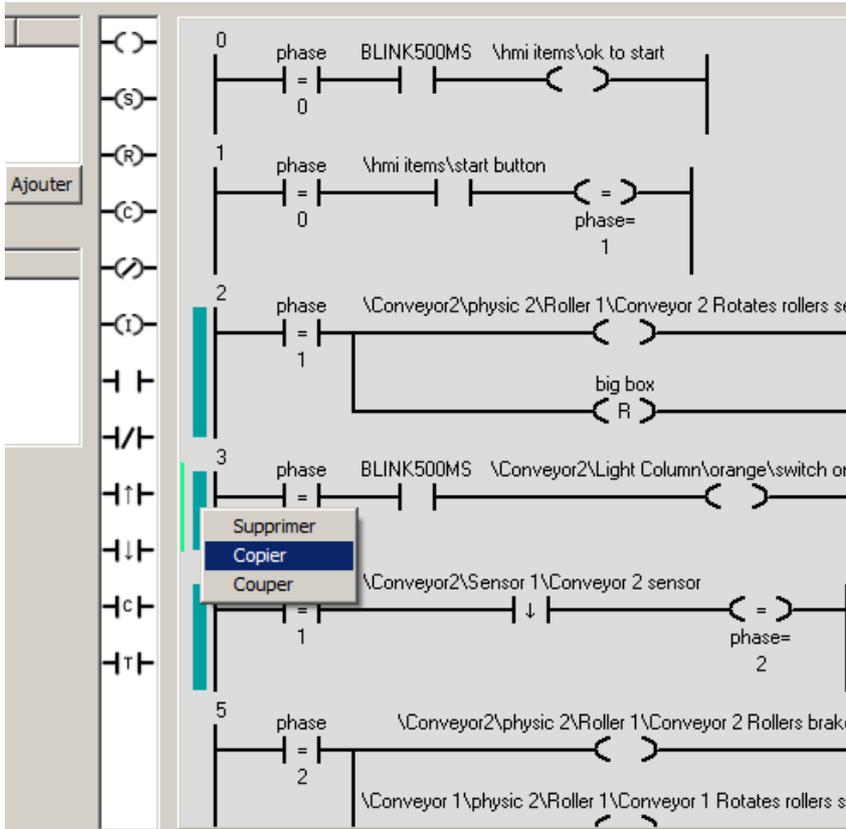
Fonctions d'édition

- pour supprimer un élément, sélectionner celui-ci (clic gauche) puis ouvrez le menu contextuel (clic droit sur l'élément) et sélectionnez "Supprimer" :

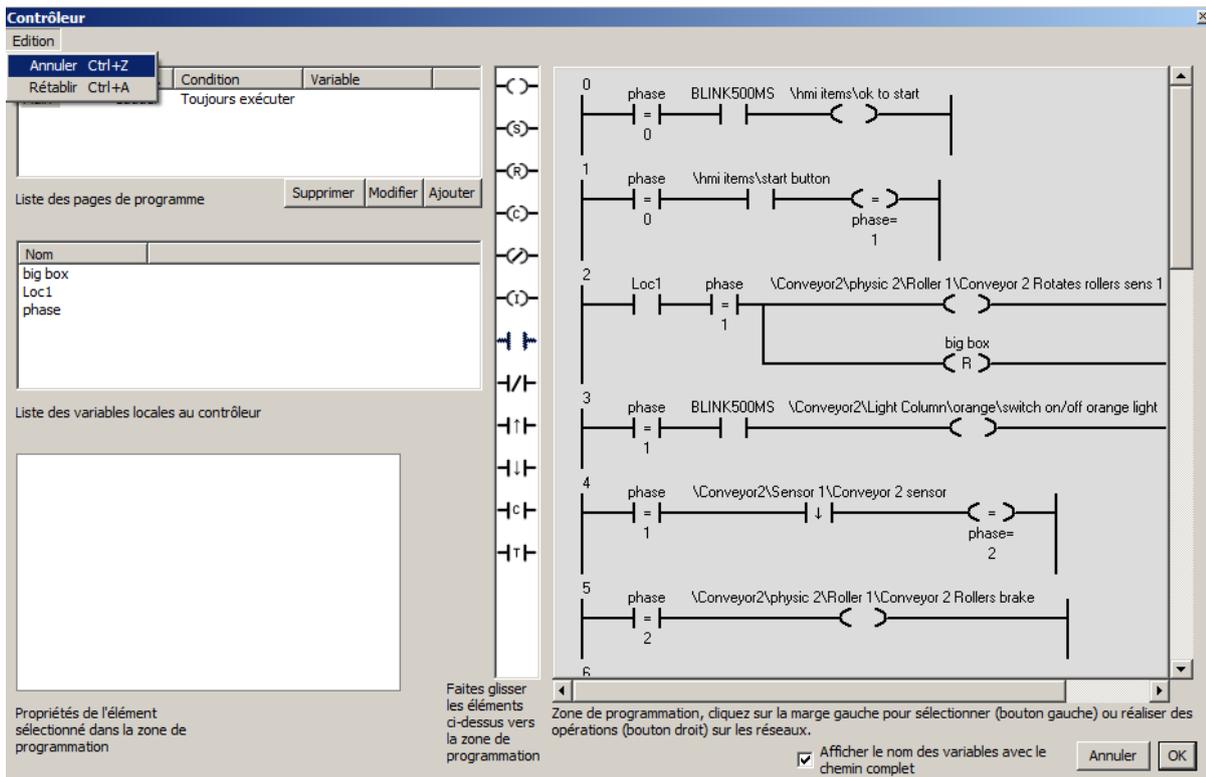


Le dernier élément de type bobine d'un réseau ne peut être supprimé que si aucun élément contact ne s'y trouve.

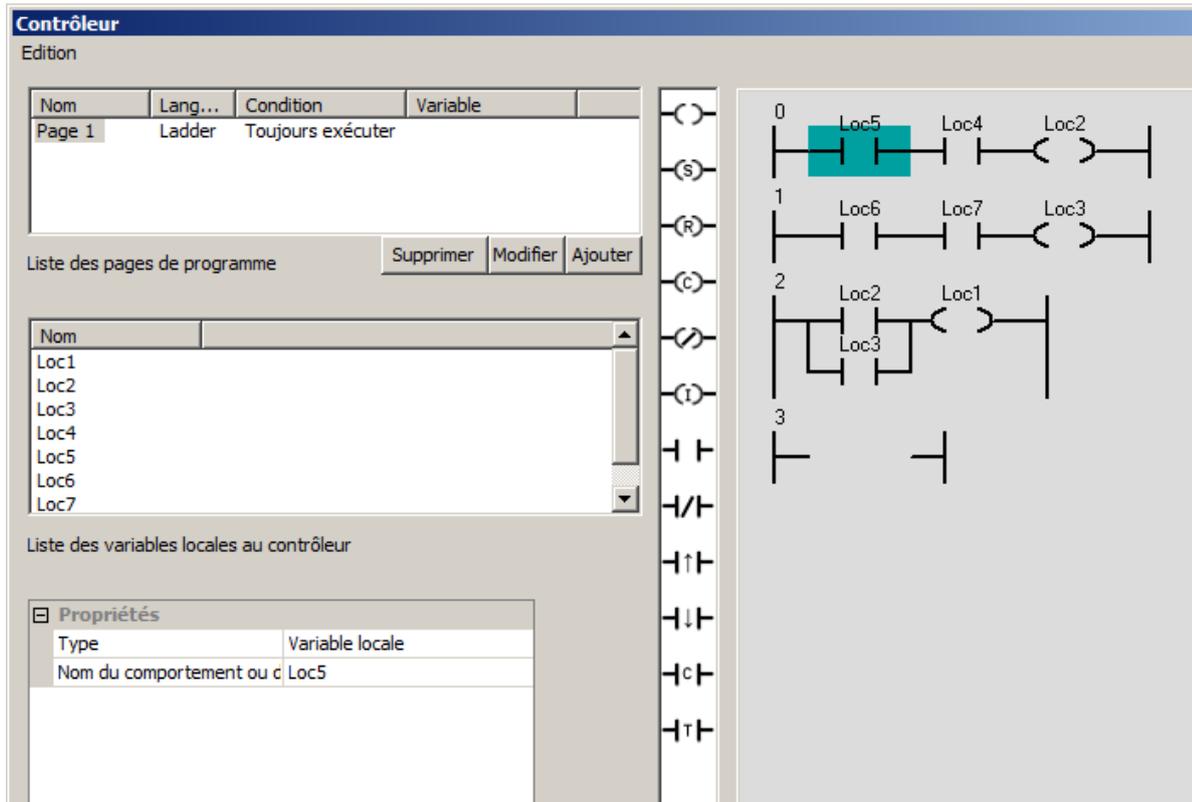
- pour supprimer / insérer / copier ou coller un réseau ou plusieurs réseaux, sélectionnez le ou les réseaux (clic gauche sur la marge gauche de ceux-ci) puis ouvrez le menu contextuel (clic droit sur la marge gauche) :



- pour annuler ou rétablir les dernières modifications apportées, utilisez le menu Edition de la fenêtre principale :



Lorsqu'un élément est sélectionné, on accède à ses propriétés en bas à gauche:



On peut ainsi sélectionner la variable associée à l'élément.

Liste des éléments



bobine : la variable associée est écrite avec la valeur 0 ou 1 suivant l'état défini par l'équation, formée par le réseau (0 si faux, 1 si vrai)



bobine inversée : pareil que la bobine mais la valeur écrite dans la variable associée est 1 si faux, 0 si vrai



bobine de mise à 1, la variable est mise à 1 si le réseau est vrai



bobine de mise à 0, la variable est mise à 0 si le réseau est vrai



bobine d'inversion, la variable est inversée si le réseau est vrai. Cette inversion est réalisée à chaque exécution de la page, l'utilisation du contact "front montant" permet de ne réaliser cette inversion qu'une fois



bobine calcul, réalise un calcul mathématique ou une affectation entre des variables ou des constantes et des variables si le réseau est vrai

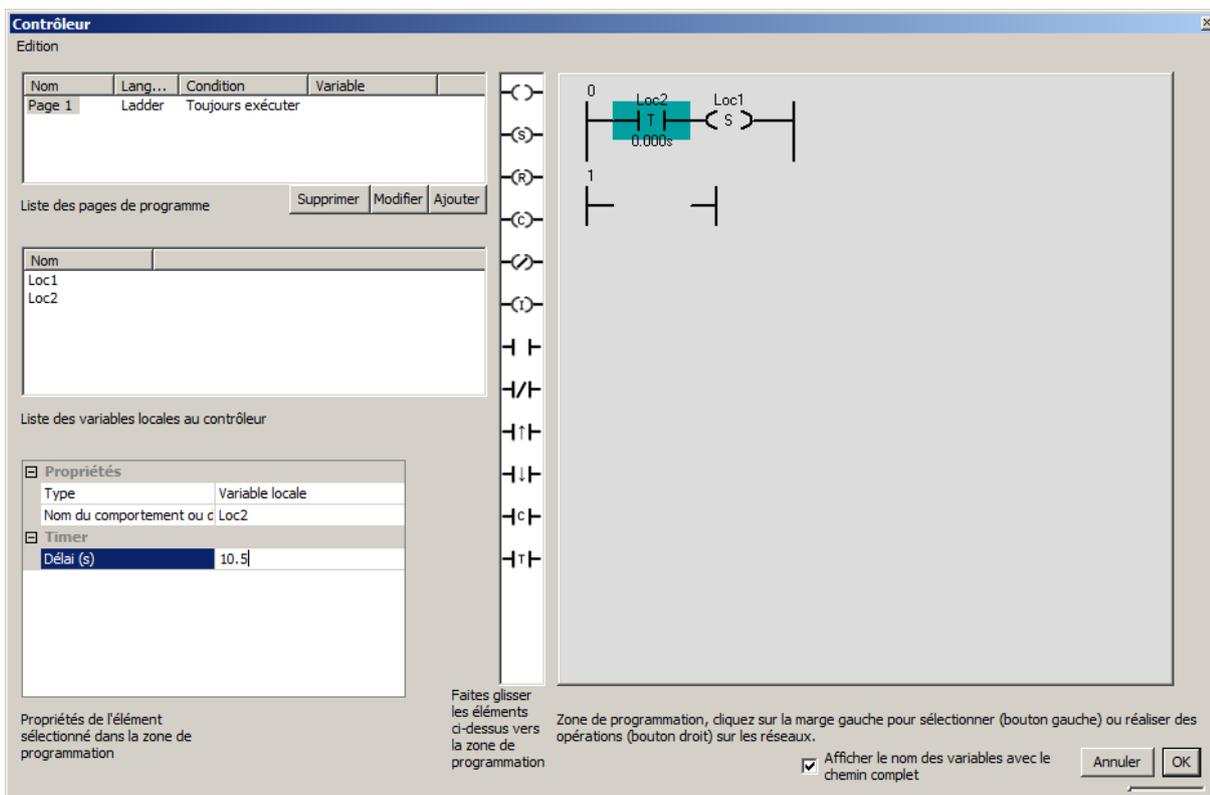
 Contact NO : vrai si l'état de la variable associée est vrai (différent de 0)

 Contact NF : vrai si l'état de la variable associé est faux (égal à 0)

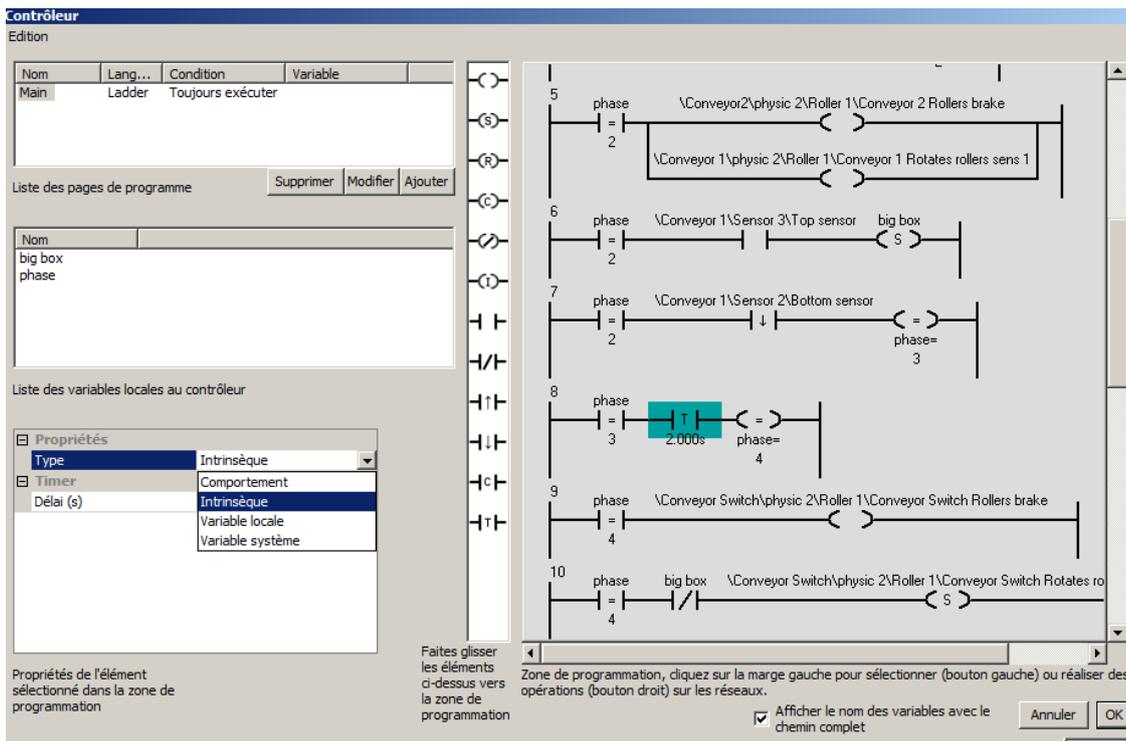
 Contact front montant : vrai pendant un cycle d'exécution si la variable associée passe de l'état faux à l'état vrai

 Contact front descendant : vrai pendant un cycle d'exécution si la variable associée passe de l'état vrai à l'état faux

 Contact temporisation : vrai après que la variable associée soit à l'état vrai depuis un certain temps. La durée exprimée en seconde est paramétrable dans les propriétés du contact :

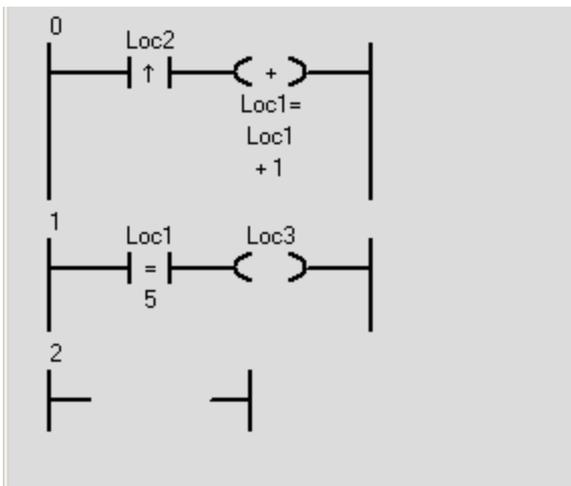


Pour ce contact, un pseudo type de variable "Intrinsèque" peut être sélectionné, ceci signifie que le signal temporisé est celui qui entre à gauche du contact :



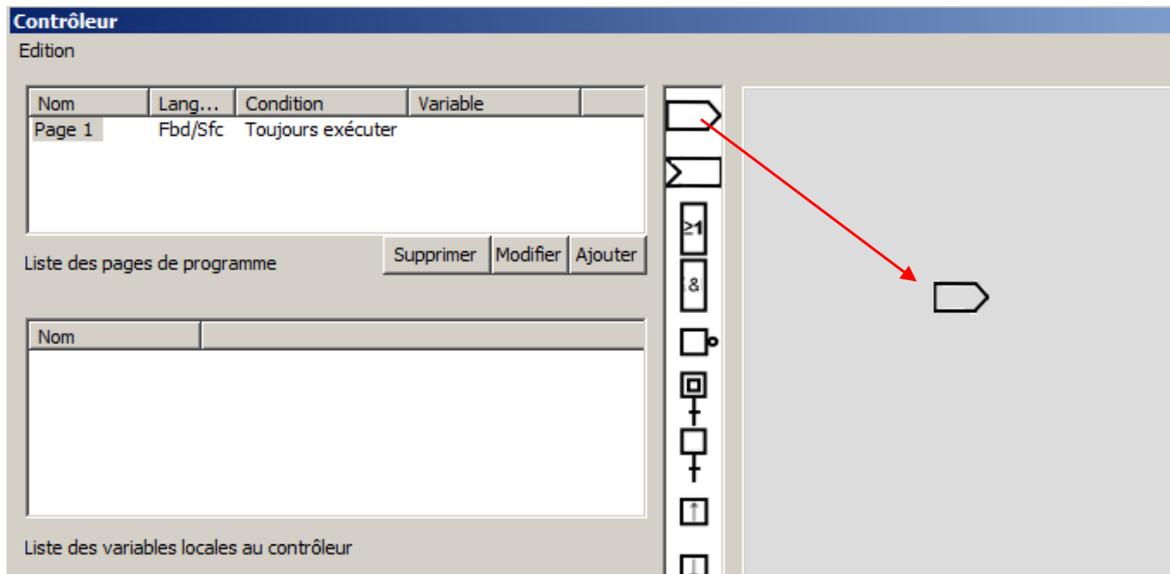
⌚ Contact comparaison : vrai si une comparaison numérique est vraie

Exemple (Activation de Loc3 après détection de 5 front montants sur Loc2. Loc1 est utilisé comme compteur):



Langage Fbd/Sfc

Création des programmes



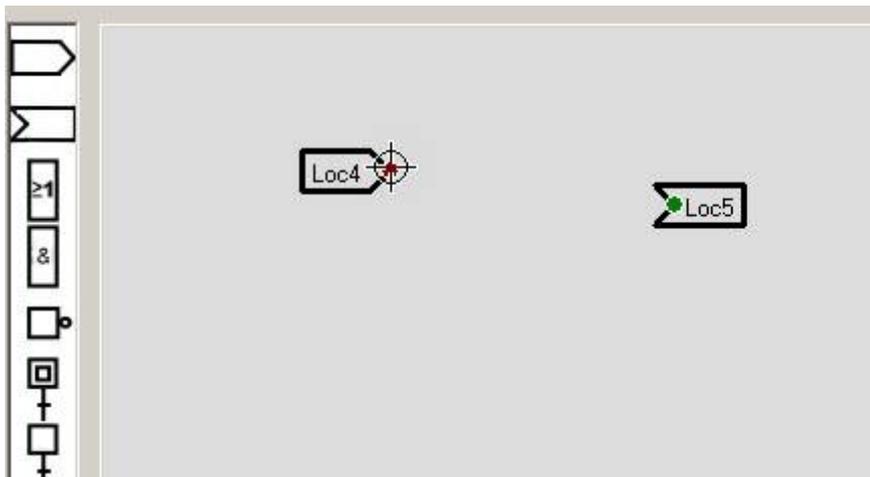
La création des programmes est réalisée en faisant glisser les éléments vers la zone de programmation. Une grille permet d'aligner les éléments pour une meilleure présentation.

Les éléments sont ensuite reliés en créant des liens. Les connexions des objets sont matérialisés par des ronds colorés :

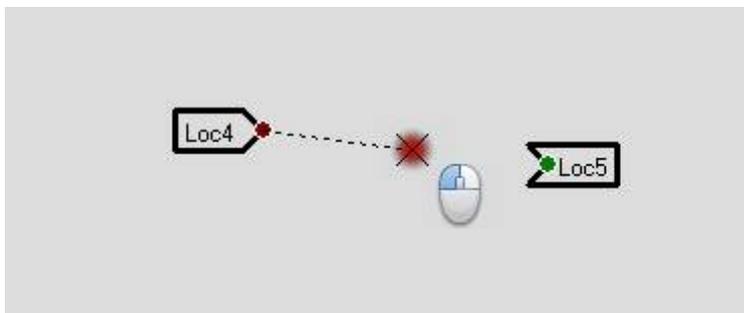
- rond rouge : connexion de sortie, peut être connectée à une ou plusieurs connexions d'entrée,
- rond vert : connexion d'entrée, peut être connectée à une connexion de sortie,
- rond bleu connexion SFC. peut être connectée à une autre connexion SFC.

Tracé des liaisons :

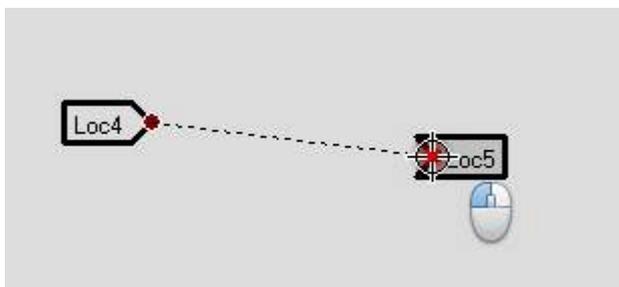
- 1- déplacez le curseur au dessus d'une connexion (le curseur se change en cible) et enfoncez le bouton gauche de la souris (ne relâchez pas le bouton gauche) :



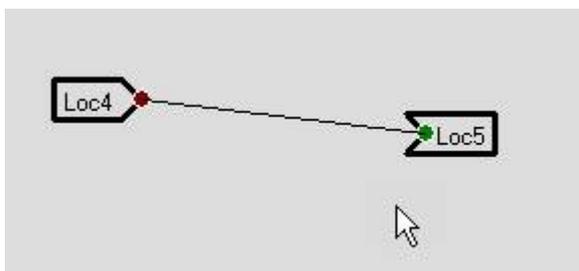
2- déplacez le curseur (le lien en cours de création est matérialisé) :



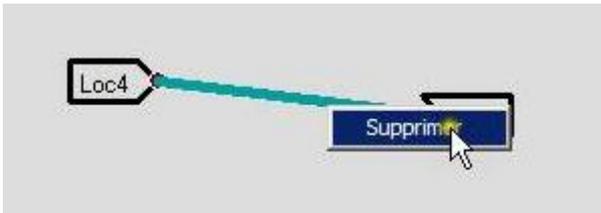
3- déplacez le curseur au dessus de la connexion de destination :



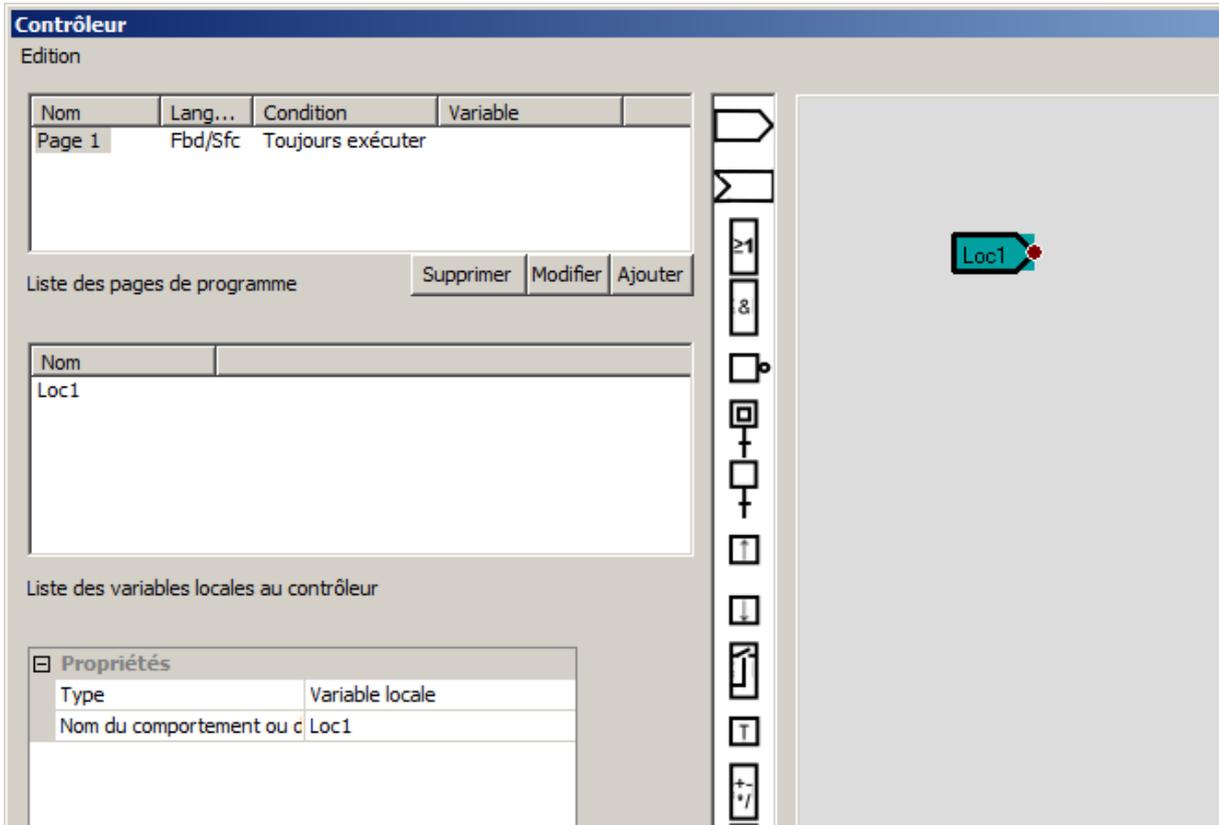
4- relâchez le bouton gauche de la souris :



Suppression d'une liaison : cliquez avec le bouton droit de la souris sur le lien et choisissez "Supprimer" :



Sélection d'un élément : un clic gauche sur un élément le sélectionne. Lorsqu'un élément est sélectionné, ses propriétés sont accessibles en bas à gauche :

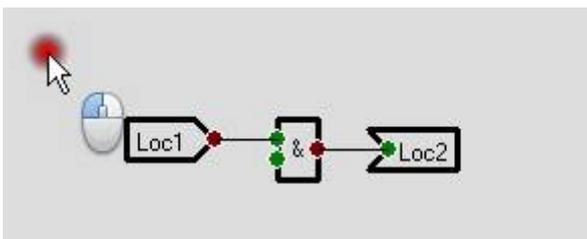


Si la touche Shift est enfoncée, l'élément est ajoutée à une éventuelle sélection existante.

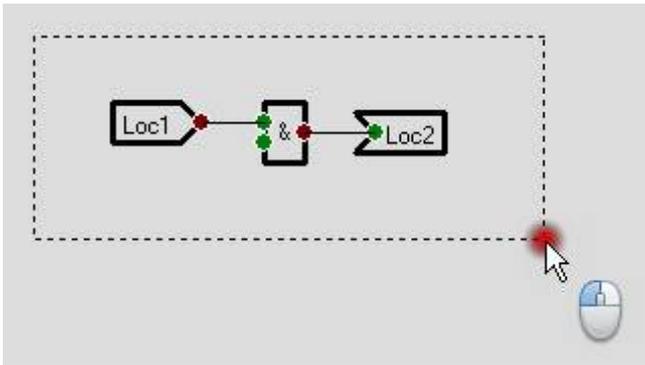
Si la touche Ctrl est enfoncée, l'élément est ajouté ou retiré de la sélection.

Sélection d'un ensemble d'éléments :

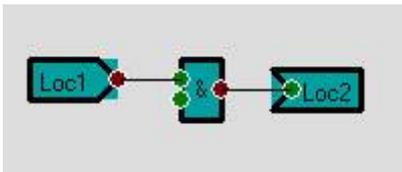
1- enfoncez le bouton gauche de la souris lorsque le curseur se trouve sur un emplacement vide de la zone de programmation (laissez le bouton enfoncé) :



2- déplacez la souris pour sélectionner les éléments :



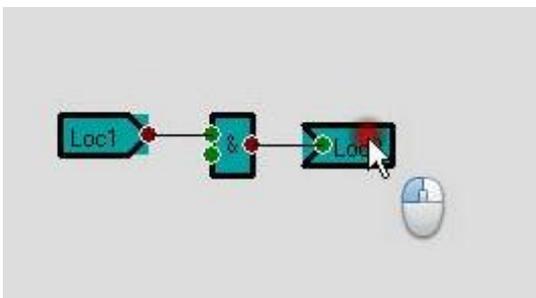
3- relâchez le bouton gauche de la souris :



Si la touche Shift est enfoncée, les éléments sont ajoutés à une éventuelle sélection existante.

Déplacer un ou plusieurs éléments :

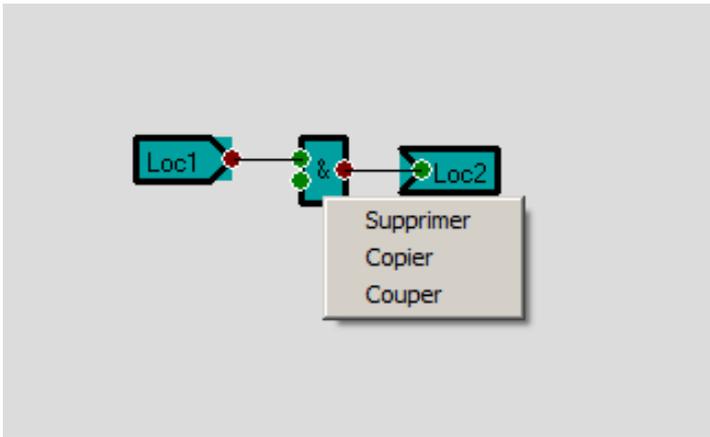
1- enfoncez le bouton gauche de la souris lorsque le curseur se trouve sur un des éléments sélectionnés (laissez le bouton enfoncé) :



2- déplacez le ou les objets,

3- relâchez le bouton gauche de la souris.

Suppression ou copie d'un ou plusieurs éléments : cliquez avec le bouton droit de la souris sur un des éléments puis choisissez dans le menu :



Les liaisons connectées à un élément supprimé sont également supprimées.

Les liaisons connectées à deux éléments copiés sont également copiées.

Collage d'un élément : cliquez avec le bouton droit de la souris sur un emplacement vide de la zone de programmation et choisissez "Coller" dans le menu.

Liste des éléments



entrée : signal d'entrée associé à une variable (lecture de l'état d'une variable) ou une constante



sortie : signal de sortie associé à une variable (écriture de l'état d'une variable)



fonction logique non : le signal de sortie correspond au signal d'entrée complémenté



fonction logique ET : le signal de sortie correspond à l'équation booléenne signal d'entrée 1 ET signal d'entrée 2



fonction logique OU : le signal de sortie correspond à l'équation booléenne signal d'entrée 1 OU signal d'entrée 2



front montant : le signal de sortie est vrai pendant un cycle d'exécution du programme quand le signal d'entrée passe de l'état faux à l'état vrai



front descendant : le signal de sortie est vrai pendant un cycle d'exécution du programme quand le signal d'entrée passe de l'état vrai à l'état faux



timer : le signal de sortie est activé après un délai. Ce délai exprimé, en secondes est défini dans les propriétés du bloc.



affectation : si l'entrée du bas est vraie, l'entrée du haut est recopiée sur la sortie, sinon, la sortie conserve son état



calcul : réalise un calcul mathématique entre les valeurs des deux entrées et place le résultat sur la sortie



comparaison : compare les valeurs des deux entrées et place le résultat (vrai ou faux) sur la sortie



Etape Grafcet initiale + transition

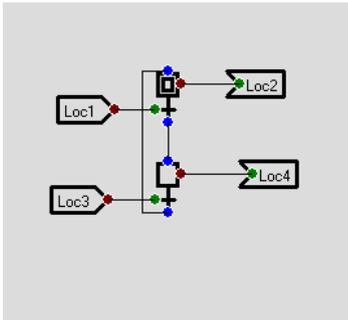


Etape initiale + transition



Encapsulation

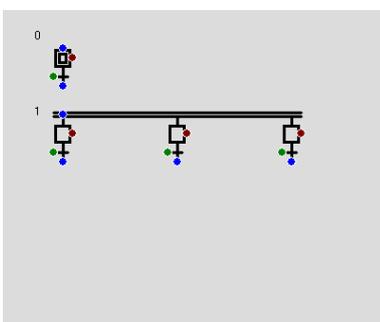
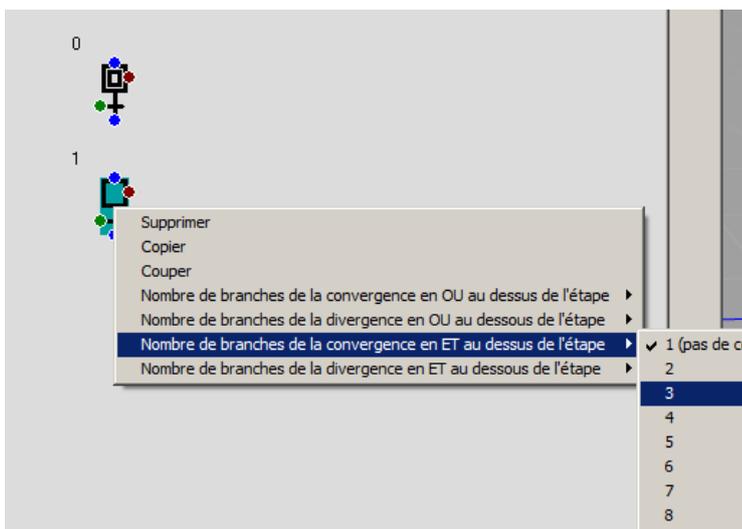
Grafcet



Les blocs Etape + transition possèdent une connexion verte pour connecter la condition de la transition et une connexion rouge pour connecter une éventuelle action.

Divergences / convergences en ET et en OU

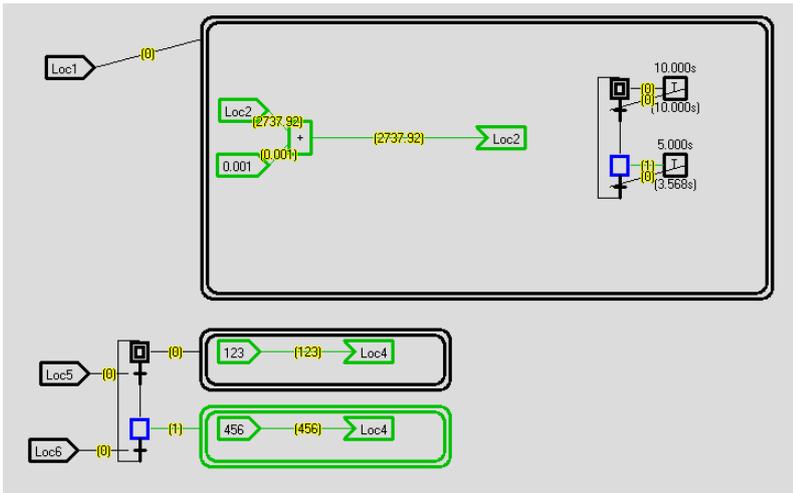
En cliquant avec le bouton droit de la souris sur une étape sélectionnée (l'étape doit être la seule sélectionnée), un menu contextuel permet de définir le nombre de branches pour les divergences ou convergences en ET ou en OU :



Encapsulation

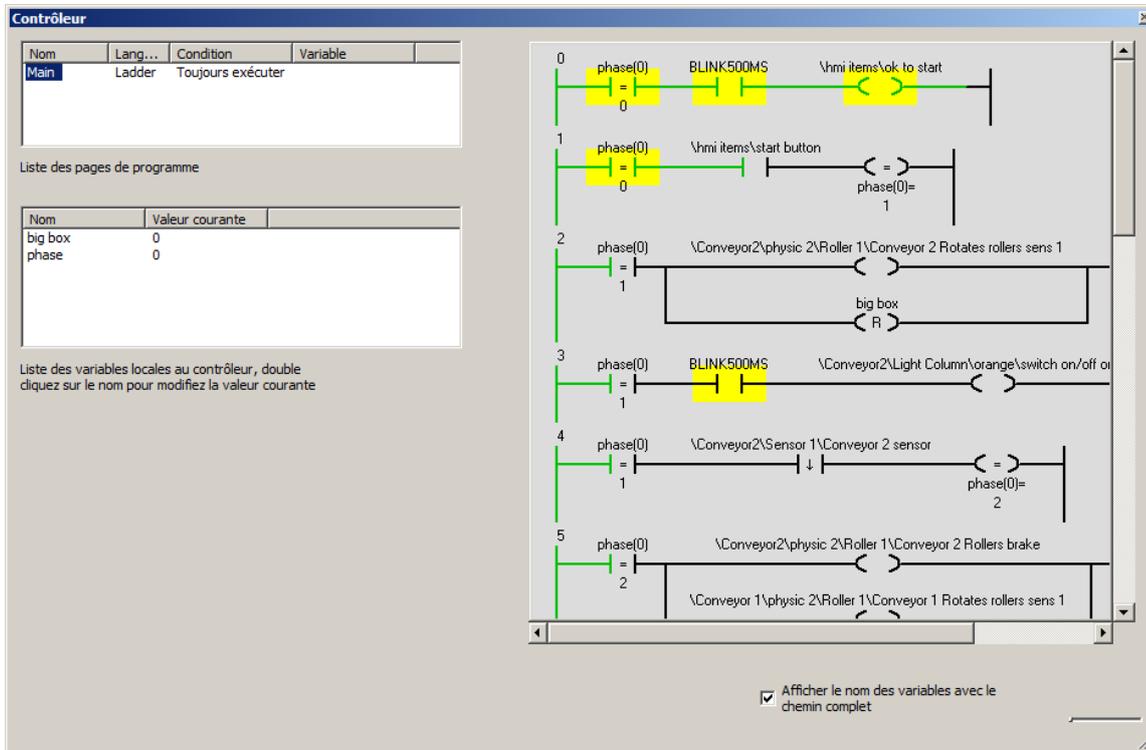
Le bloc  permet d'encapsuler une partie d'un programme. Ce bloc possède une entrée dont l'état conditionne l'exécution des blocs se trouvant dans l'encapsulation. La taille de ce bloc est modifiable en cliquant avec le bouton gauche de la souris sur les carrés noirs qui l'entourent : laisser le bouton gauche de la souris enfoncé, déplacer le curseur pour modifier la taille puis relâcher le bouton gauche. Si un ou plusieurs Graficets ou blocs se trouvent dans l'encapsulation, leurs évolutions sont figées si l'entrée est fautive.

Exemple :

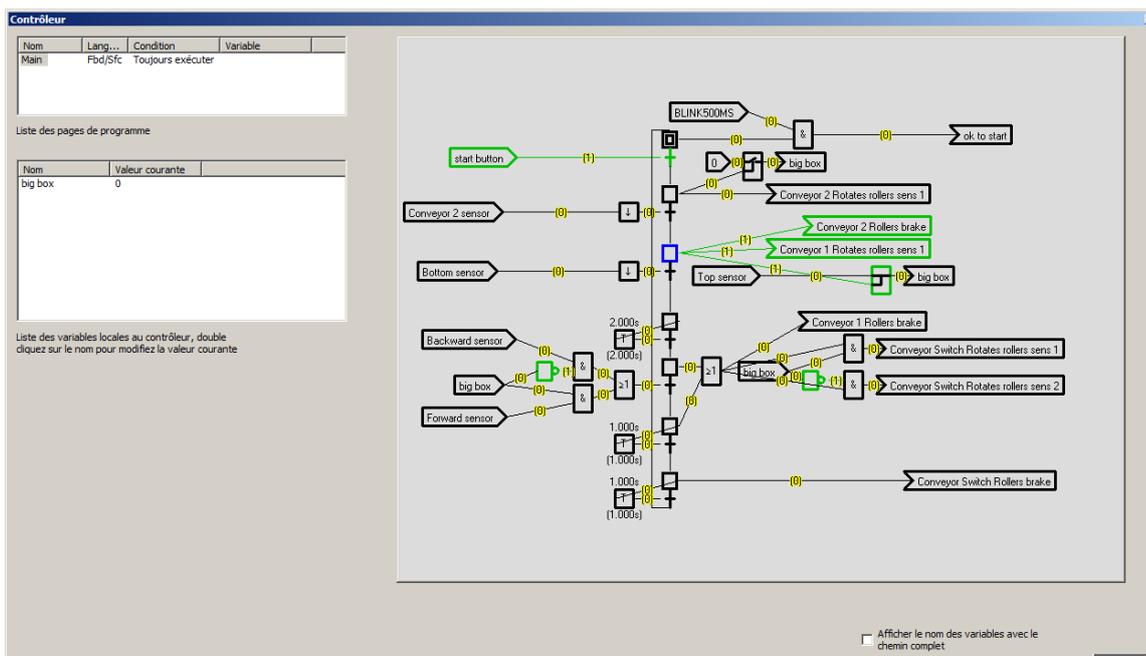


Mode RUN

En mode RUN, l'état des programmes sont affichés ainsi que l'état des variables locales dans la liste de celles-ci :

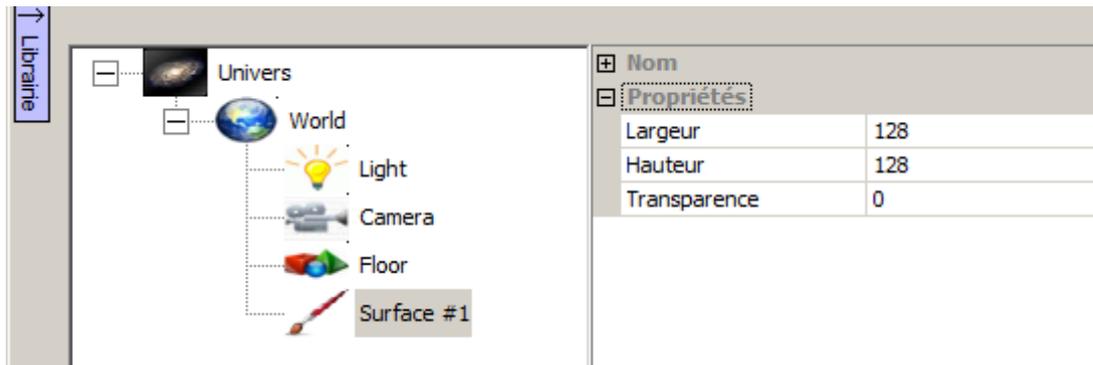


Un double clic sur le nom d'une variable locale permet de modifier son état. Un clic droit inverse l'état (considérant la variable comme une variable booléenne).



L'accès au pages de programme des contrôleur est possible dans la fenêtre de configuration ou le menu "Simulation / Mise au point / Programme simulation" si l'option "Menu simulation" est activée dans les propriétés du contrôleur.

Propriétés détaillées des surfaces



Les paramètres Largeur / Hauteur définissent le nombre de pixels de la surface. Plus la résolution est élevée, plus le rendu de la surface sera précis. Les performances de rendu peuvent être dégradées si ces dimensions sont élevées. Le paramètre transparence peut varier de 0 (invisible) à 1 (opaque). Par exemple 0.5 définira une transparence de 50%.

Lien entre Sprite3d et surface

La primitive prédéfini "Plan" peut être utilisé pour plaquer facilement une texture sur une surface. Pour les autres types d'objet 3D les coordonnées UV devront être réglées avec un logiciel approprié, Unwrap3D est par exemple un outils permettant de réaliser cette tâche. Il est également possible d'ajouter des modificateurs de géométrie UV à la suite du nom de la texture (paramètre Dessin/Fichier texture des Sprites 3d). Les paramètres suivants sont utilisables : SCALEX=échelle sur X, SCALEY=échelle sur Y, POSX=position sur X, POSY=position sur Y, ROT=rotation. Si plusieurs paramètres sont utilisées, il doivent être séparés par des virgules.

Le concept de surface permet de créer des textures 2D dynamiques composées d'une ou plusieurs couches. Chaque couche peut être un des éléments suivants :

- une couleur,
- un fichier bitmap,
- un fichier vidéo (la lecture - stop, pause, retour, etc.- peut être contrôlée),
- un IHM (les éléments IHM sont affiché sur la surface, ils sont également réactifs aux actions de l'utilisateur : déplacement du curseur et clics sur la surface,
- un folio de simulation AUTOMSIM,
- une caméra virtuelle (une caméra de VIRTUAL UNIVERSE PRO),
- une caméra réelle (une webcam connectée au PC).

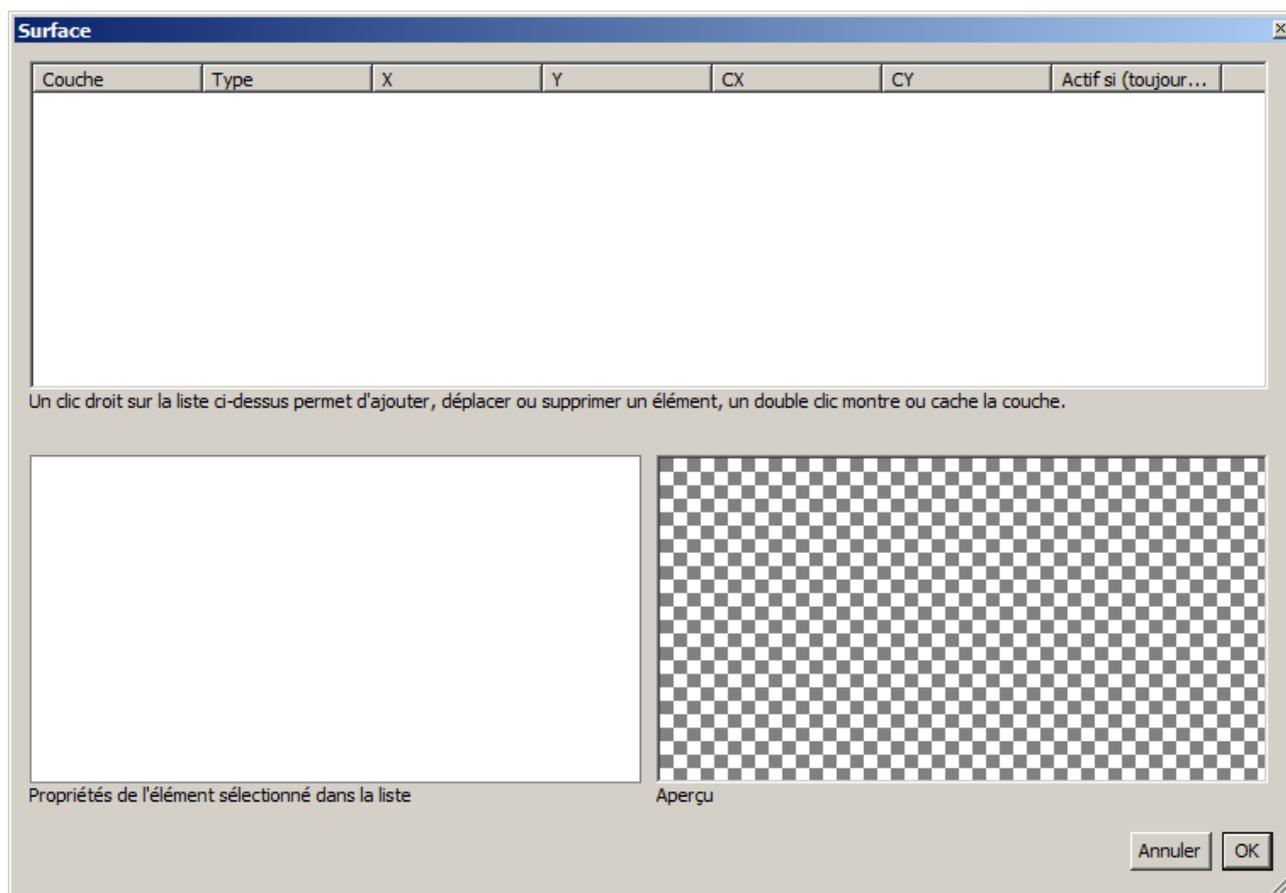
Des couches permettent de détecter le survol ou les clics souris peuvent également être créées.

Ajout et paramétrage d'une surface

Une surface peut être enfant du monde ou d'un Sprite 3D, dans ce dernier cas, la surface peut être exportée avec un objet. Pour créer une surface, faire un clic droit sur le monde ou un Sprite 3d dans l'arborescences et choisissez "Ajoutez>Surface".

Pour paramétrer une surface, double cliquez sur un élément "Surface" de l'arborescence en mode STOP.

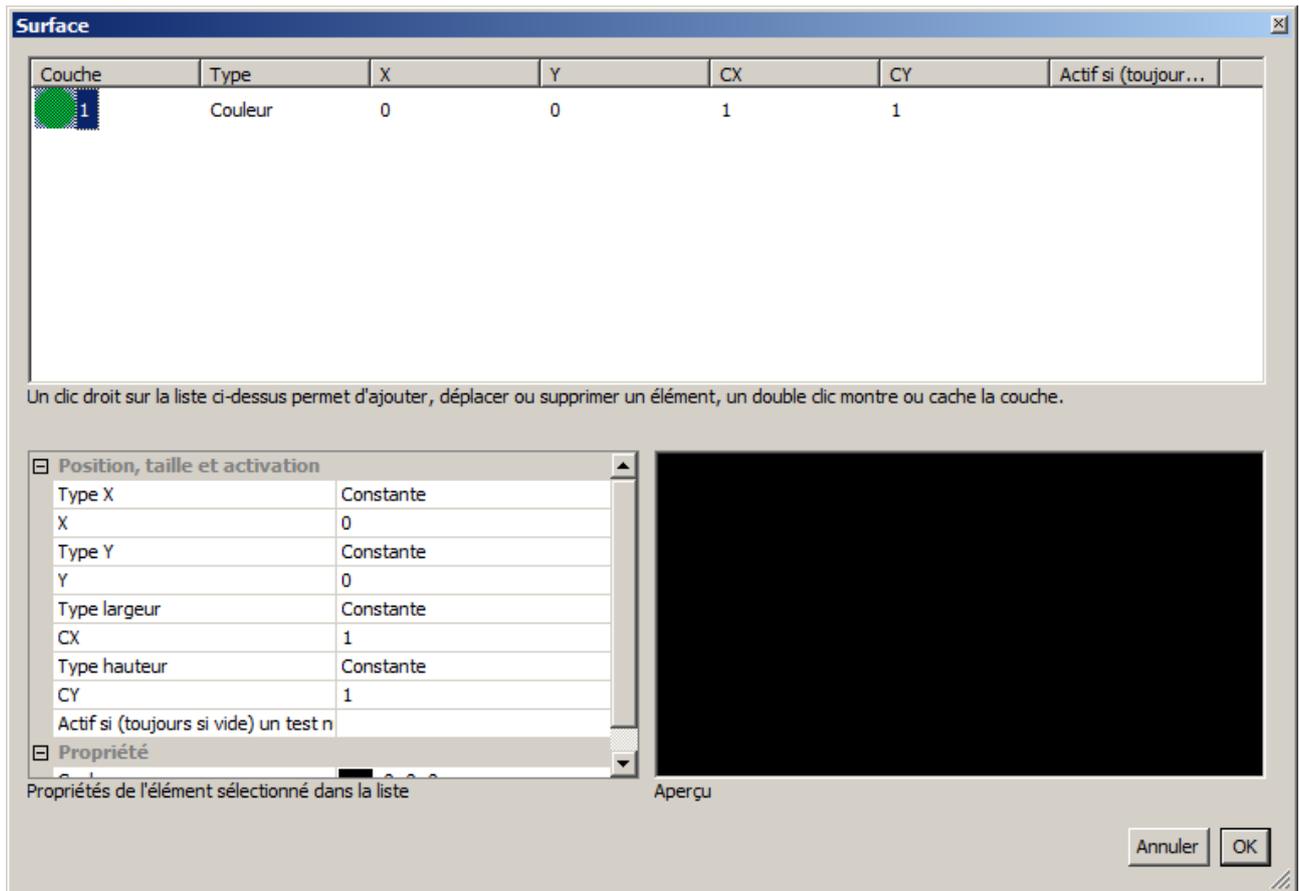
La fenêtre suivante s'ouvre :



Pour ajouter une couche, cliquez avec le bouton droit de la souris sur la liste (partie haute) et choisissez "Ajouter" puis le type de la couche à ajouter.

Pour supprimer une couche, un clic droit sur celle-ci permet d'accéder à une option "Supprimer".

Paramètres communs



Les paramètres X/Y/CX/CY/Actif si sont communs à l'ensemble des types de couches. Ils permettent de définir la position d'affichage de la couche, sa taille et sa visibilité.

X/Y/CX/CY sont toujours exprimées en valeurs réelles entre 0 et 1.

Pour X, 0 représente la gauche de la surface, 1, la droite, pour Y 0 le haut, 1 le bas. Pour CX et CY 0 représente une taille nulle, 1 une taille normale (couvrant toute la surface), 0.5 représentera par exemple une taille couvrant la moitié de la surface. Ces 4 paramètres peuvent être des constantes ou la valeur d'un comportement.

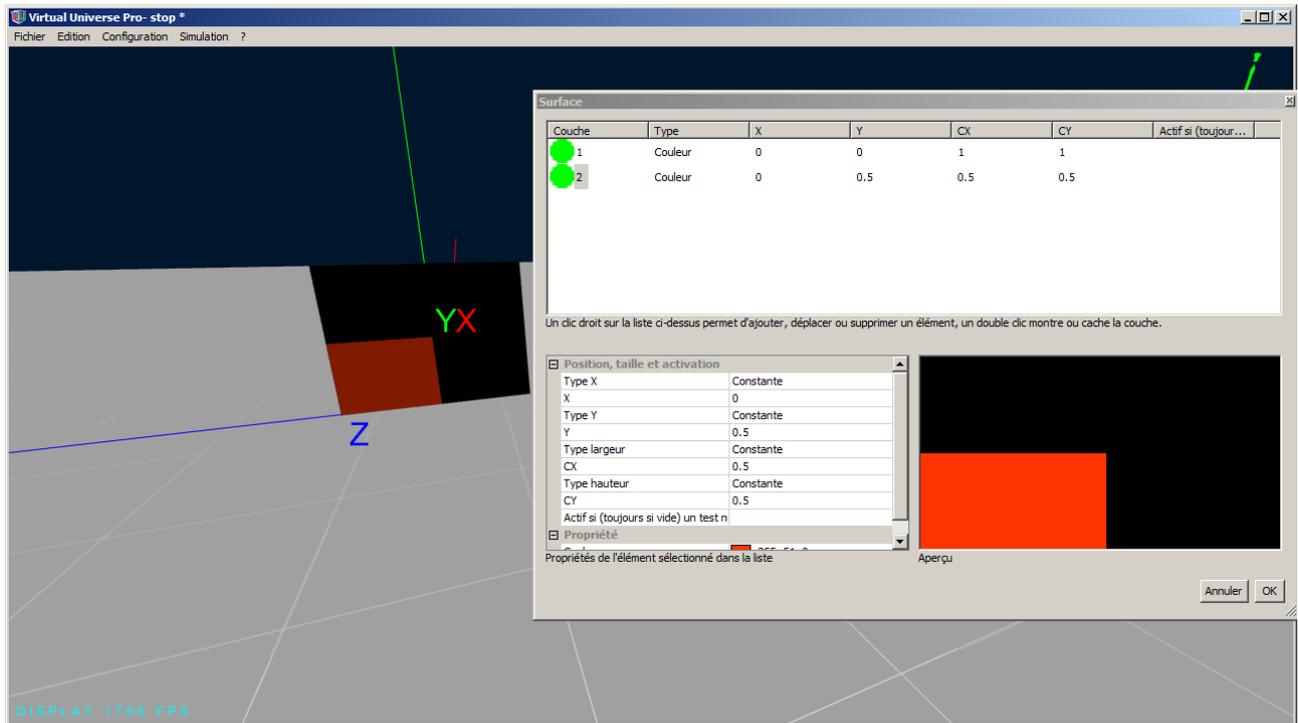
Le paramètre "Actif si" si non vide définit le nom d'un comportement qui conditionne l'affichage de la couche.

Ordre des couches

L'ordre des couches est important, les couches sont "empilées" par numéro d'ordre croissant". La couche 1 se trouve donc en dessous des éventuelles autres couches, la couche 2 au dessus de la couche 1 et en dessous des éventuelles autres couches, etc.

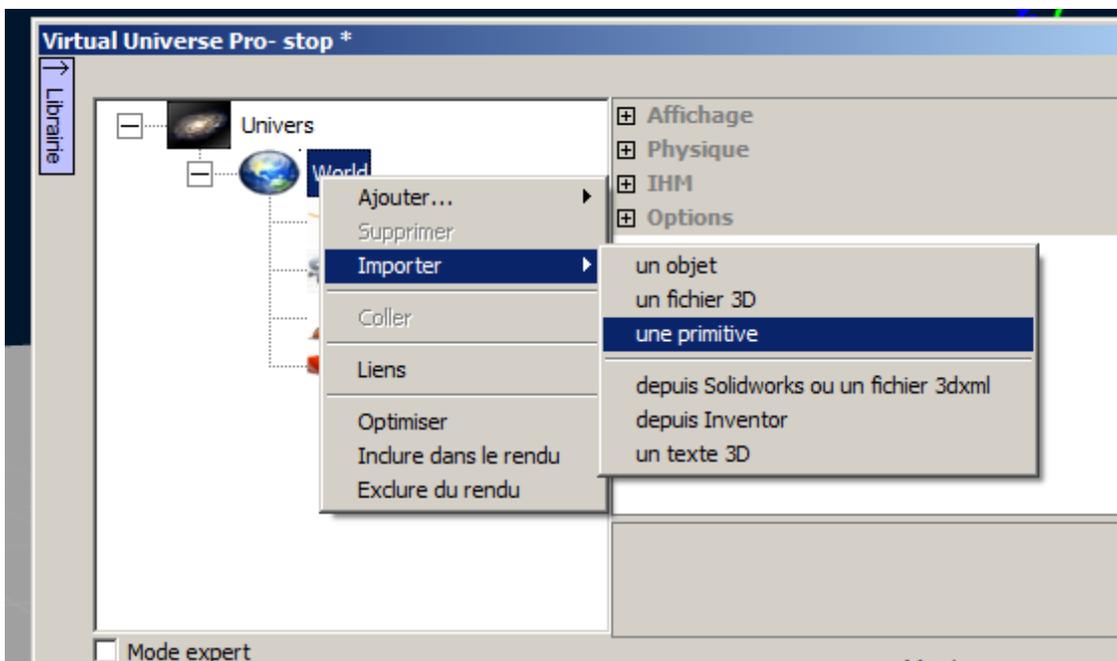
Pour changer l'ordre d'une couche, un clic droit sur celle-ci permet d'accéder à des options "Monter" et "Descendre".

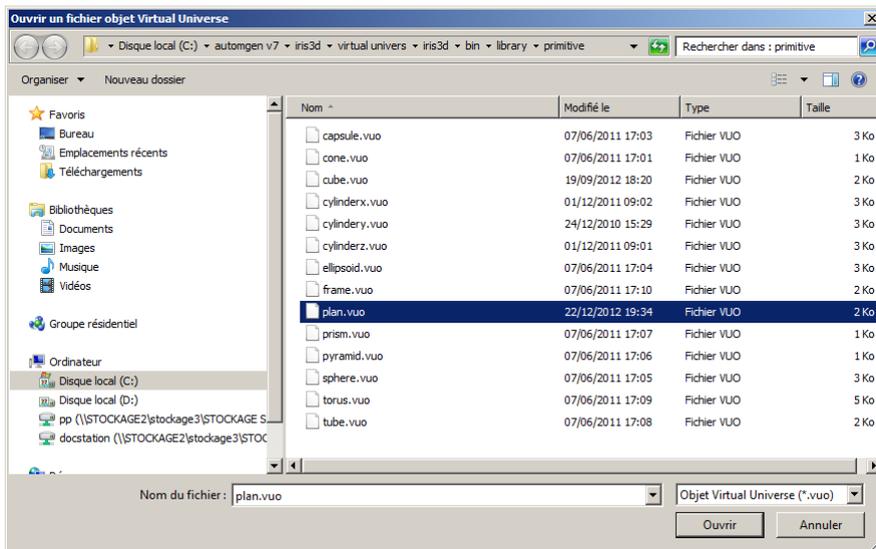
Exemple d'une surface composée de 2 couleurs :



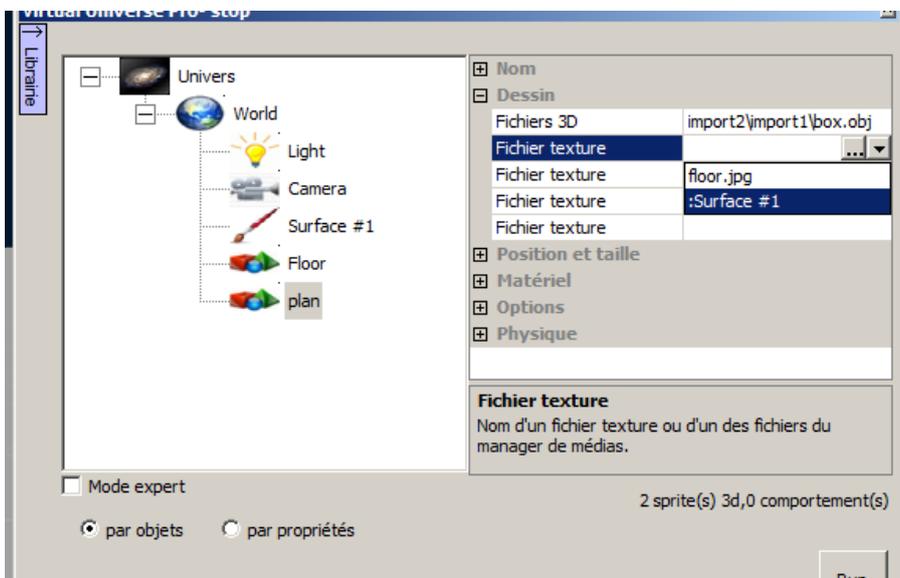
L'objet affiché dans la zone de rendu 3D est obtenu de la façon suivante :

1- importation d'une primitive "Plan" :



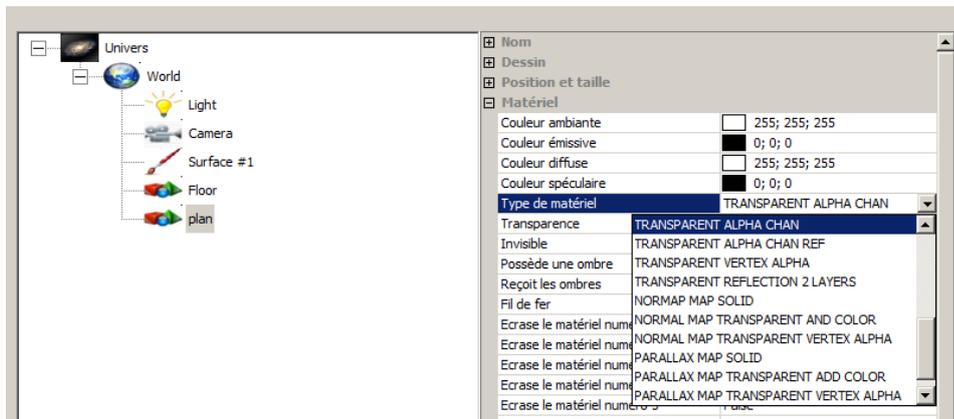


2- sélection de la surface dans les propriétés de texture de l'objet importé :

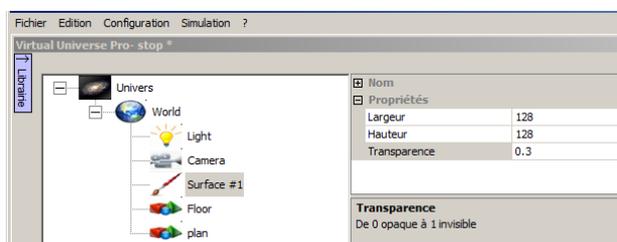


Remarque : si une surface transparente est souhaitée, procéder comme ceci :

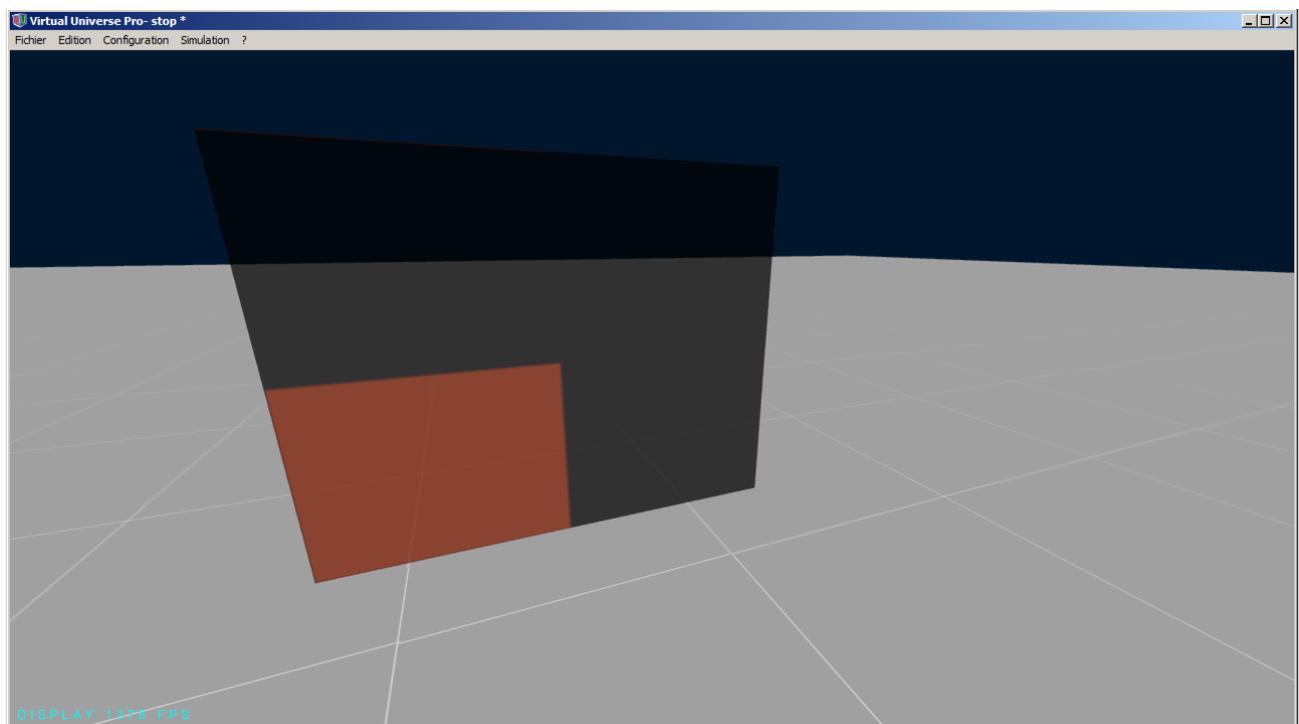
- dans les propriétés de l'objet, sélectionnez "TRANSPARENT ALPHA CHAN",



- réglez la transparence dans les propriétés de la surface :



Le résultat est le suivant :



Type couleur

Ce type de couche permet de définir une zone rectangulaire de couleur.

Paramètre spécifique : la couleur.

Type bitmap

Ce type de couche permet de définir l'affichage du contenu d'un fichier bitmap. Les formats utilisables pour le fichier bitmap sont (BMP, JPG, PNG, et TIFF). Attention au format compressé (JPG par exemple) qui peuvent produire des surface de mauvaise qualité. Le format PNG est un bon compromis entre la taille du fichier et la qualité de l'image. Le fichier bitmap peut être placé dans les médias du projet.

Paramètres spécifiques : nom du fichier bitmap, mise à l'échelle ou non du bitmap (couvre ou non toute la surface), couleur transparente (les pixels possédant la couleur sélectionnée seront transparents).

Type fichier vidéo

Ce type de couche permet d'afficher une vidéo sur la surface. Les formats vidéos avi, mp4, et mpg sont supportés. Au delà de ces extensions, il convient de vérifier le fonctionnement en fonction du codec utilisé dans chaque fichier. Le fichier vidéo peut être placé dans les médias du projet.

En mode STOP, la vidéo est jouée en boucle. En mode RUN, la vidéo est jouée et s'arrête en fin de fichier.

Paramètres spécifiques : nom du fichier vidéo, mise à l'échelle ou non de la vidéo (couvre ou non toute la surface), nom du comportement recevant la position de lecture, nom du comportement définissant la position d'écriture.

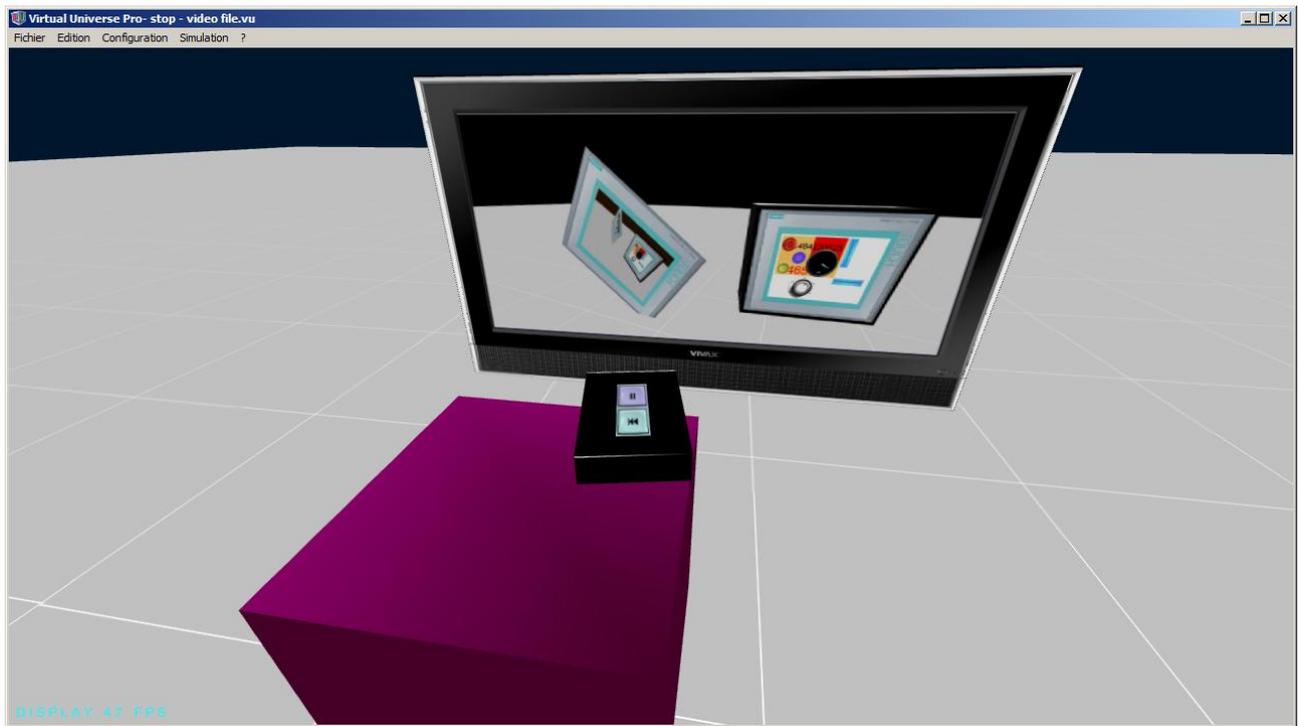
Gérer la lecture

Le comportement "Résultat position de lecture" reçoit en ms la position de la lecture courante.

Le comportement "Ecriture de la position de lecture" permet de définir la position, si sa valeur est différente de -1, la position de lecture (en ms) est défini par cette valeur puis remise automatiquement à la valeur -1.

Avec ces deux comportements, il est par exemple possible de "rembobiner" la vidéo avec une équation du type "si la position de lecture est égale à x ms, alors écrire la position de lecture à 0".

Un exemple nommé "video files" se trouvant dans le sous-répertoire "Technical tips\Surfaces" du répertoire des exemples illustre tout ceci :



Type IHM

Ce type de couche permet de définir sur la surface un ou plusieurs éléments IHM (boutons, voyants, etc.). Ces éléments seront affichés et interactifs.

Paramètres spécifiques : nom de l'IHM (tel que défini dans le projet).

L'IHM sélectionné doit avoir été paramétré comme suit :

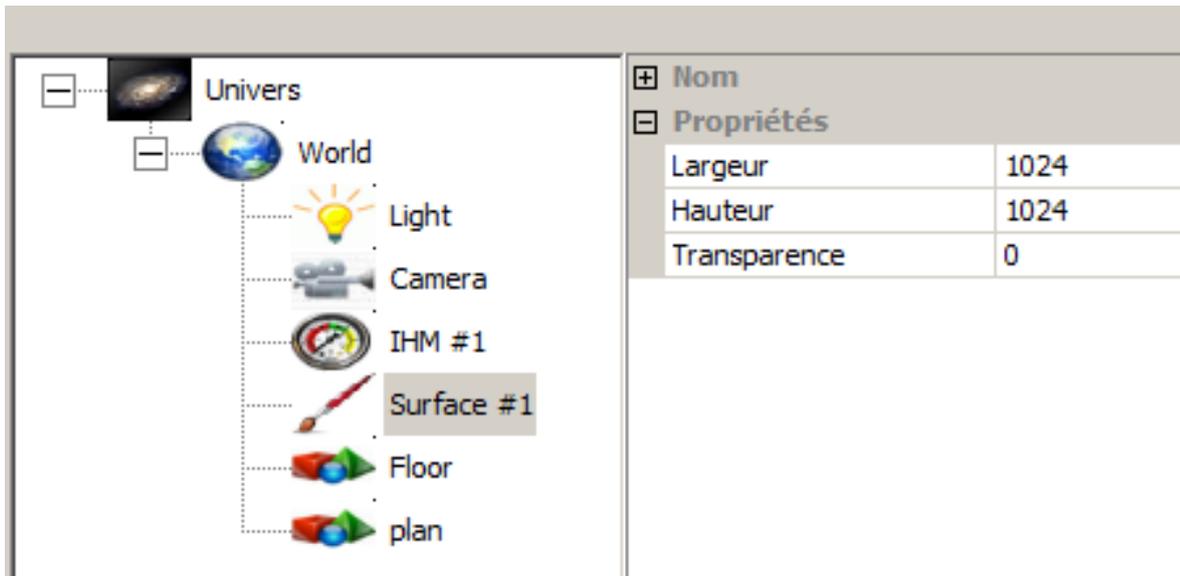
Rendu vers une texture : Oui

Largeur de la texture : la même que celle de la surface.

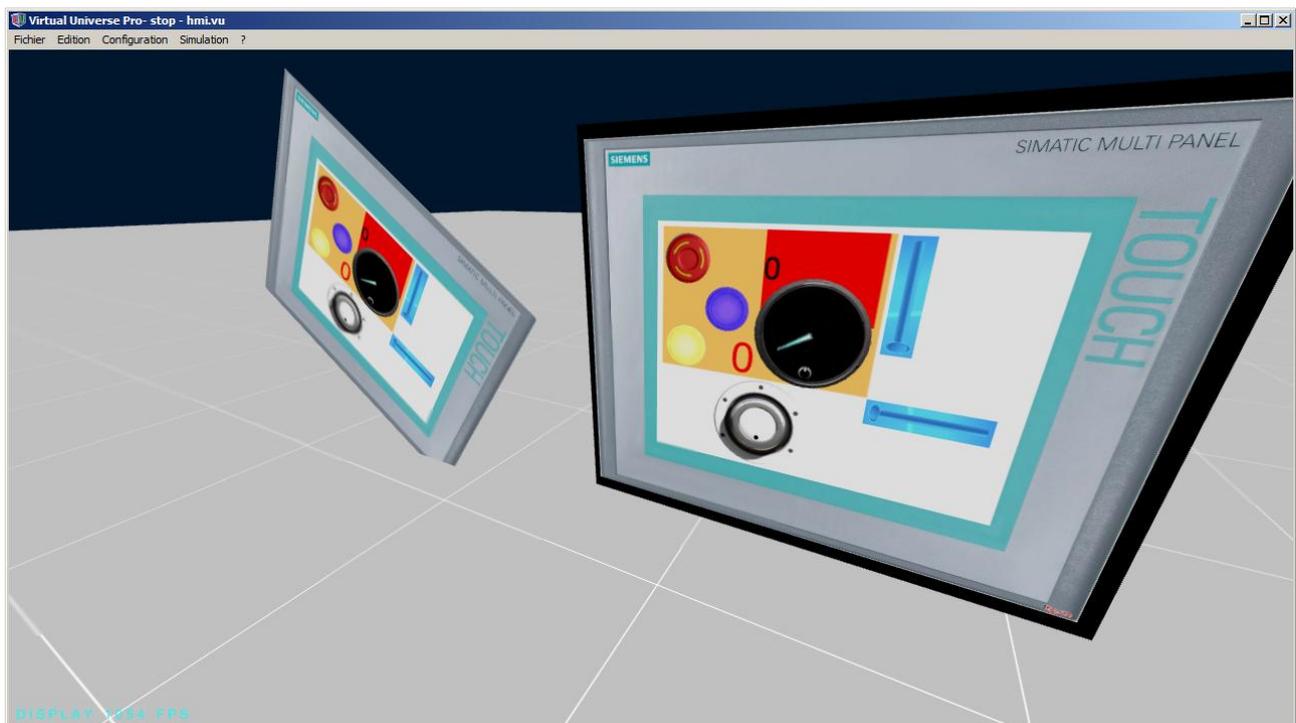
Hauteur de la texture : la même que celle de la surface.

Exemple :

Propriétés	
Transparence	0
Condition de visibilité	
Rendu vers une texture	True
Largeur de la texture	1024
Hauteur de la texture	1024



Un exemple nommé "hmi" se trouvant dans le sous-répertoire "Technical tips\Surfaces" du répertoire des exemples illustre tout ceci :



Type Folio AUTOMSIM

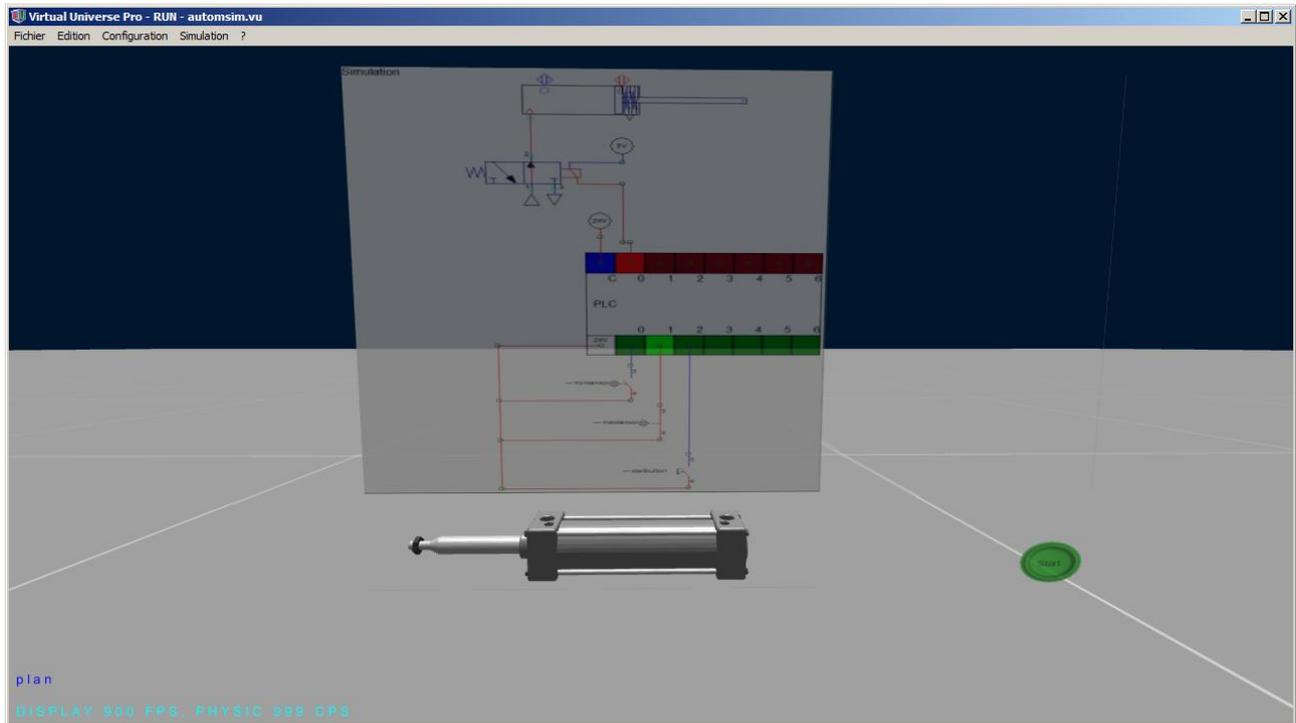
Ce type de couche permet de définir sur la surface un folio de simulation.

Les folios ainsi référencés sont affichés, il est également possible de lire la valeur instantanée de la connexion survolée ainsi que de l'objet survolé.

Paramètres spécifiques : nom du comportement associé au folio de simulation, couleur transparente, résultat survol connexion (un comportement qui recevra la valeur

instantanée de la connexion survolée), résultat survol connexion intensité (dans le cas d'une connexion électrique, reçoit l'intensité), technologie (reçoit le type de la connexion survolée : 0=aucune connexion, 1=pneumatique, 2=électrique, 3=hydraulique), résultat désignation objet survolé (reçoit dans la zone alpha du comportement la désignation de l'objet survolé).

Un exemple nommé "automsim" se trouvant dans le sous-répertoire "Technical tips\Surfaces" du répertoire des exemples illustre tout ceci :



Type caméra virtuelle

Ce type de couche reçoit l'image provenant d'une caméra de VIRTUAL UNIVERSE PRO.

Paramètres spécifiques : caméra virtuelle (nom de la caméra), mettre à l'échelle (couvre ou non toute la surface).

La caméra sélectionnée doit avoir été paramétré comme suit :

Associé à une texture : Oui

Exemple d'association Surface/Caméra :

The screenshot shows a scene hierarchy on the left and a properties panel on the right. The hierarchy includes: Univers, World, Light, Camera, **Caméra #2**, IHM #1, Contrôleur #1, and Surface #1. The properties panel for 'Caméra #2' is as follows:

Nom	
Position	
Position courante	
Limites	
Options	
Fixe	False
Tourne avec l'objet p	False
Active au démarrage	False
Coefficient de zoom	0
Coefficient de déplac	0
Associé à une texture	True

The 'Surface' dialog box contains a table of layers and a preview window. The table is as follows:

Couche	Type	X	Y	CX	CY	Actif si (toujour...
1	Caméra virtuelle	0	0	1	1	
2	Bitmap	0	0	1	1	

Un clic droit sur la liste ci-dessus permet d'ajouter, déplacer ou supprimer un élément, un double clic montre ou cache la couche.

The preview window shows a 3D scene with a Siemens control panel and a camera. The text 'TOUCH' is visible on the panel.

Properties of the selected element in the list:

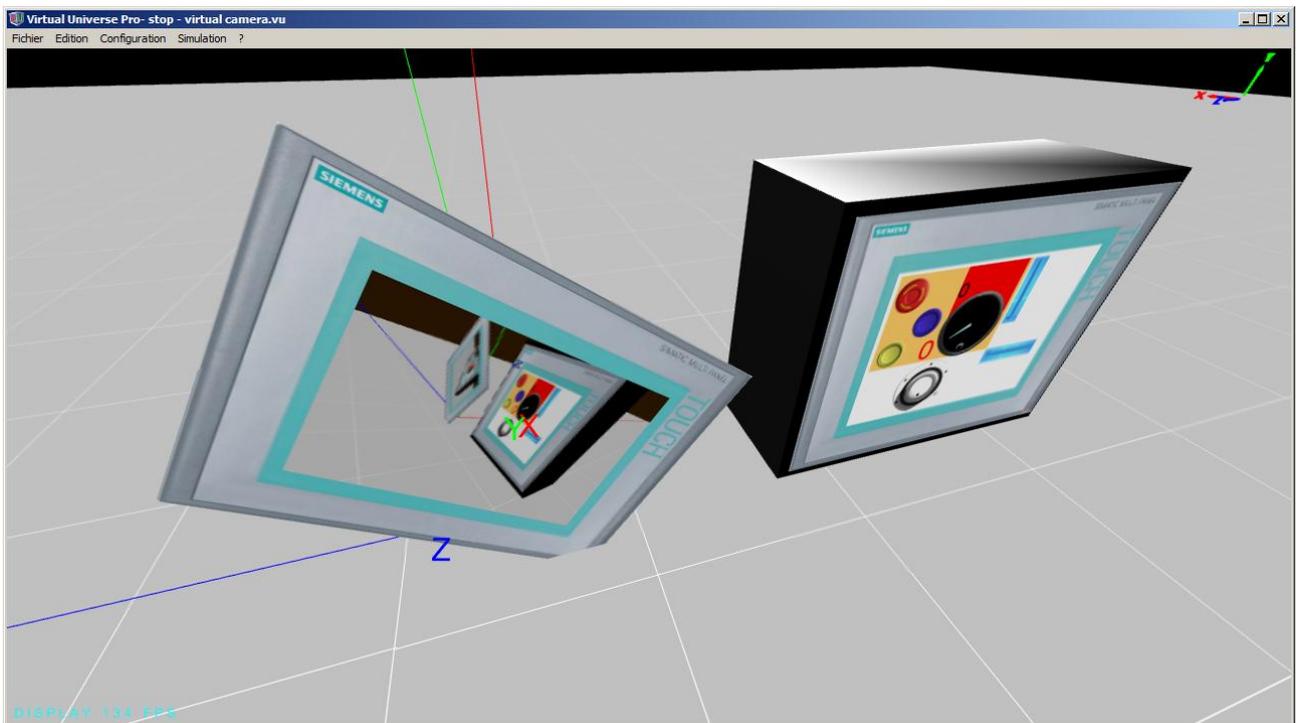
Type X	Constante
X	0
Type Y	Constante
Y	0
Type largeur	Constante
CX	1
Type hauteur	Constante
CY	1
Actif si (toujours si vide) un test n	
Caméra virtuelle	\Caméra #2
Mettre à l'échelle	True

Propriétés de l'élément sélectionné dans la liste

Aperçu

Buttons: Annuler, OK

Un exemple nommé "virtual camera" se trouvant dans le sous-répertoire "Technical tips\Surfaces" du répertoire des exemples illustre tout ceci :

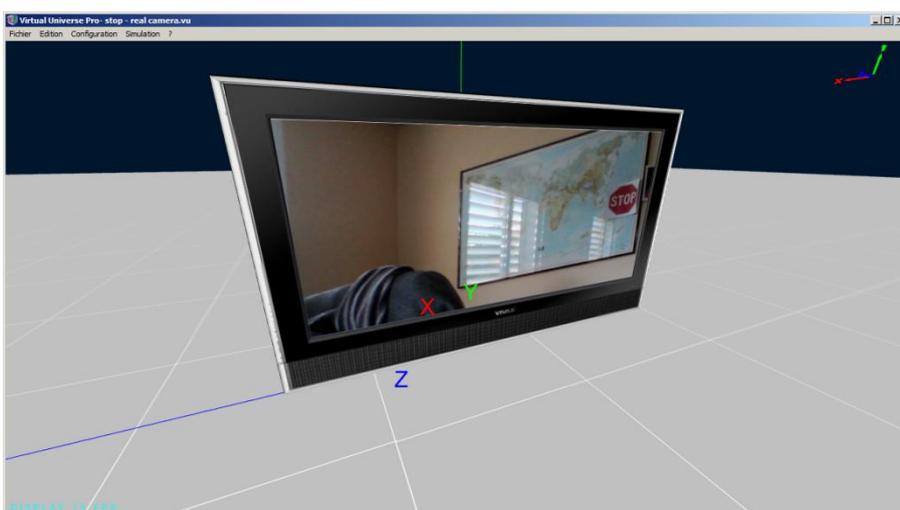


Type caméra réelle

Ce type de couche permet d'insérer une image provenant d'une webcam sur une surface.

Paramètres spécifiques : caméra réelle (numéro de la source vidéo : 0 première, 1 seconde, etc.), mise à l'échelle (couvre toute la surface ou pas).

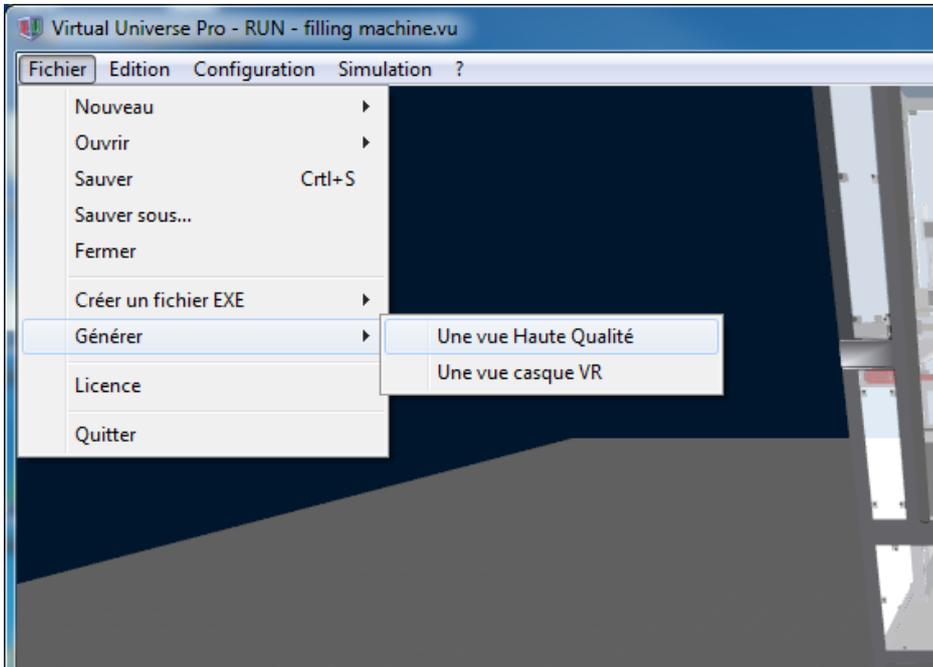
Un exemple nommé "real camera" se trouvant dans le sous-répertoire "Technical tips\Surfaces" du répertoire des exemples illustre tout ceci :



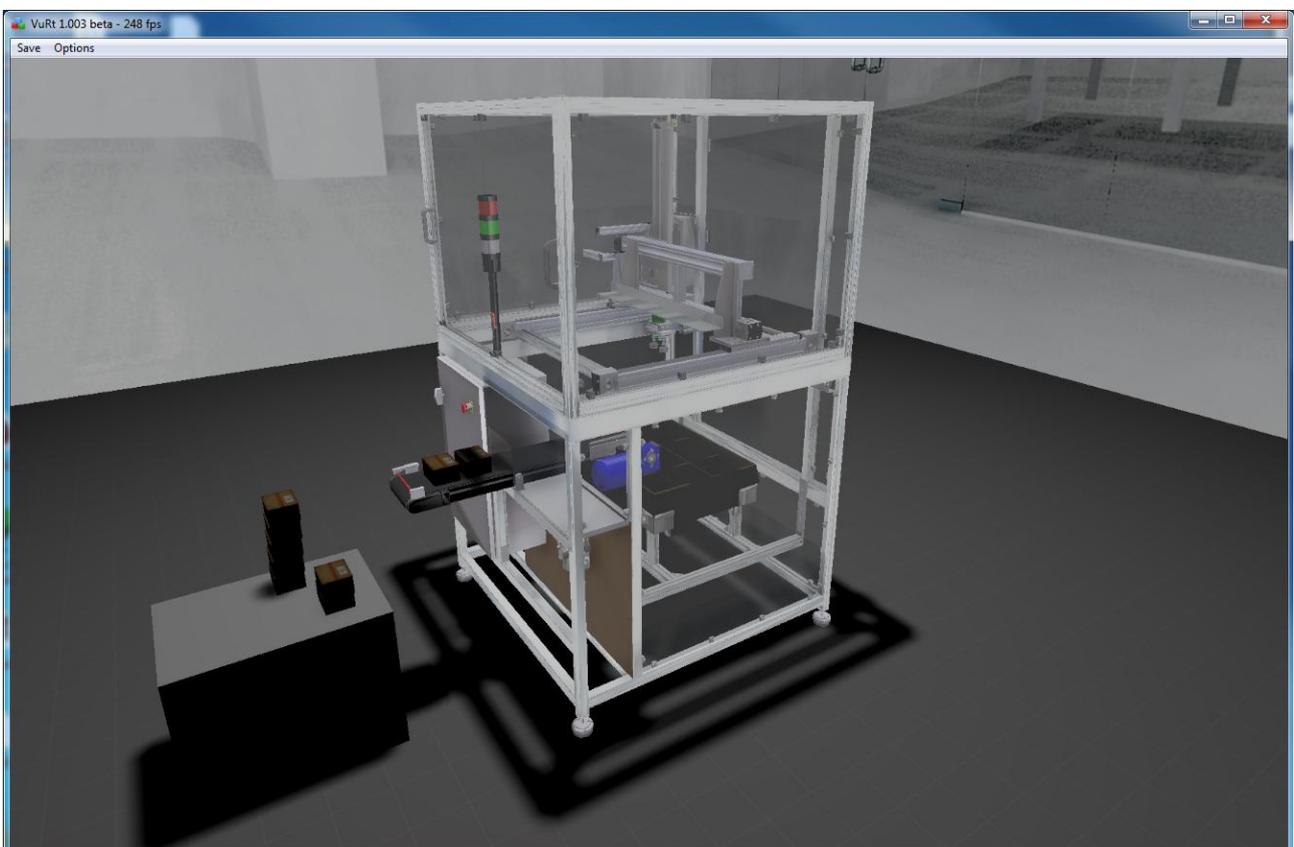
Vue haute qualité

Utilisation

Pour générer une vue haute qualité, utilisez le menu Fichier / Générer / Une vue haute qualité.



Une fenêtre contenant la vue haute qualité est alors ouverte :



En mode simulation (RUN), cette vue est active, il est possible de cliquer sur les objets et de les saisir.

Navigation

Les commandes de navigations à la souris sont identiques à celle de la vue de base de VIRTUAL UNIVERSE PRO.

Configuration

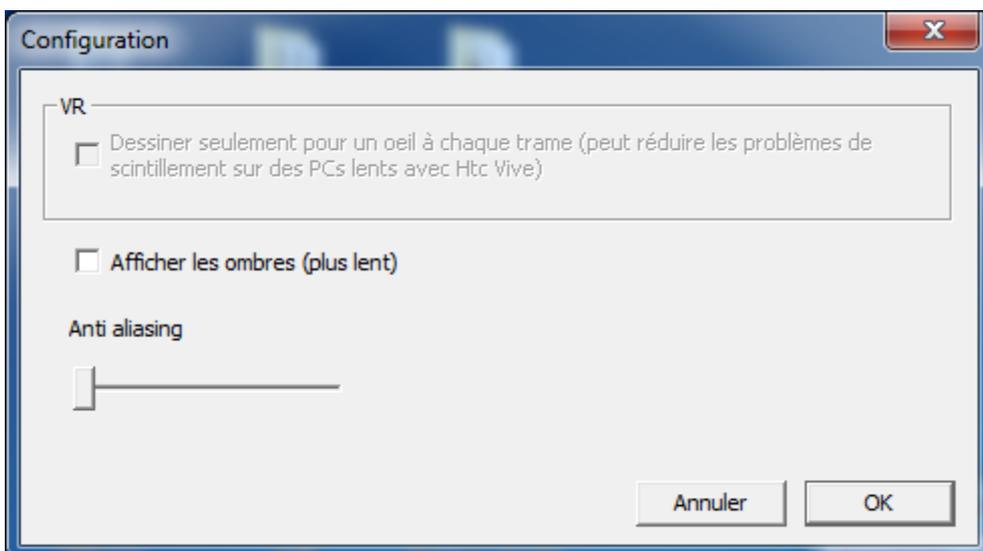
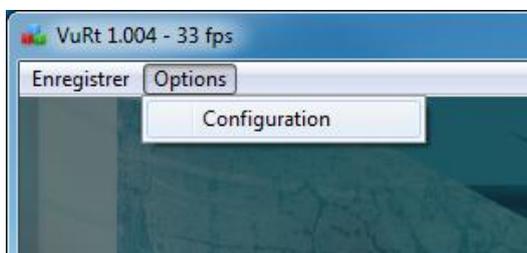
Position initiale de la caméra

Pour définir la position initiale de la caméra, naviguez dans la vue haute qualité vers la position initiale souhaitée, puis valider cette position dans le menu suivant :



Qualité

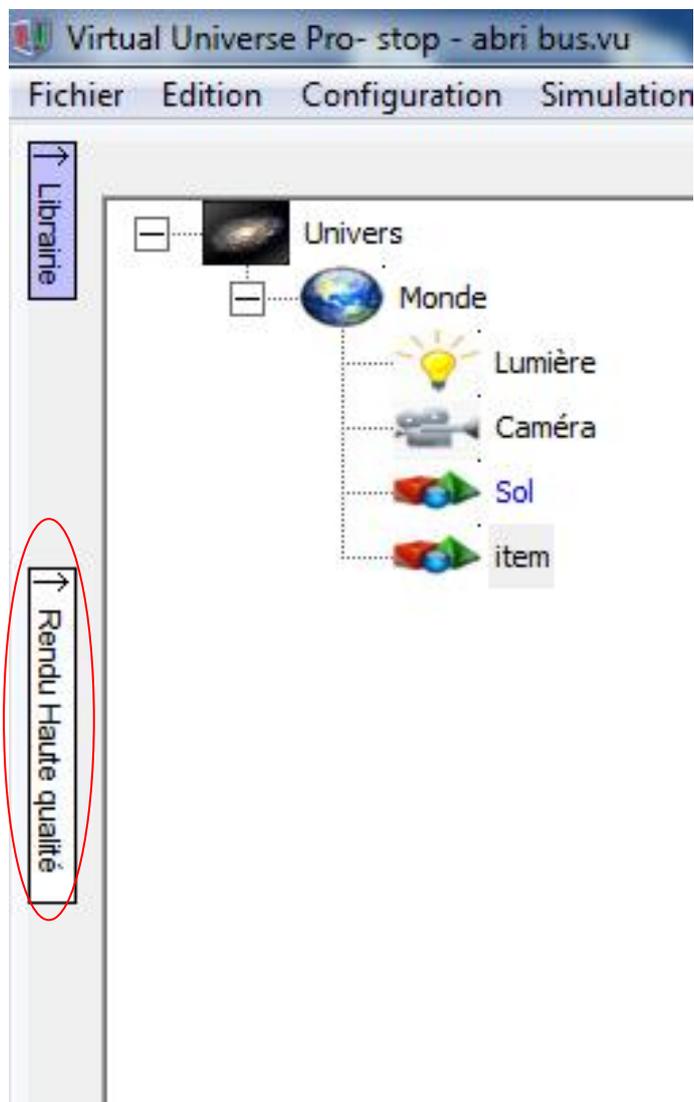
La qualité d'affichage peut être modulée afin d'obtenir le meilleur compromis entre vitesse d'affichage et qualité.



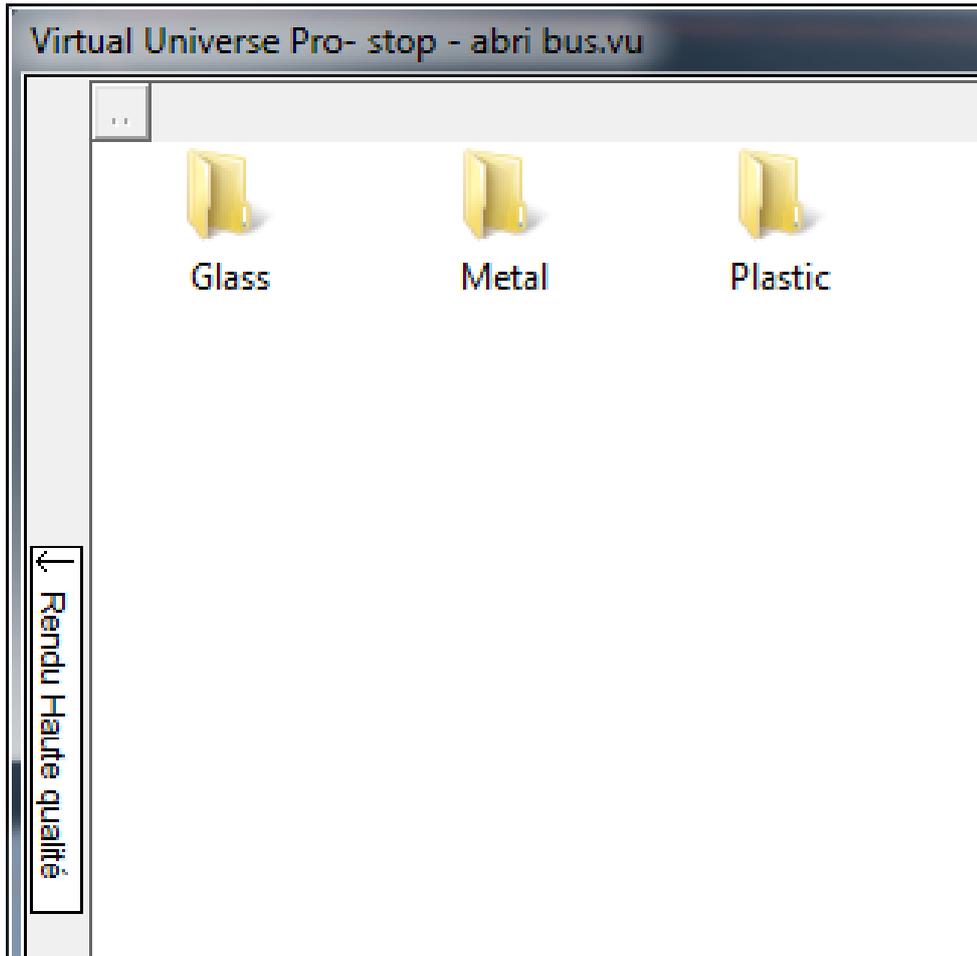
Matériaux

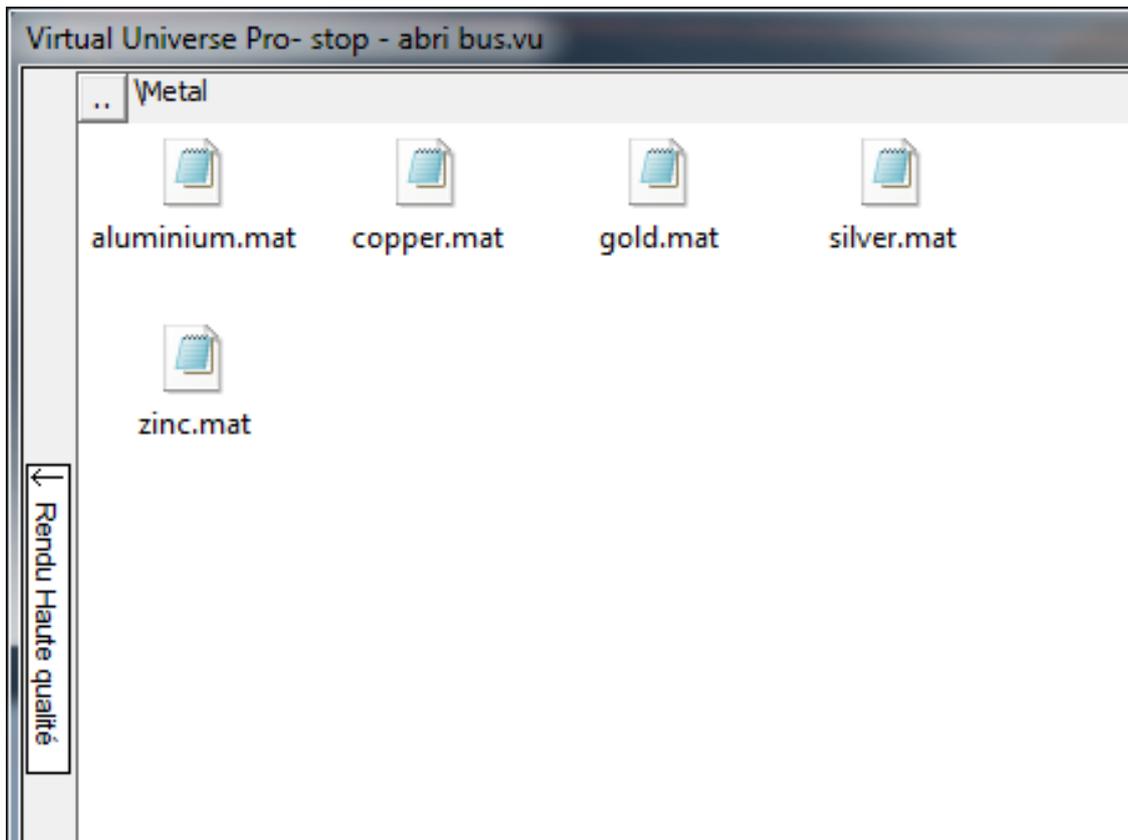
Associer un matériau prédéfini à un objet ou à une partie d'objet.

Depuis la fenêtre de configuration, cliquez sur le bouton "Rendu haute qualité" :

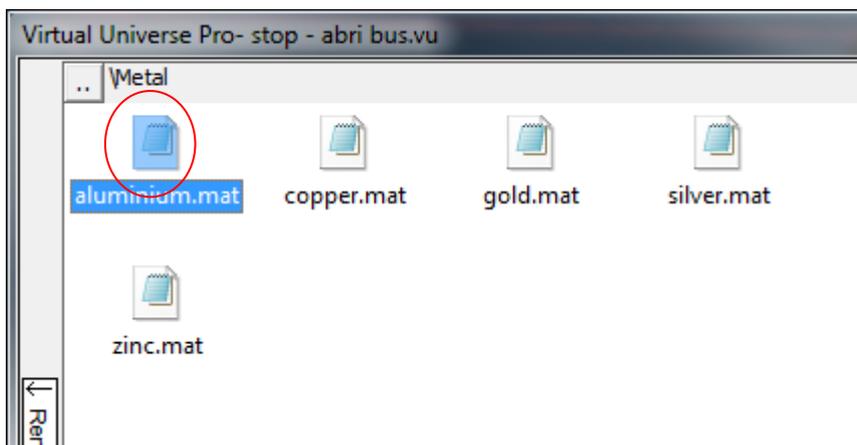


Vous pouvez choisir un matériau prédéfini dans la liste :

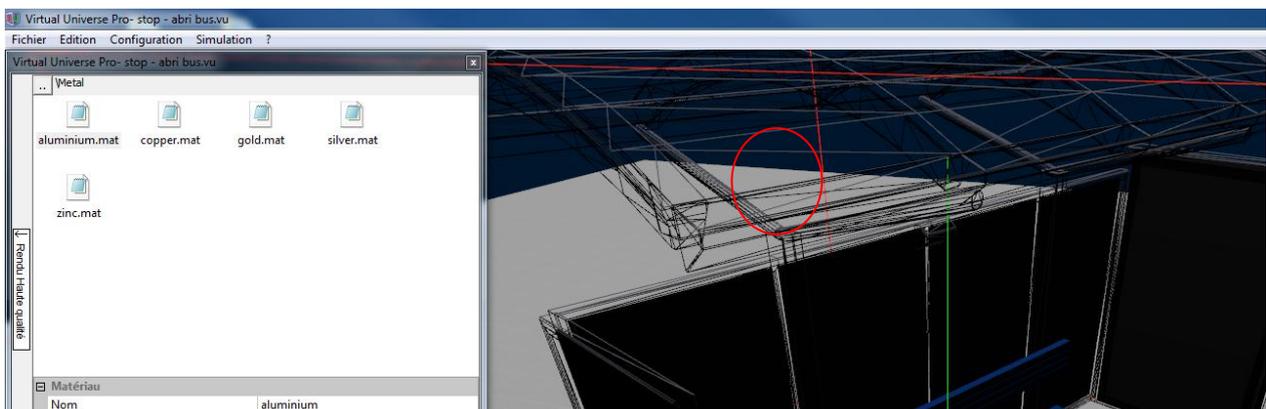




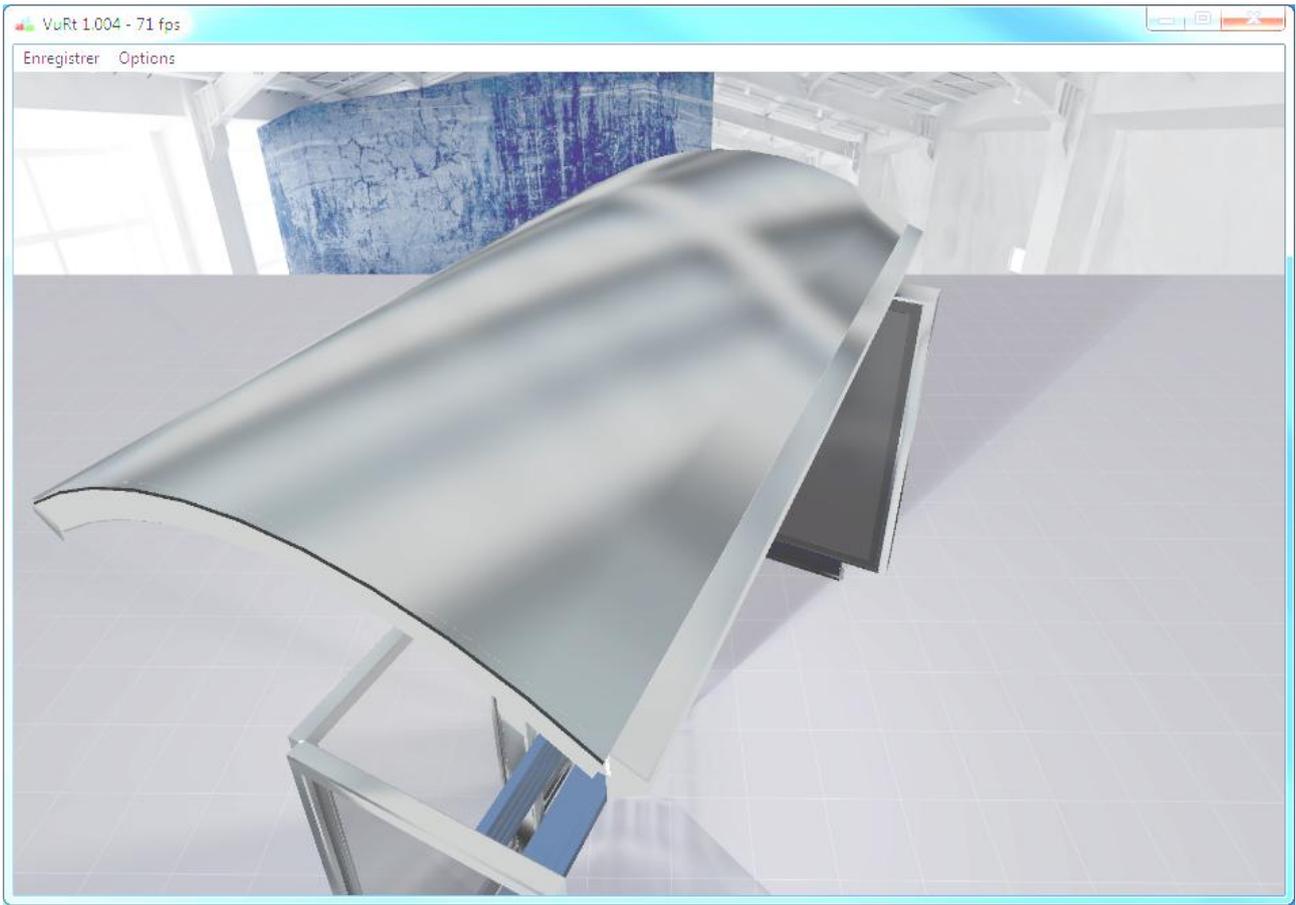
Sélectionnez un des matériaux :



Puis cliquez sur l'objet ou la partie d'objet :

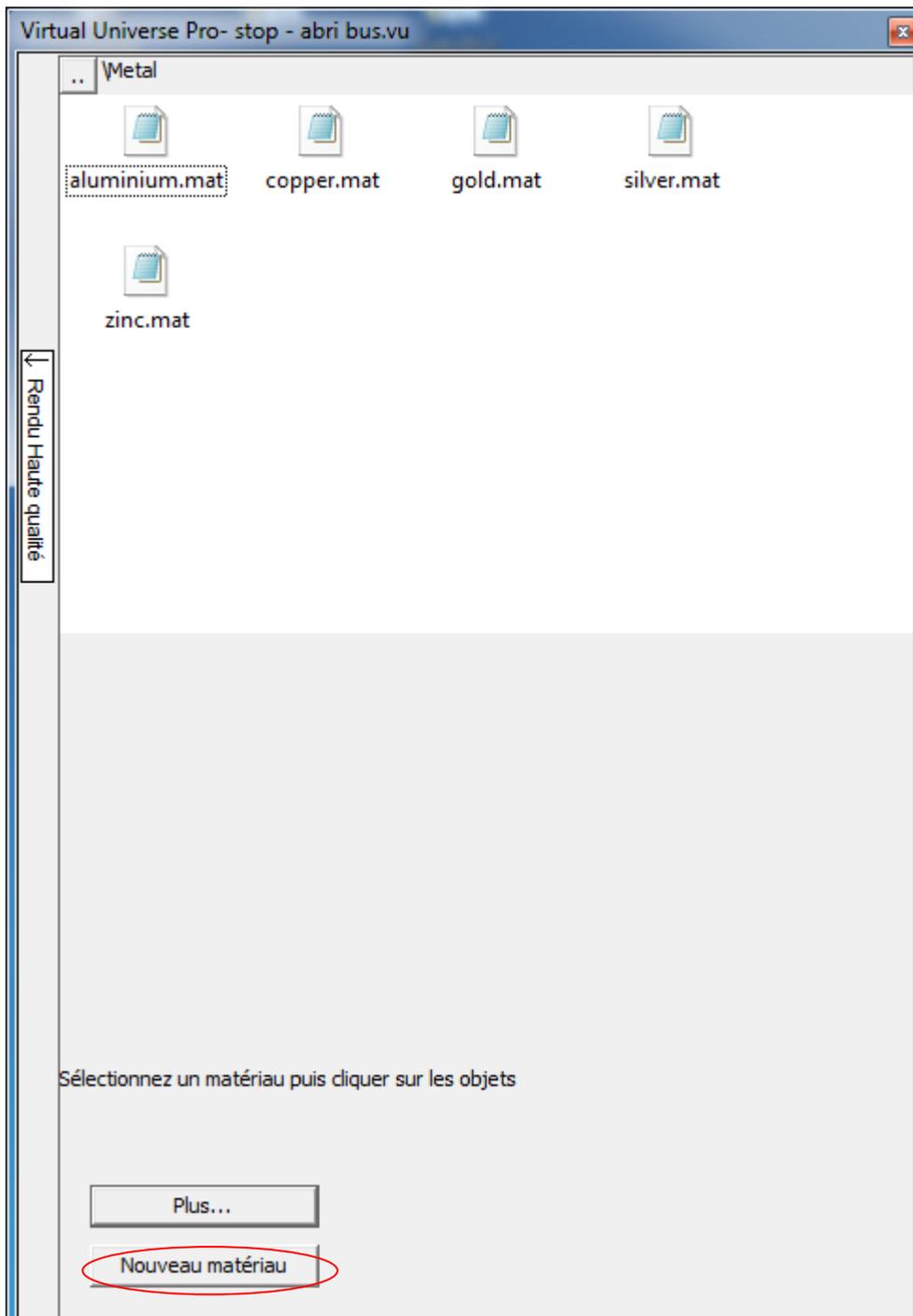


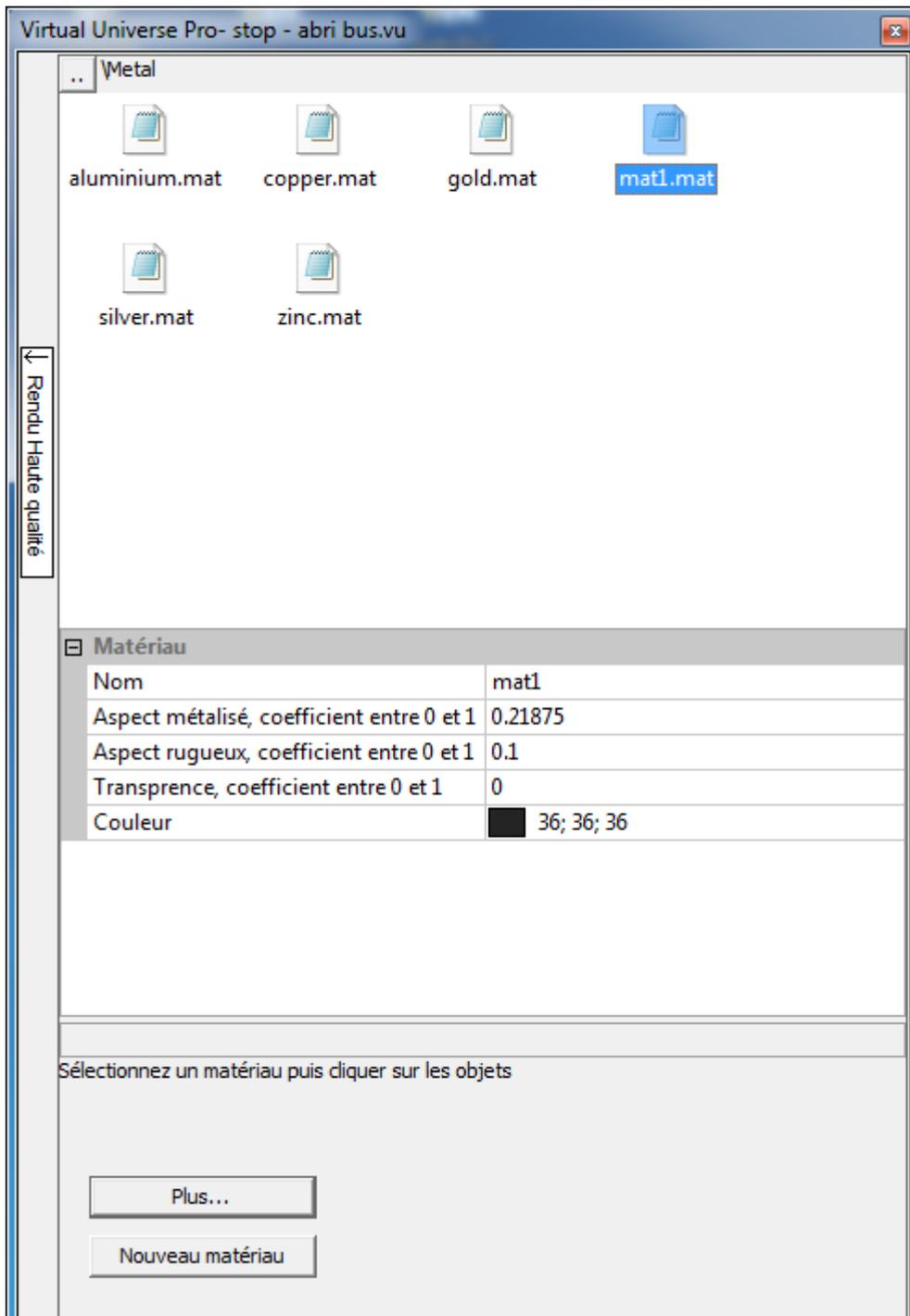
La vue haute qualité affiche immédiatement la modification



Créer un nouveau matériau

Cliquez sur "Nouveau matériau" :

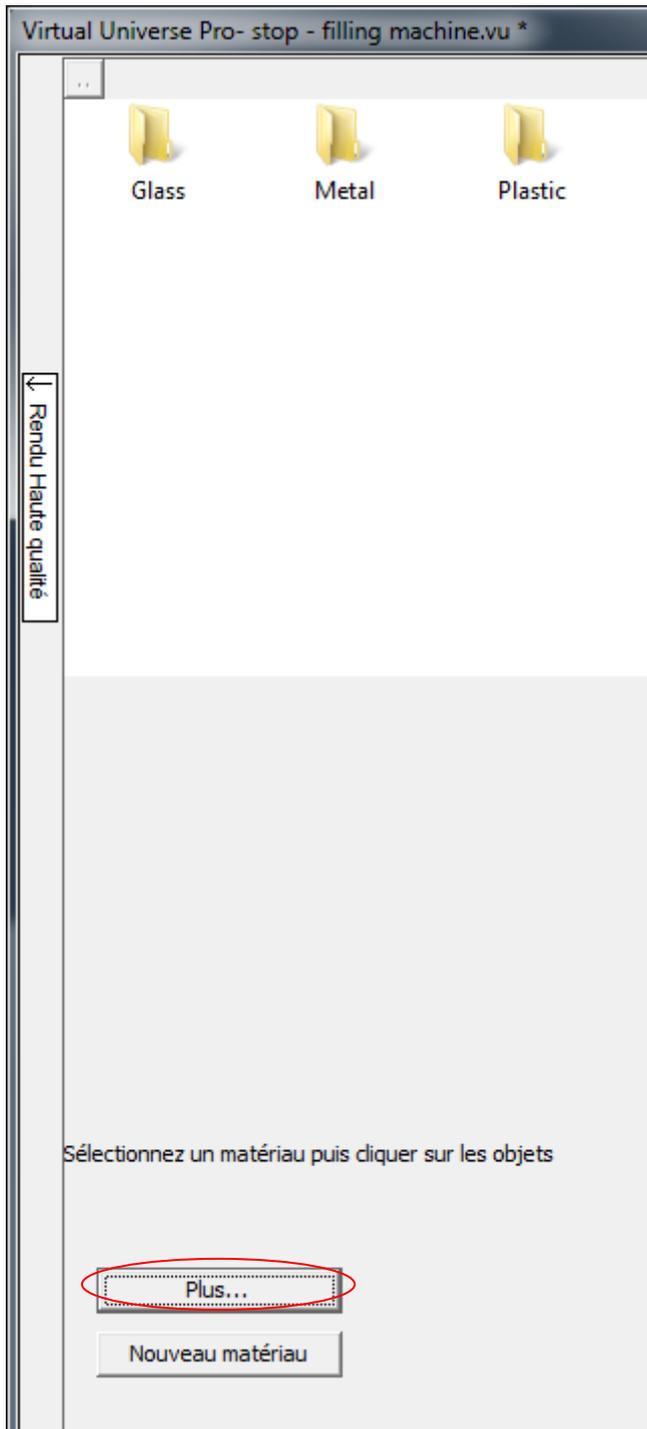




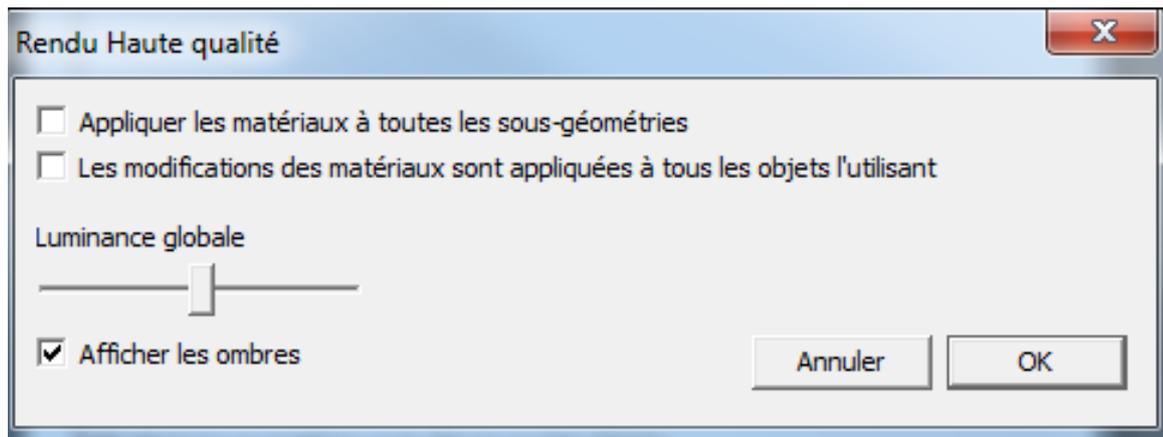
Le nouveau matériau apparaît dans la liste, vous pouvez le renommer et modifier ses propriétés.

Plus de propriétés

En cliquant sur



Vous accédez aux propriétés suivantes :



"Appliquer les matériaux à toutes les sous géométries"

Certains objets 3D contiennent plusieurs sous géométries. Si cette case est cochée, l'affectation des matériaux est faite pour l'ensemble de ces géométries, dans le cas contraire, on peut affecter différents matériaux à chaque sous géométrie.

"Les modifications des matériaux sont appliquées à tous les objets l'utilisant".

Si cette case est cochée et que l'on modifie les propriétés d'un matériau, alors ces modifications sont répercutées aux objets utilisant déjà ce matériau.

"Luminance globale"

Règle la luminance globale de la vue.

"Afficher les ombres"

Autorise l'affichage des ombres.

Remarque : Les paramètres "possède une ombre" et "reçoit une ombre" des sprites 3d sont utilisés pour la gestion des ombres.

Réalité virtuelle

Installation

VIRTUAL UNIVERSE PRO utilise l'environnement STEAM et STEAMVR comme interface avec les casques et contrôleurs de réalité virtuelle.

Pour installer Steam, rendez-vous sur le site Internet www.steamvr.com, puis cliquez sur "Installer Steam".

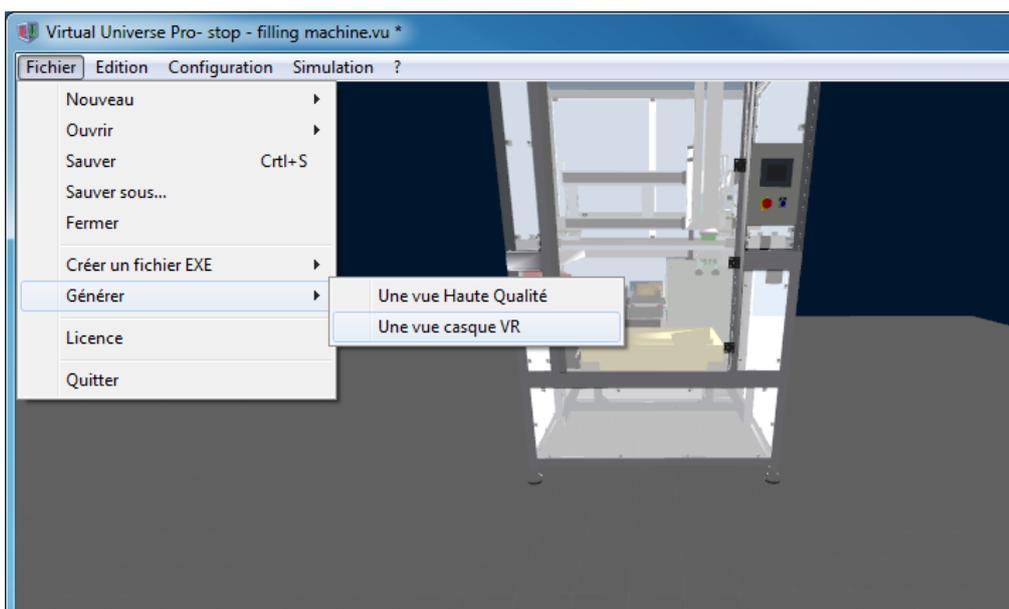
Une fois Steam installé, lancez le, puis cliquez sur "BIBLIOTHEQUE" puis sur "INSTALLER" en dessous de SteamVR.



Une fois SteamVr installé, vous pouvez procéder à l'installation du matériel , pour ceci, reportez-vous au manuel constructeur de chaque système.

Utilisation

Pour générer une vue casque VR, utilisez le menu Fichier / Générer / Une vue casque VR.



Une fenêtre indiquant l'activité de la vue VR est affichée :

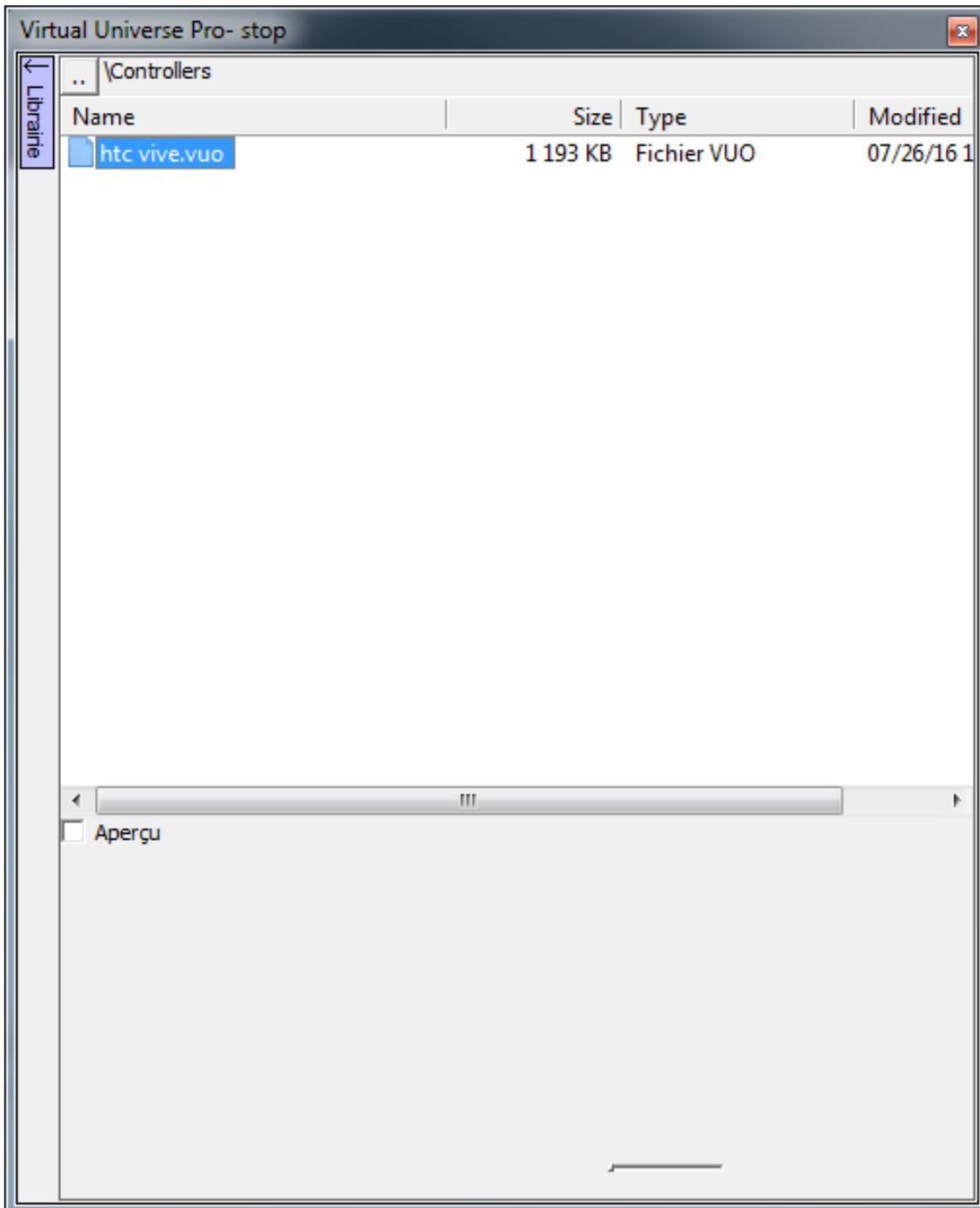


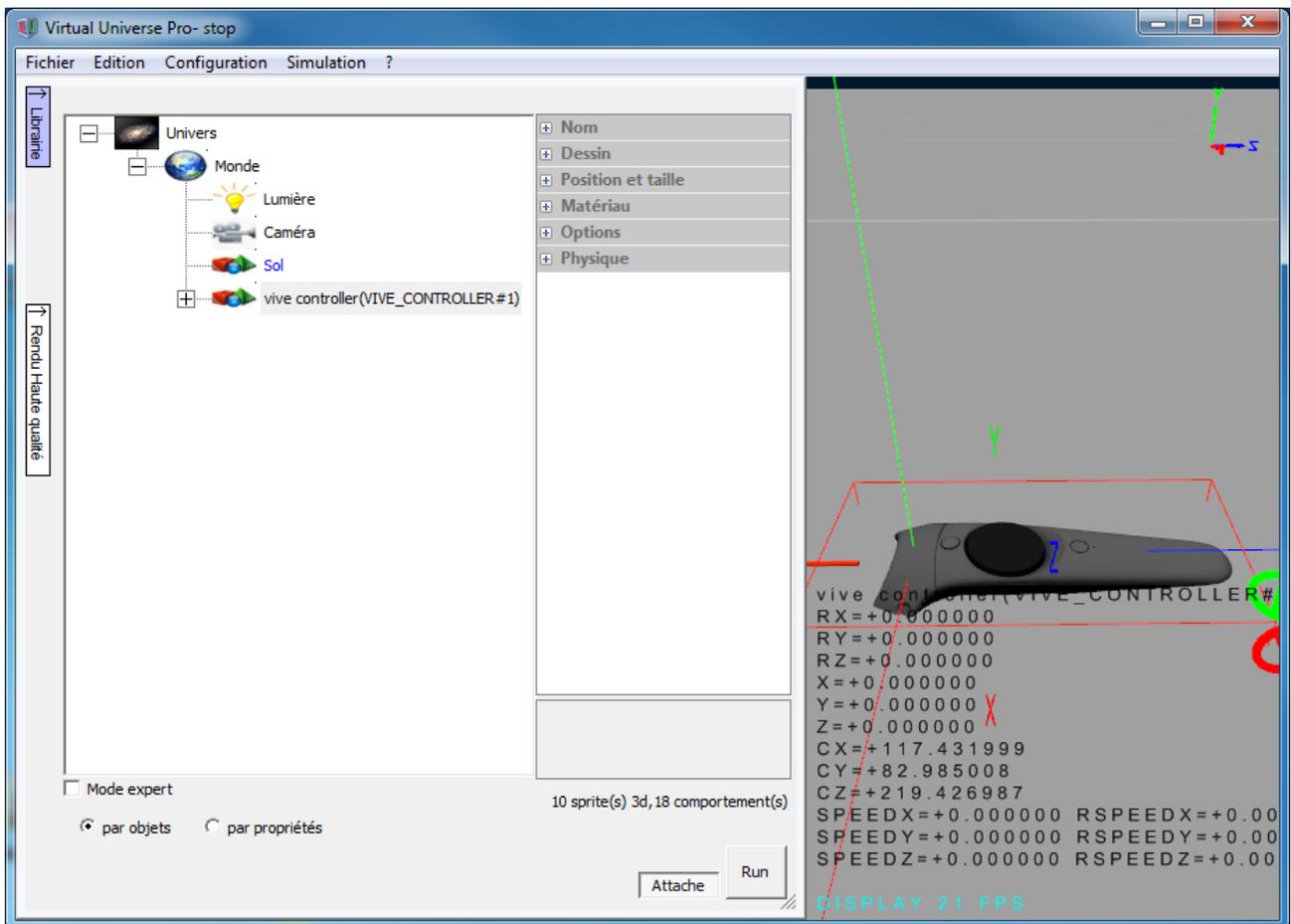
Configuration

La configuration est réalisée de la même façon que pour la vue haute qualité, voir le chapitre précédent.

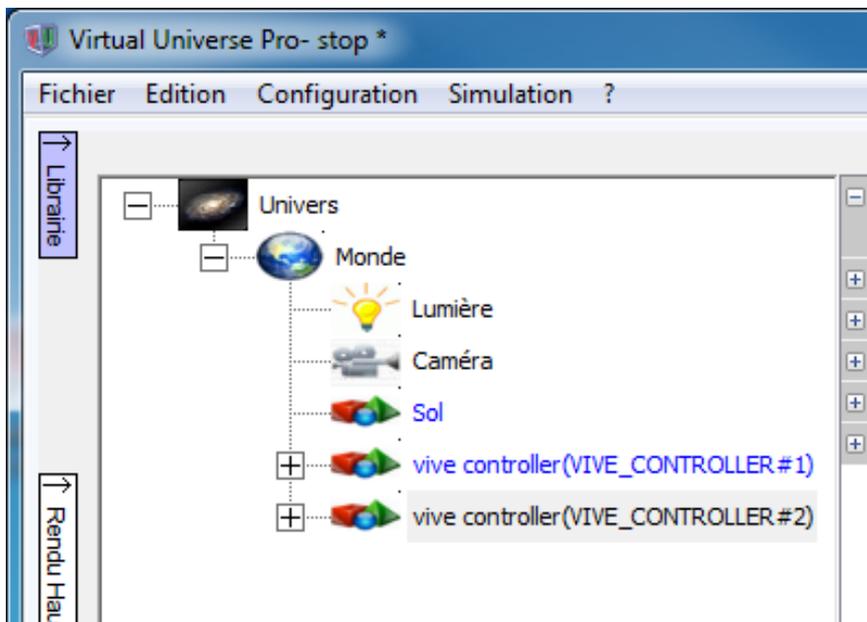
Contrôleurs Htc Vive

Pour utiliser les contrôleurs Htc Vive, des objets prédéfinis peuvent être ajoutés à un projet:





Deux contrôleurs peuvent être ajoutés. Les noms des contrôleurs vont déterminer le lien avec les contrôleurs réels. Les noms des deux contrôleurs doivent être :



Par défaut, les contrôleurs Htc Vive possèdent les comportements suivants:

- activation d'un "laser" de sélection avec la gâchette du contrôleur,

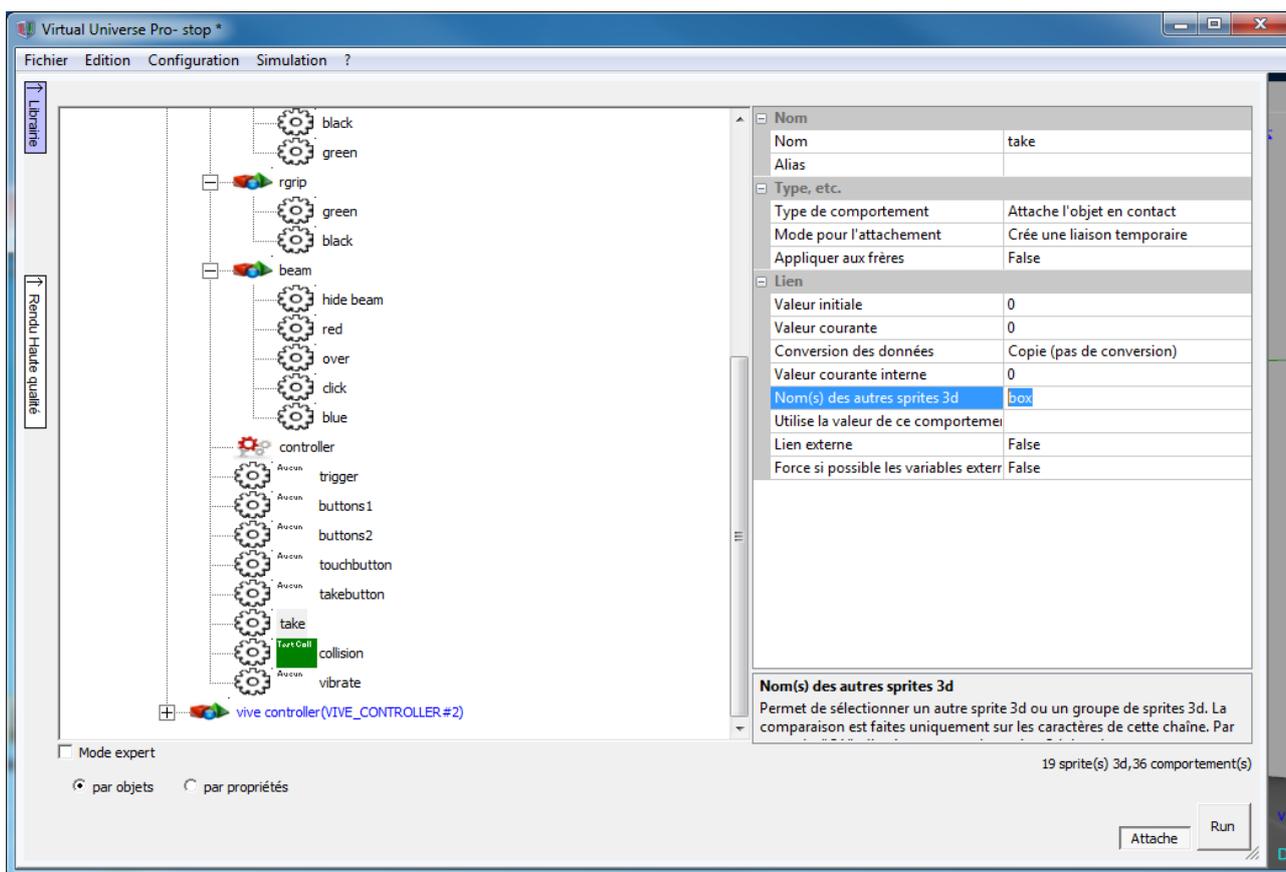


- saisie des objets avec les boutons latéraux :



L'ensemble des comportements de ces contrôleurs (saisie d'objets, sélection, etc.) sont définis par des comportements et un contrôleur virtuel. Tout ceci est entièrement paramétrable.

Concernant la saisir des objets, le nom des objets à saisir peut être paramétré dans cette propriété :



Contrôleur XBOX

Si vous utilisez une solution de casque sans contrôleur (casque Oculus Rift sans contrôleur par exemple), vous pouvez utiliser une manette XBOX One.

Les boutons arrière permettent de déplacer la vue vers le haut et le bas :



Ce joystick permet de translater la vue :



Le bouton "A" permet de réaliser les actions (sélectionner, attraper un objet, etc.), un cercle blanc dans la vue VR matérialise la zone d'action.

Serveur de simulation

Principe

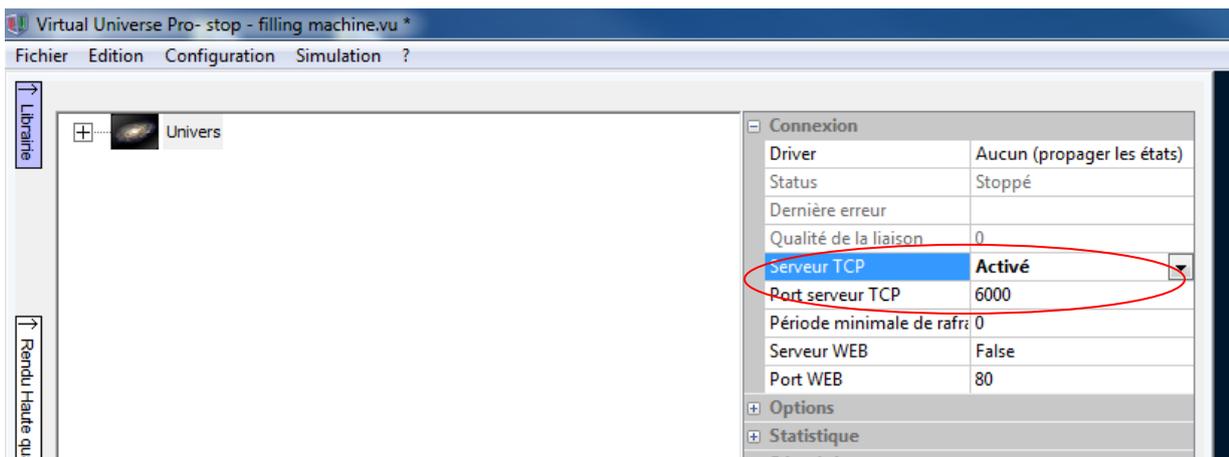
VIRTUAL UNIVERSE PRO s'exécutant sur un PC opère comme un serveur de simulation pour plusieurs clients. Ces clients peuvent être des PCs Windows exécutant le client VIRTUAL UNIVERSE PRO ou un simple navigateur Internet compatible Webgl s'exécutant sous toute plateforme (PC, MAC, Smartphones et tablettes sous Windows, IOS, Android, etc.). Le client PC peut utiliser un casque et des contrôleurs de réalité virtuelle tel que décrit au chapitre précédent.

Les clients ainsi connectés accèdent en temps réel à la simulation et peuvent interagir avec celle-ci.

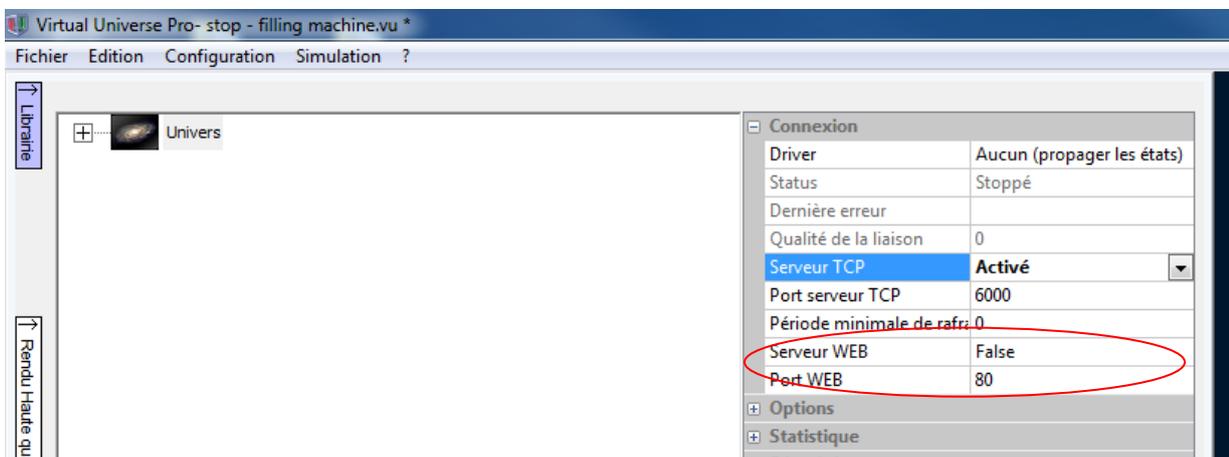
Configuration

Une configuration distincte existe pour les clients PC et les clients WEB. Pour chacun de ces types de clients, un port doit être défini.

Pour les clients PC :



Pour les clients Web :



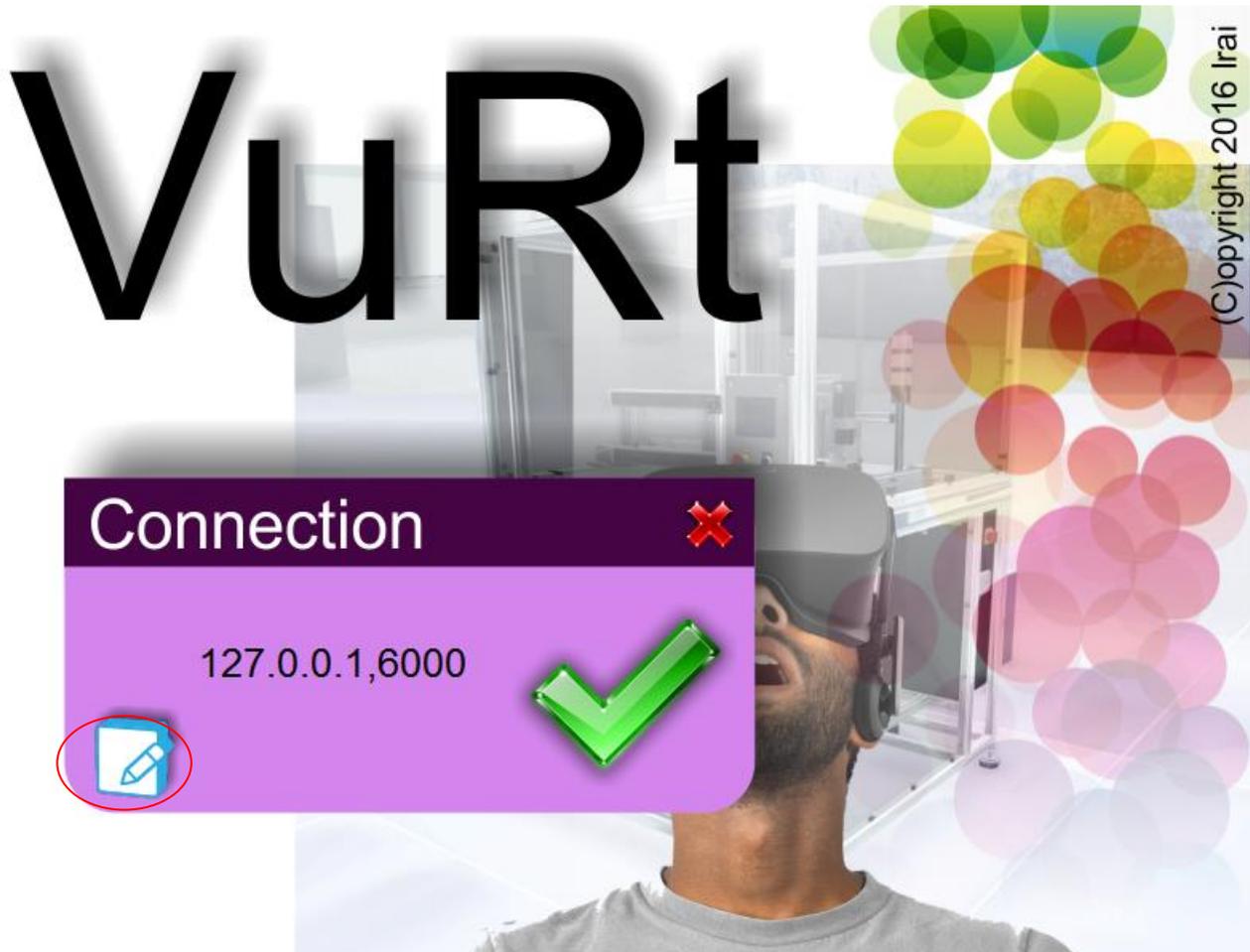
Lancement du client PC

Les exécutables du clients PC se trouvent dans le répertoire d'installation de VIRTUAL UNIVERSE PRO.

vurtwp.exe est le client vue haute qualité.

vurtvrwp.exe est le client vue casque VR.

Au lancement de ces clients, on peut paramétrer l'adresse et le port en cliquant ici :

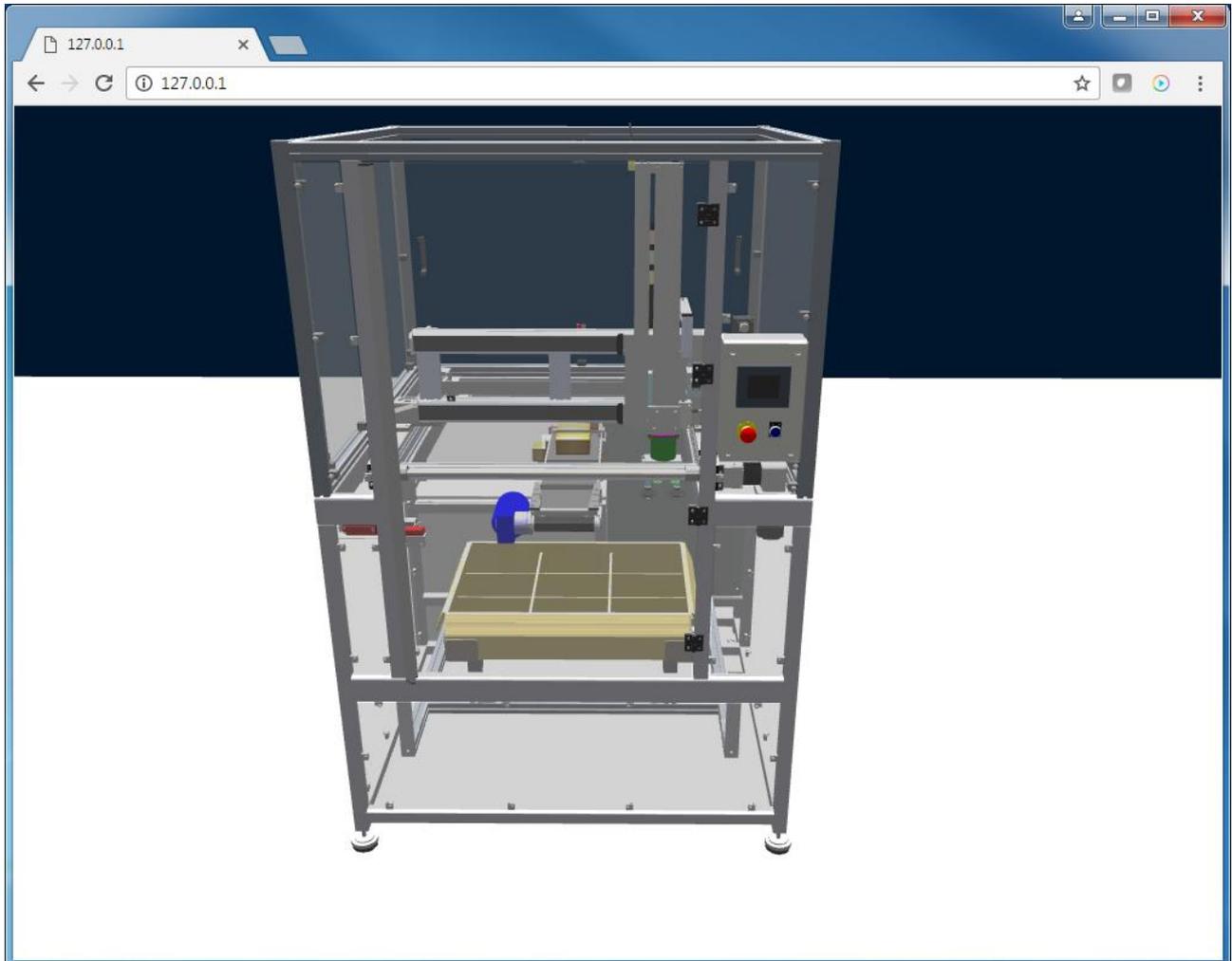


En cliquant sur la coche verte, on lance la connexion au serveur.

Lancement du client Web

A partir d'un navigateur Web, entrer simplement l'adresse du serveur, suivi éventuellement de " :<numéro de port>" si le port est autre que 80.

Exemple :



Navigations

Depuis un client PC ou un client Web de type ordinateur, la navigation à la souris est identique à celle de VIRTUAL UNIVERSE PRO.

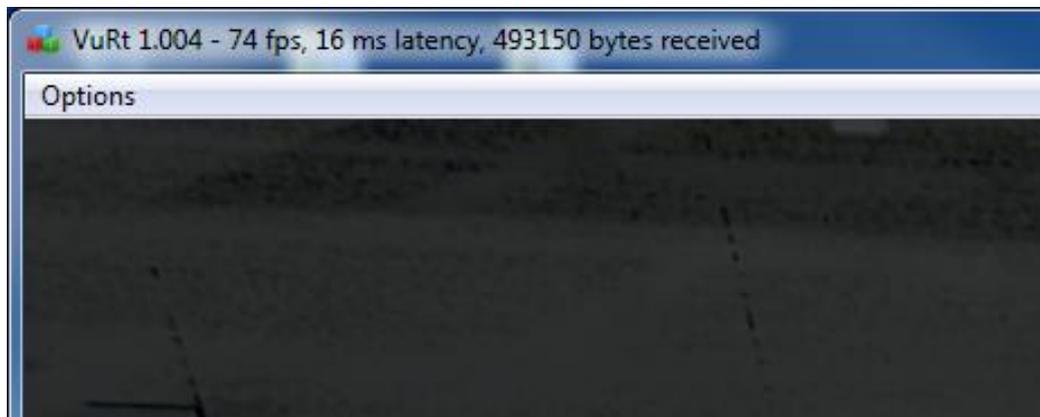
Depuis un client écran tactile, la navigation s'opère de la façon suivante :

- un doigt : sélection ou saisie d'objets,
- deux doigts : zoom,
- trois doigts : déplacements.

Serveur distant

Il est possible de rendre un serveur de simulation accessible à distance. Dans ce cas, il convient de paramétrer le serveur pour réduire au maximum la latence de la connexion.

La valeur de la latence est affichée dans la barre de titre du client PC :

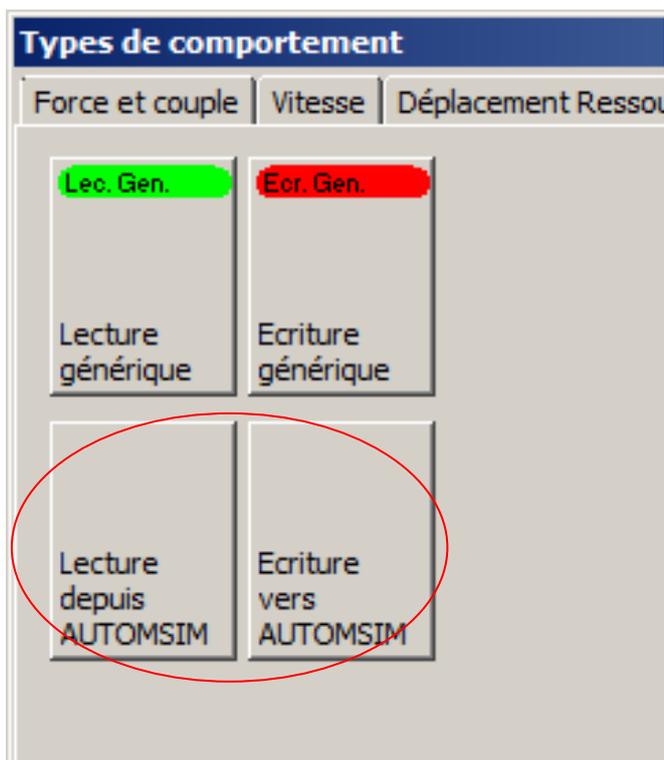


Pour une réactivité correcte de la simulation, une latence maximum de 100ms est conseillée.

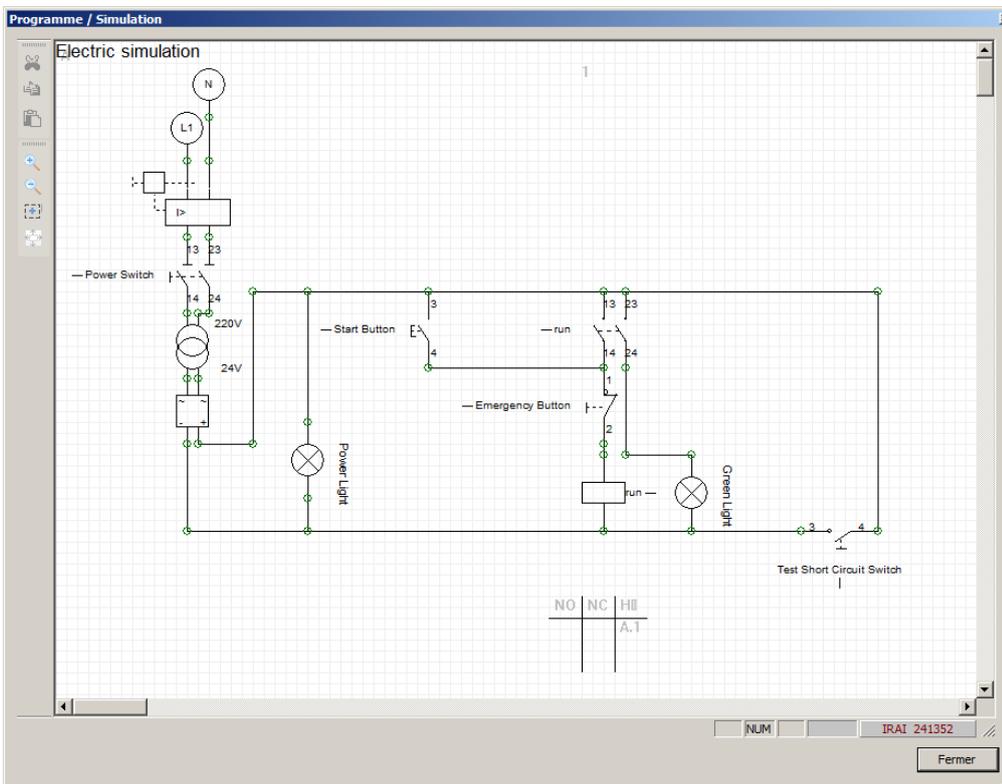
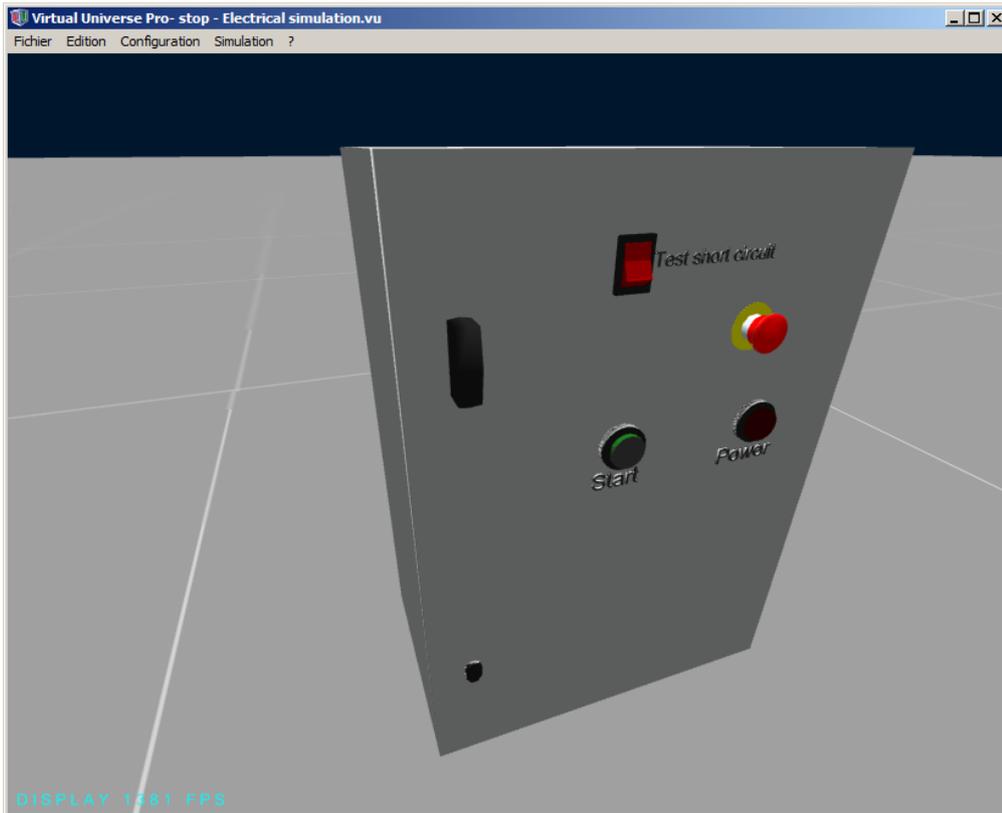
Simulation de schémas

L'éditeur simulateur de schémas intégré à VIRTUAL UNIVERSE PRO permet de créer et de simuler des schémas dans les domaines électriques, pneumatiques, hydrauliques et électroniques digitale en lien avec les éléments 3D d'un projet.

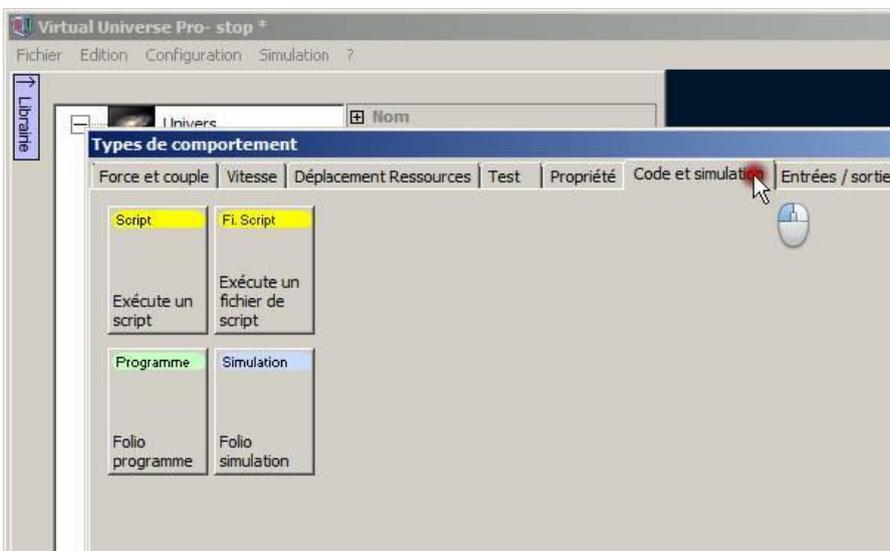
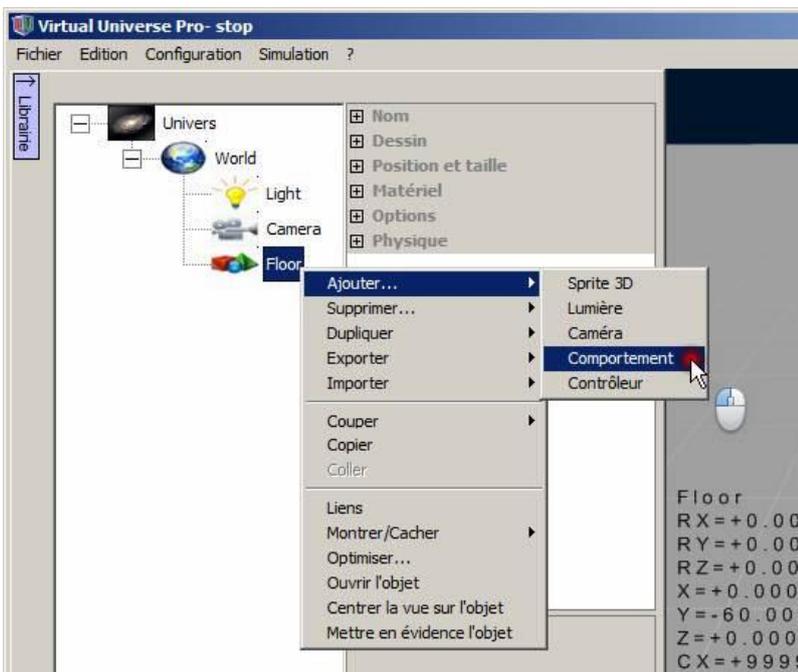
Les comportements de type "Lecture depuis AUTOMSIM" et "Ecriture depuis AUTOMSIM" permettent respectivement de lire ou d'écrire un état depuis un folio de simulation.

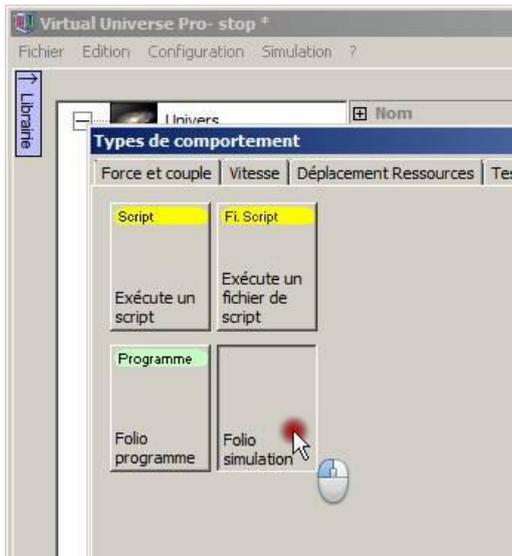


Exemple d'un pupitre électrique associé à un schéma :

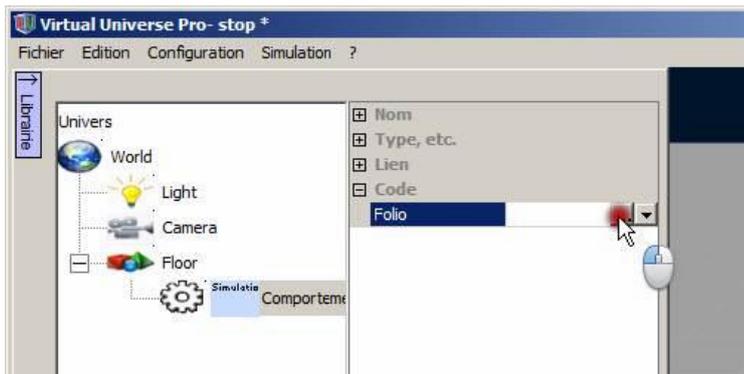


Ajout d'un folio de simulation



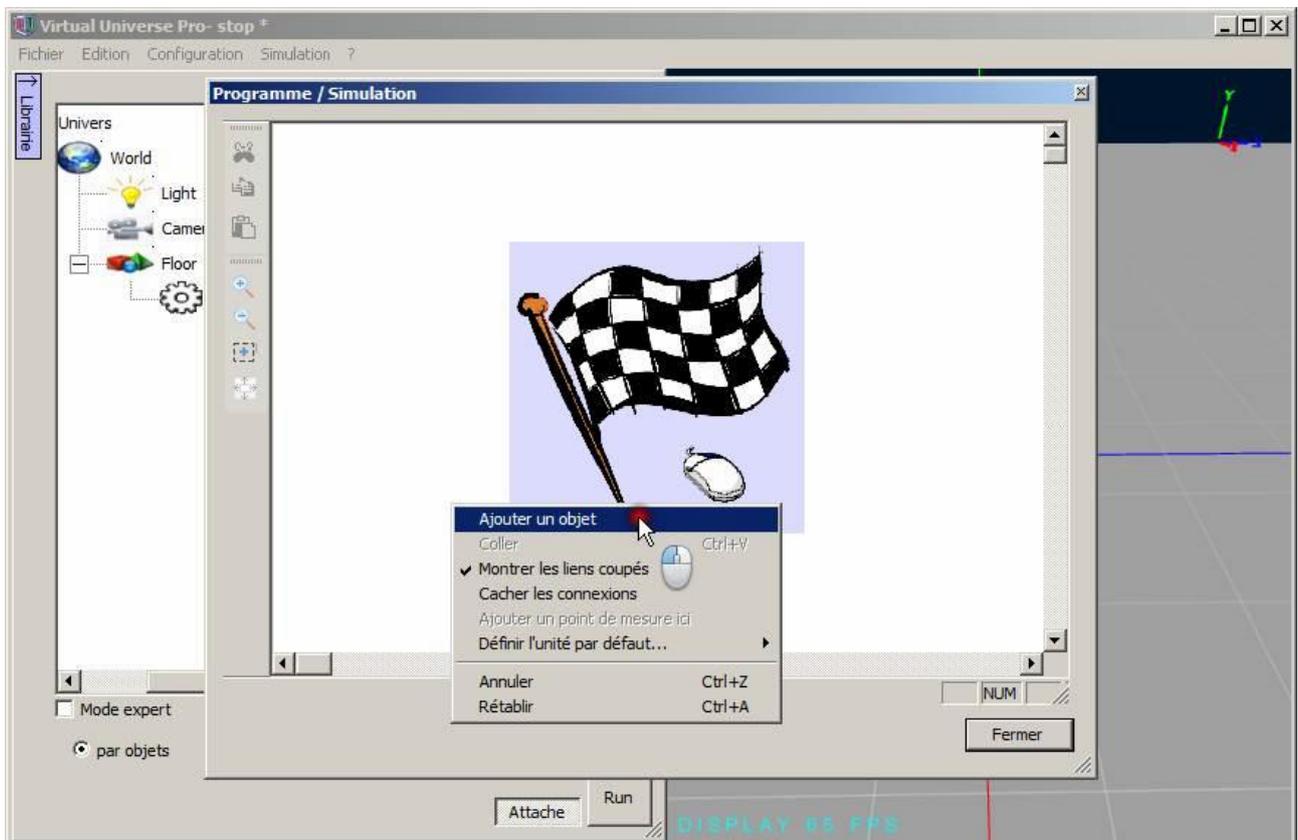
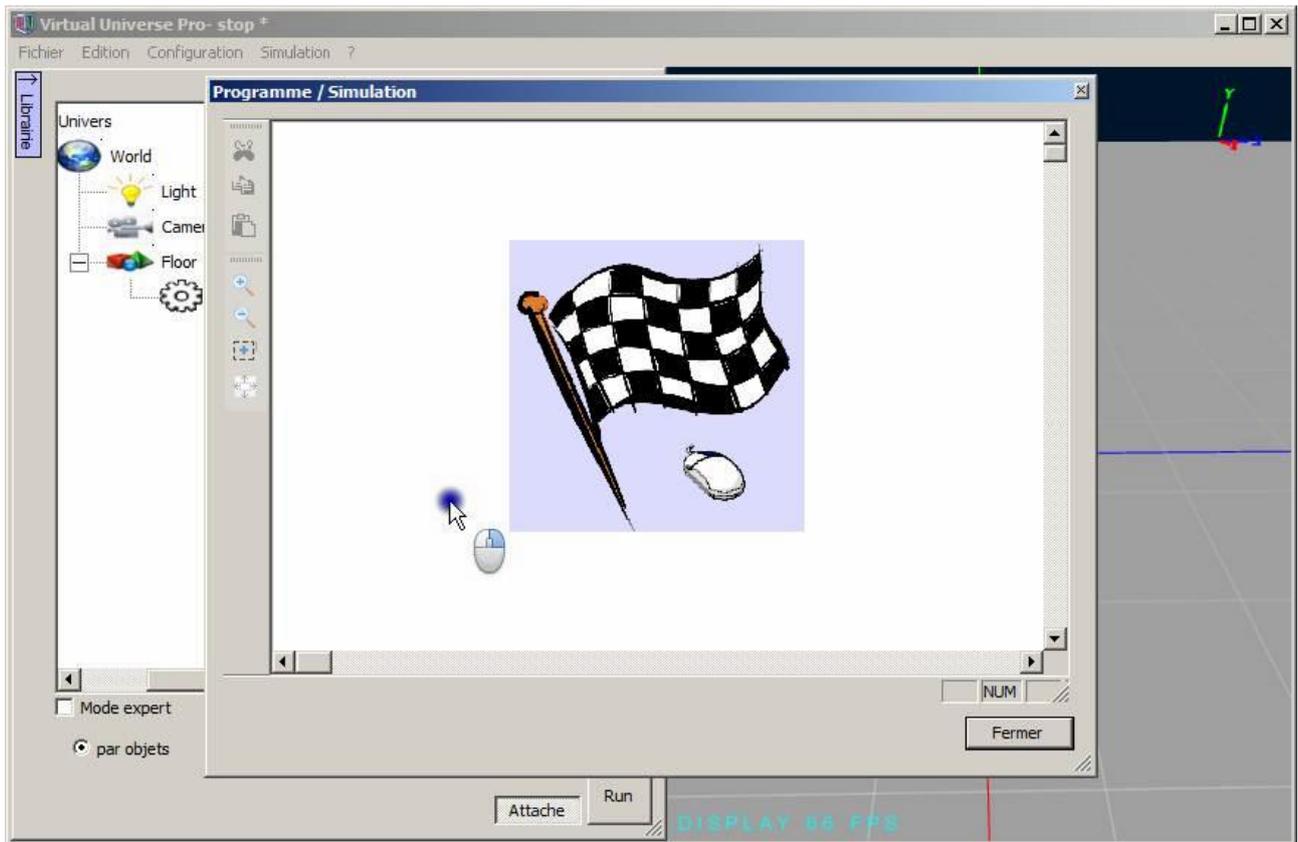


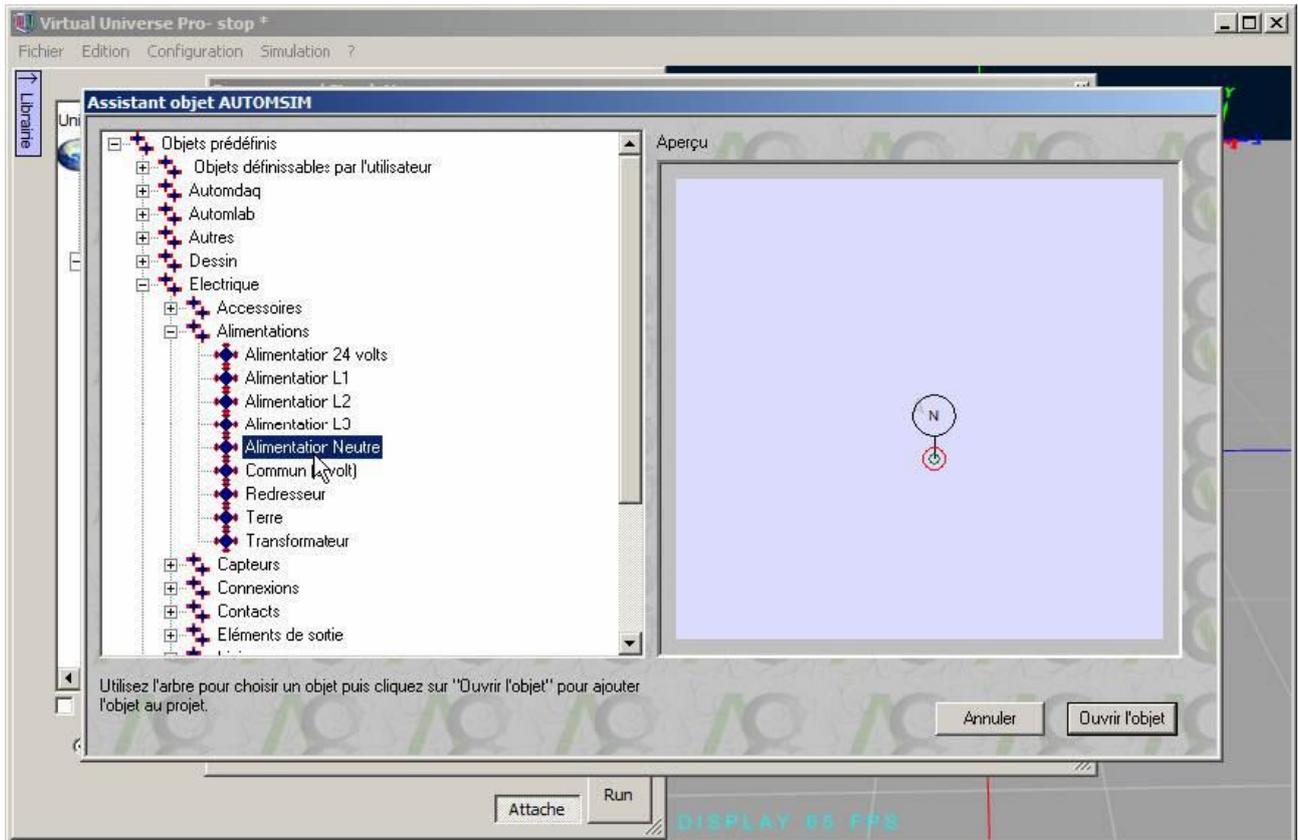
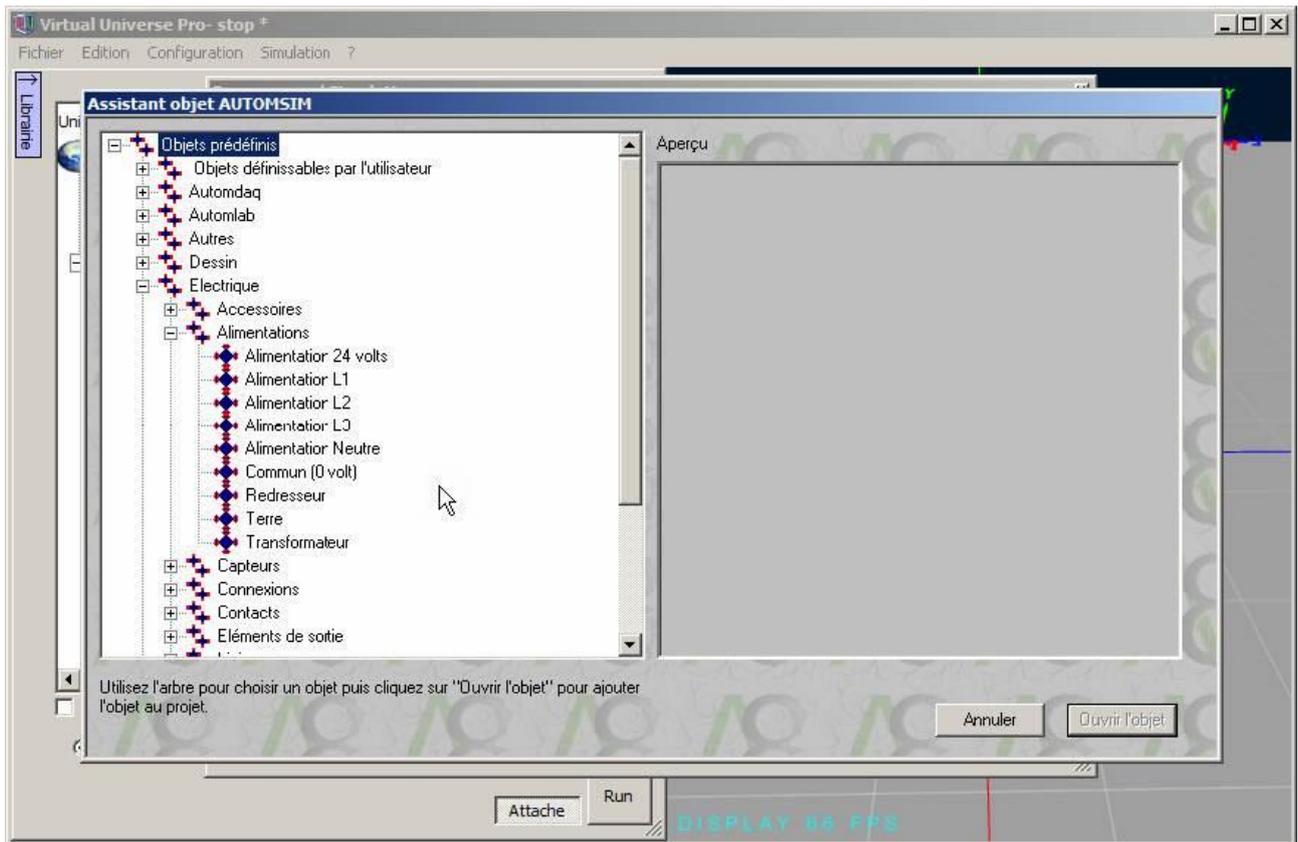
Edition du contenu d'un folio de simulation



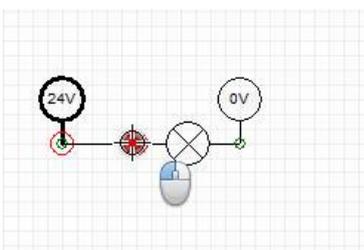
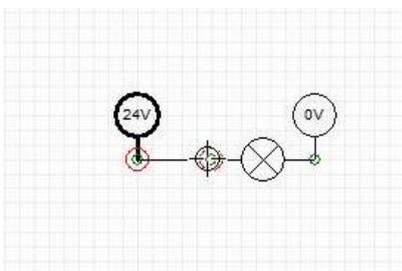
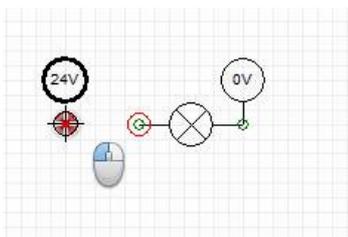
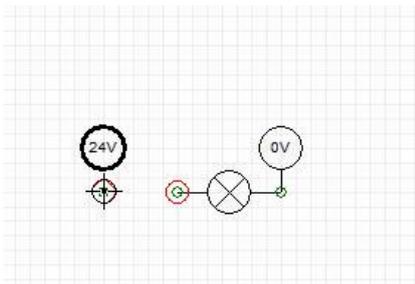
Un double clic sur le comportement permet également d'ouvrir le folio de simulation en édition.

Ajout d'un objet sur un folio de simulation





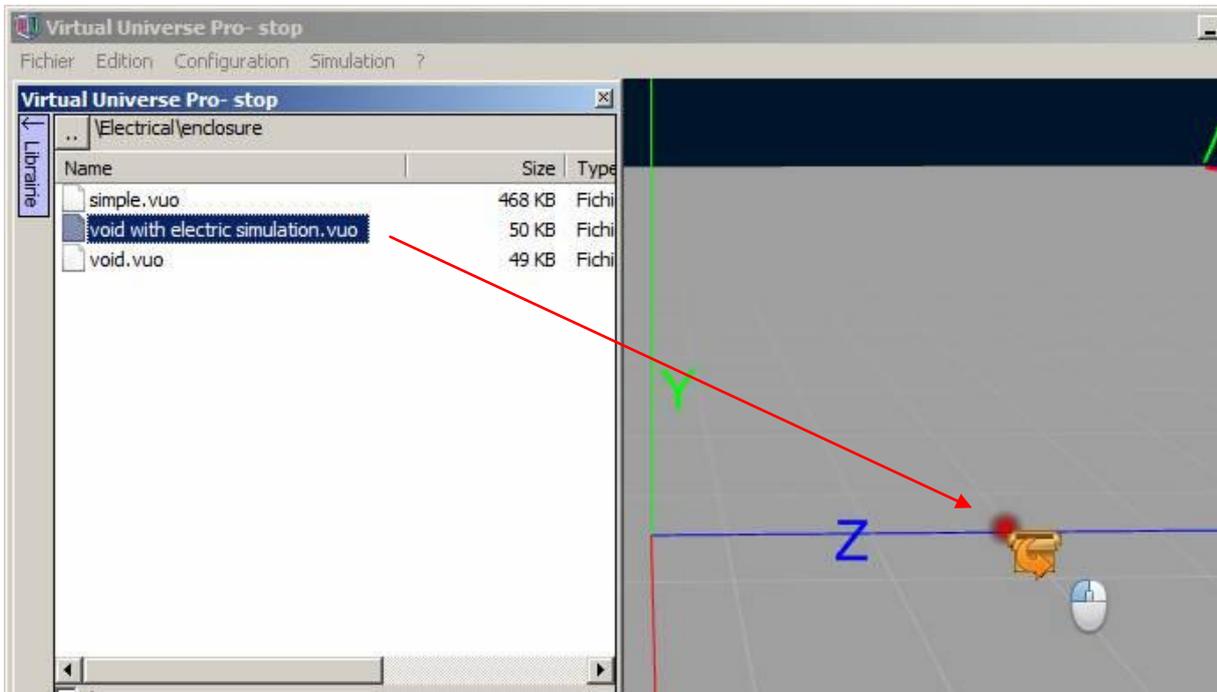
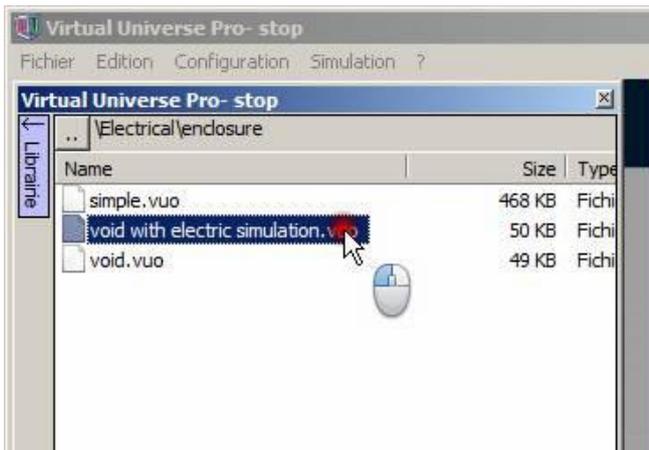
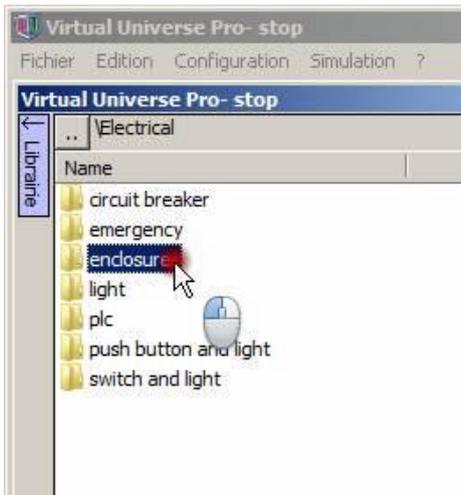
Tracé d'un lien sur un folio de simulation

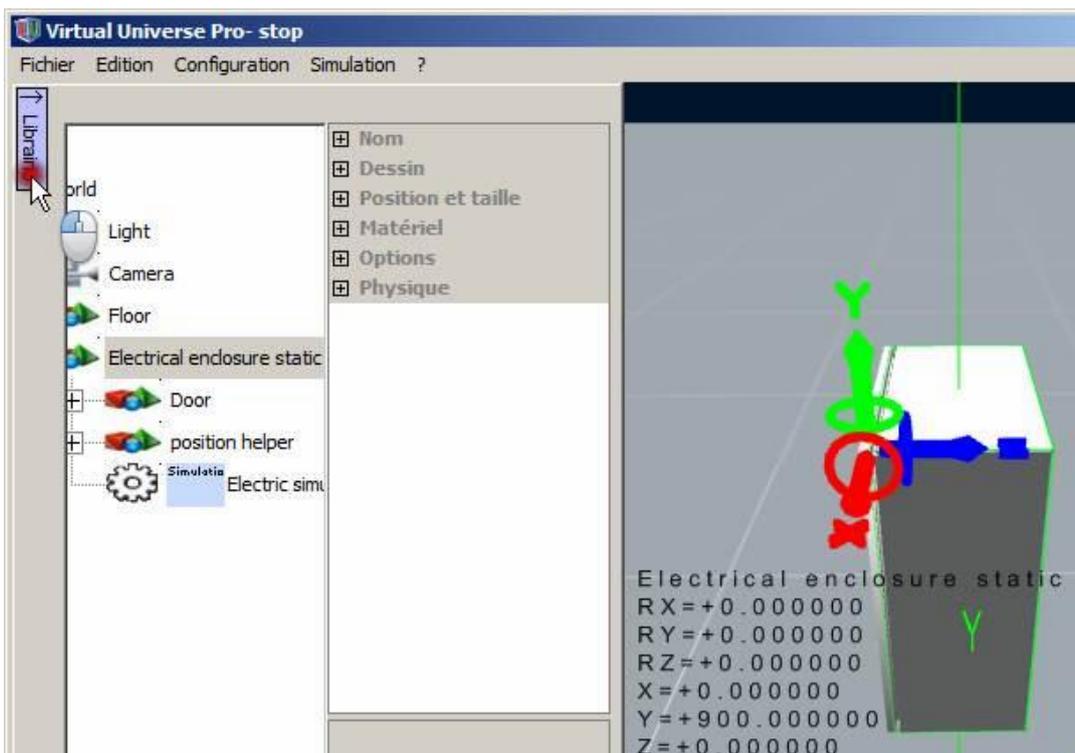
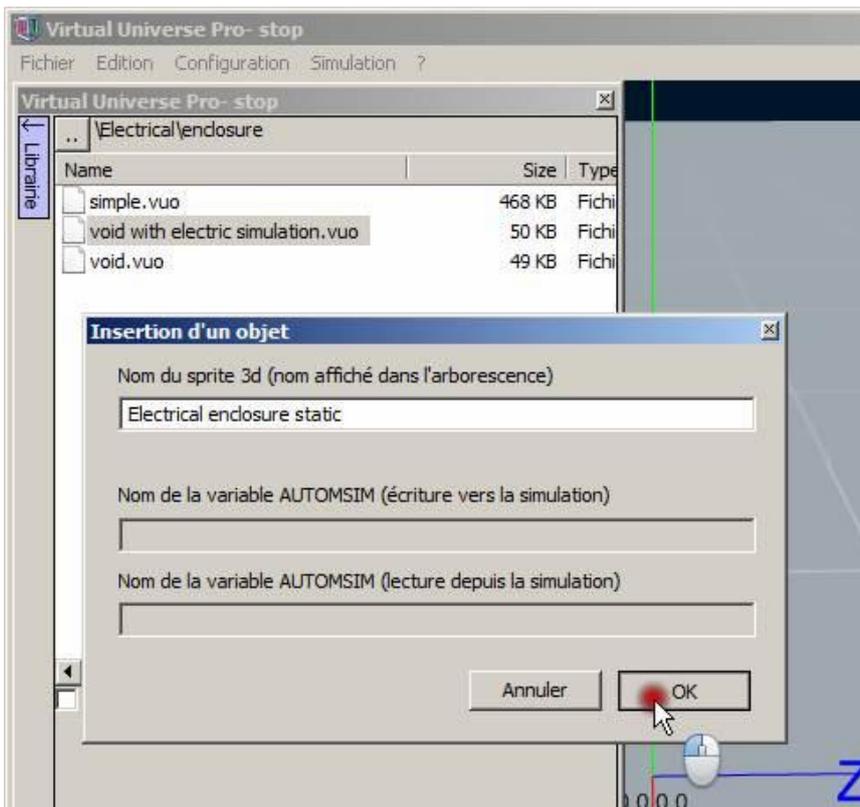


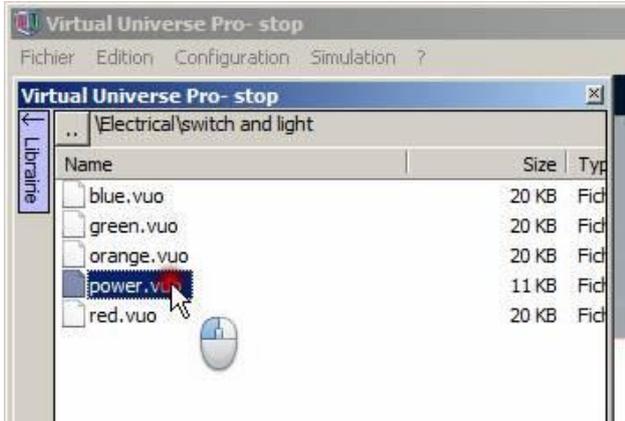
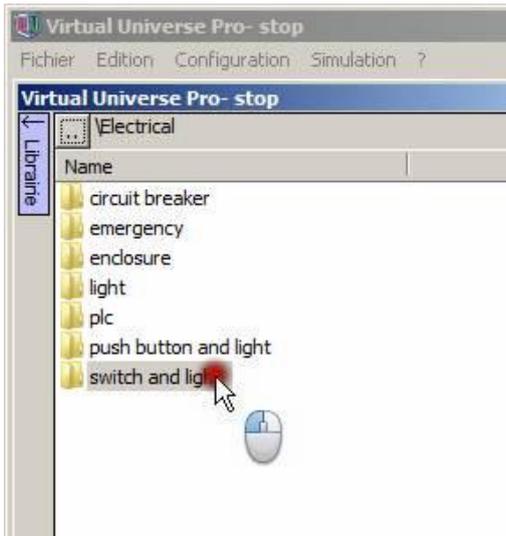
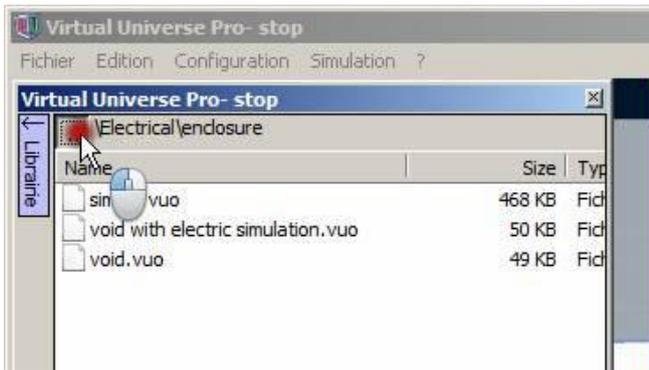
Génération automatisée d'un schéma

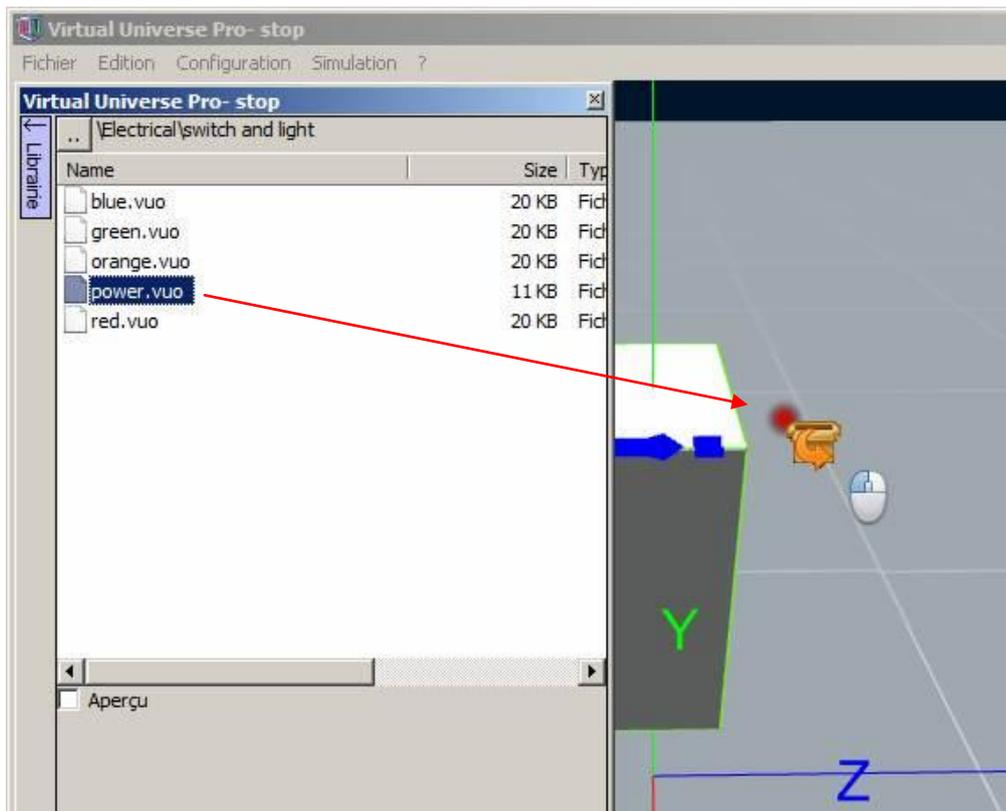
Certains éléments de la bibliothèque permettent de compléter un schéma électrique automatiquement en fonction des composants insérés. La procédure est la suivante :

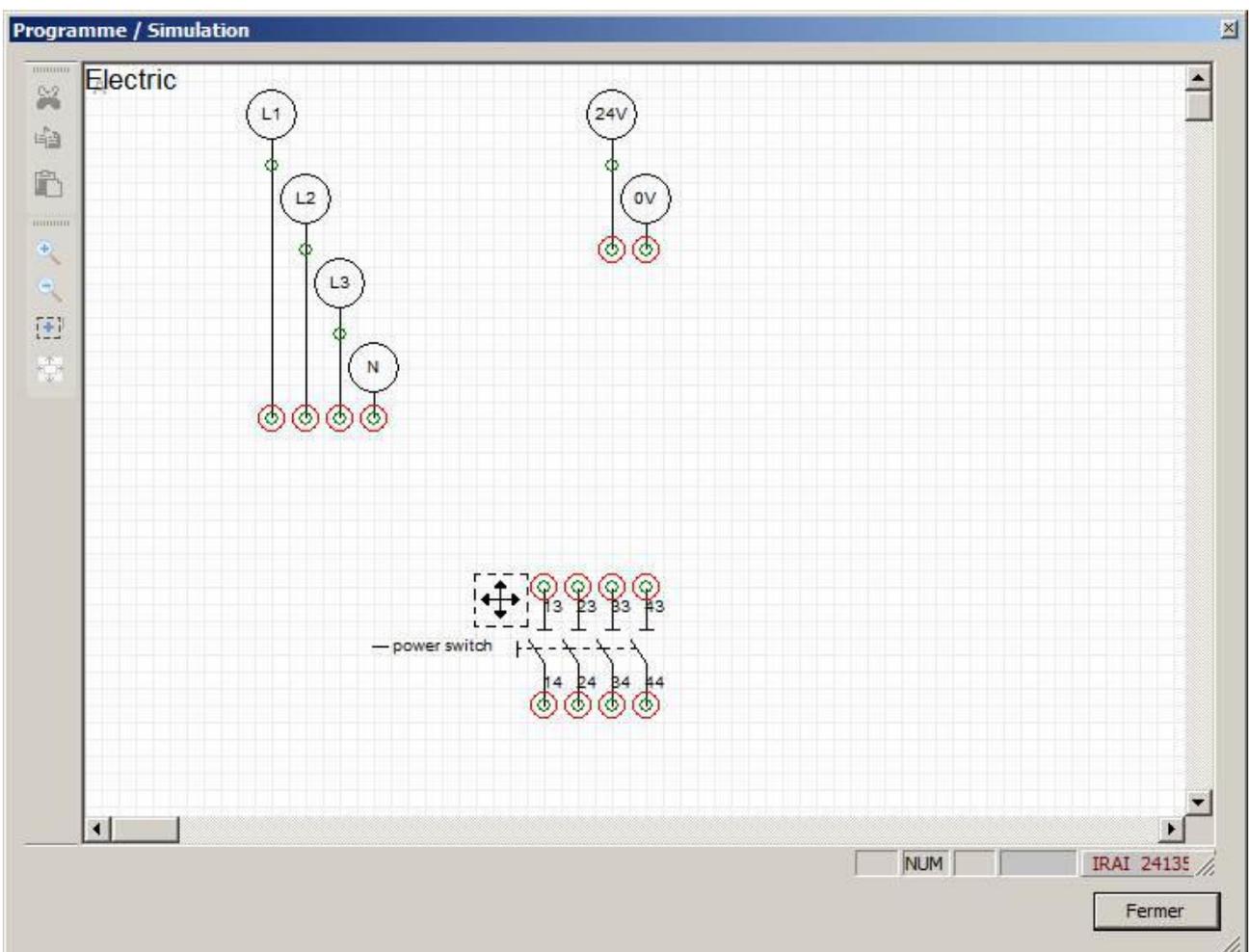
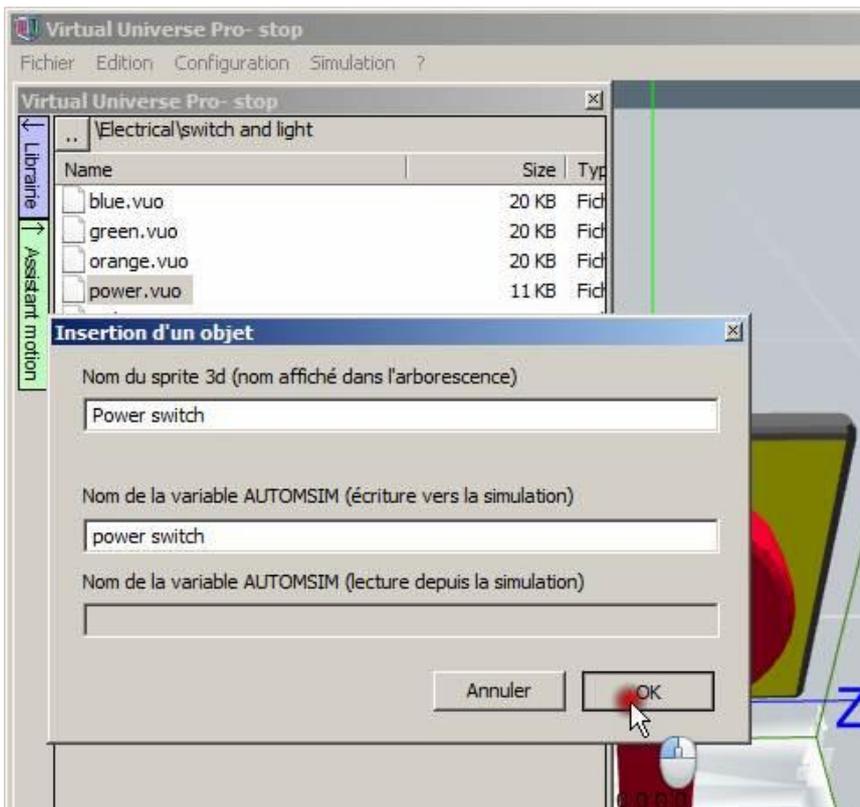


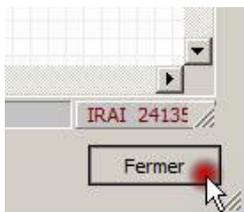
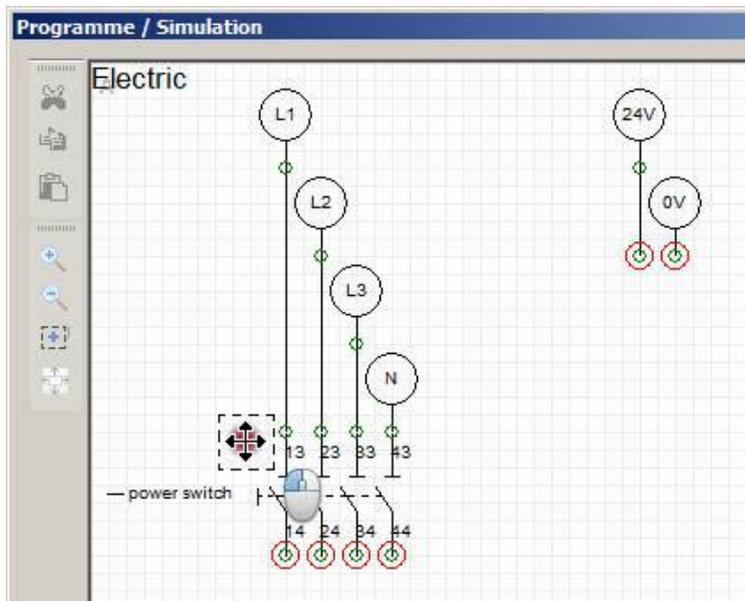
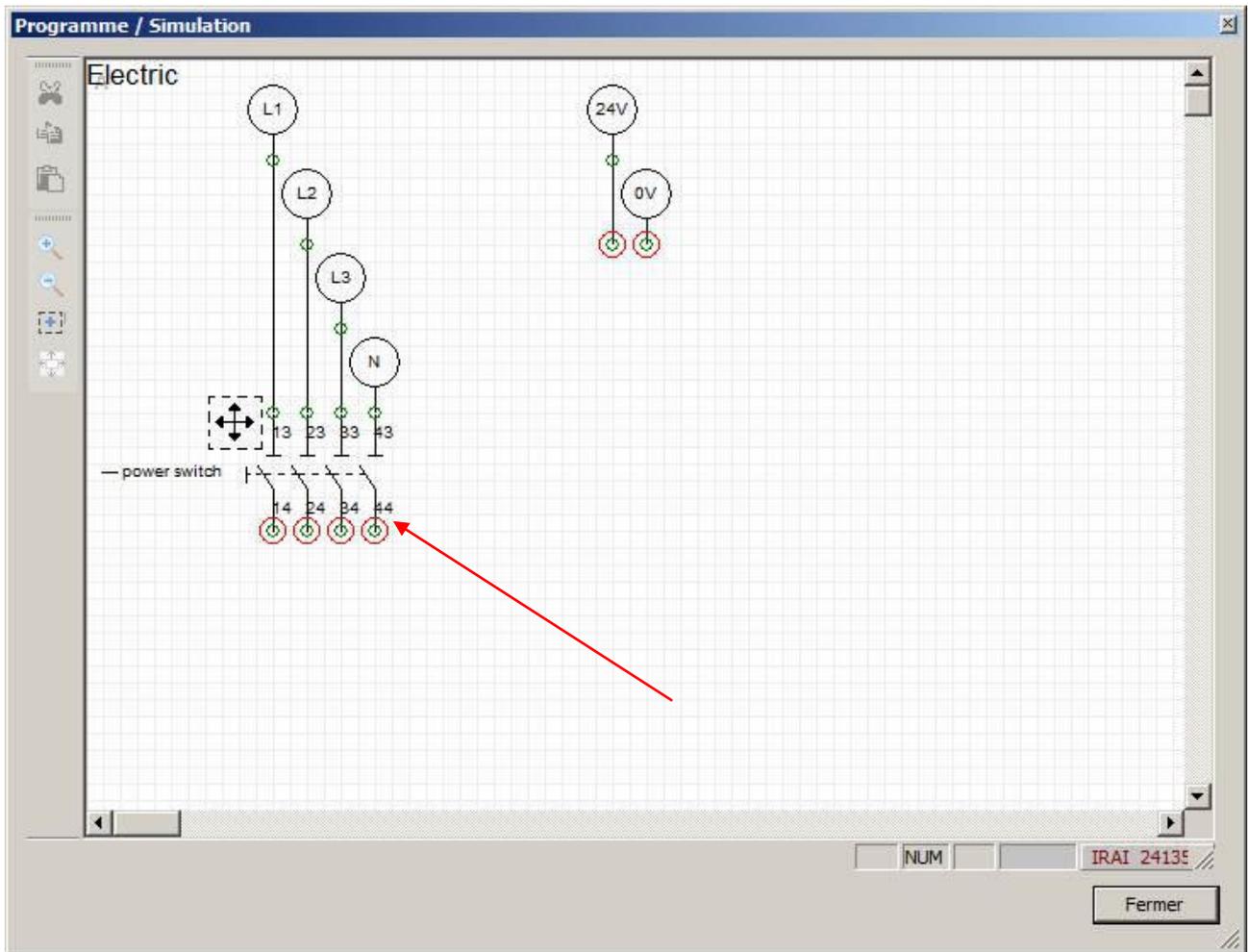


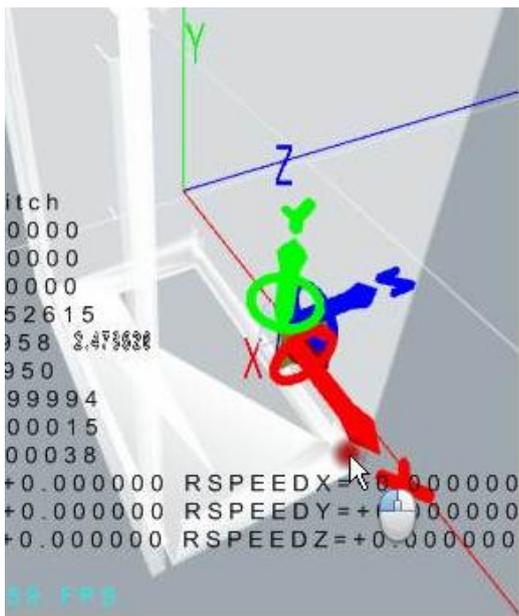
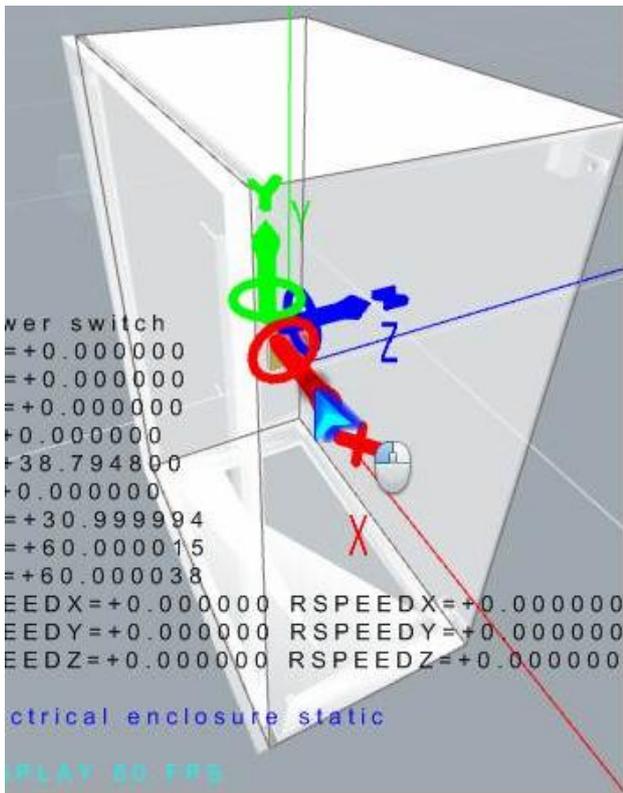


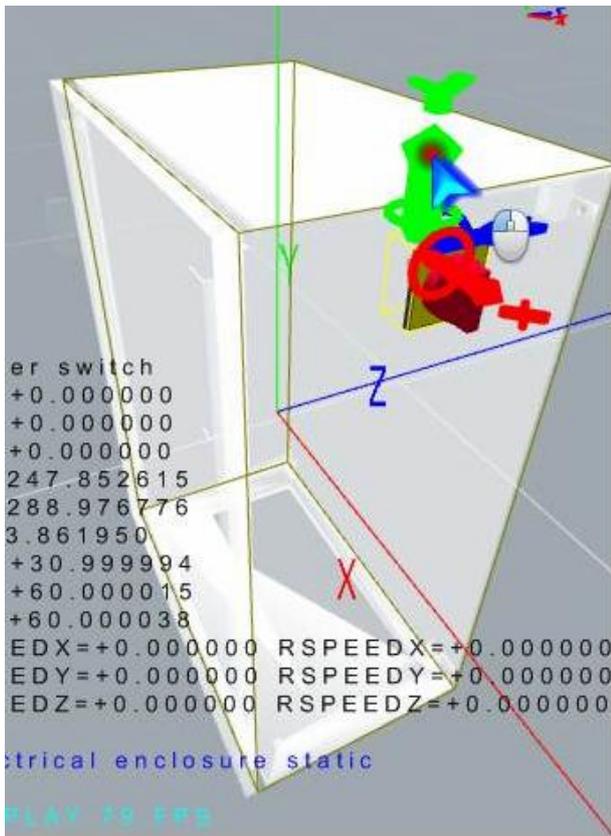




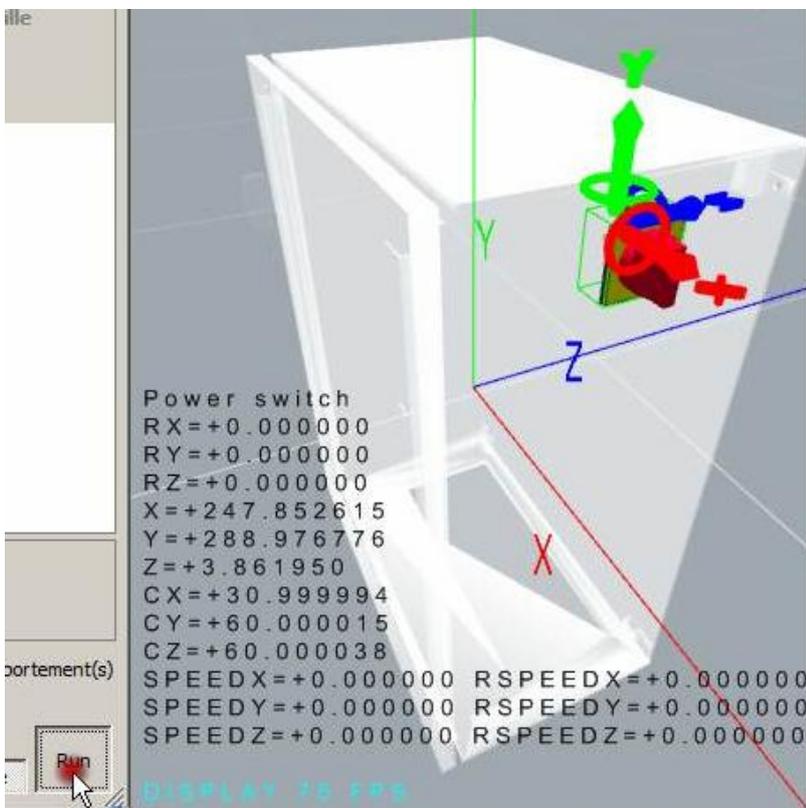


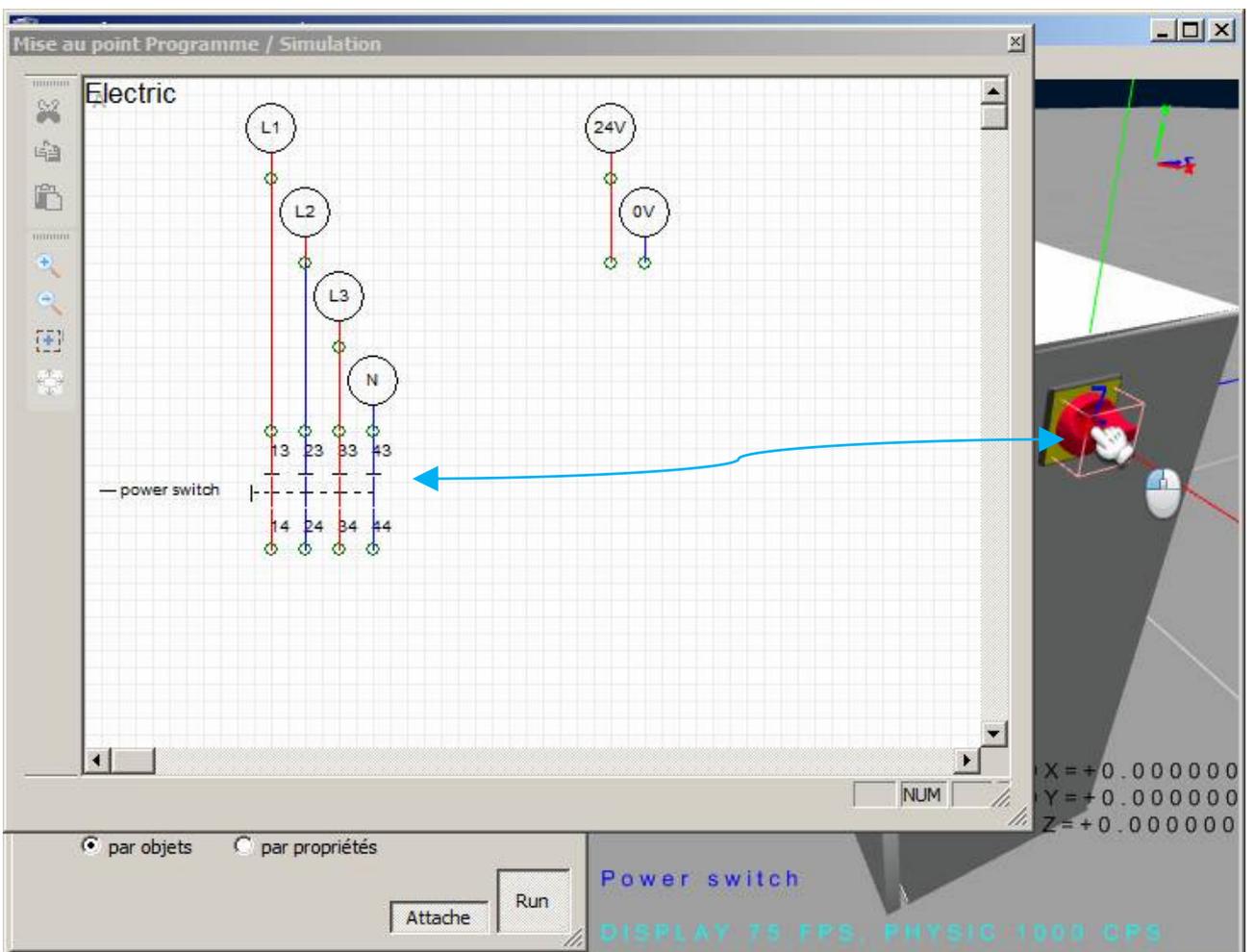
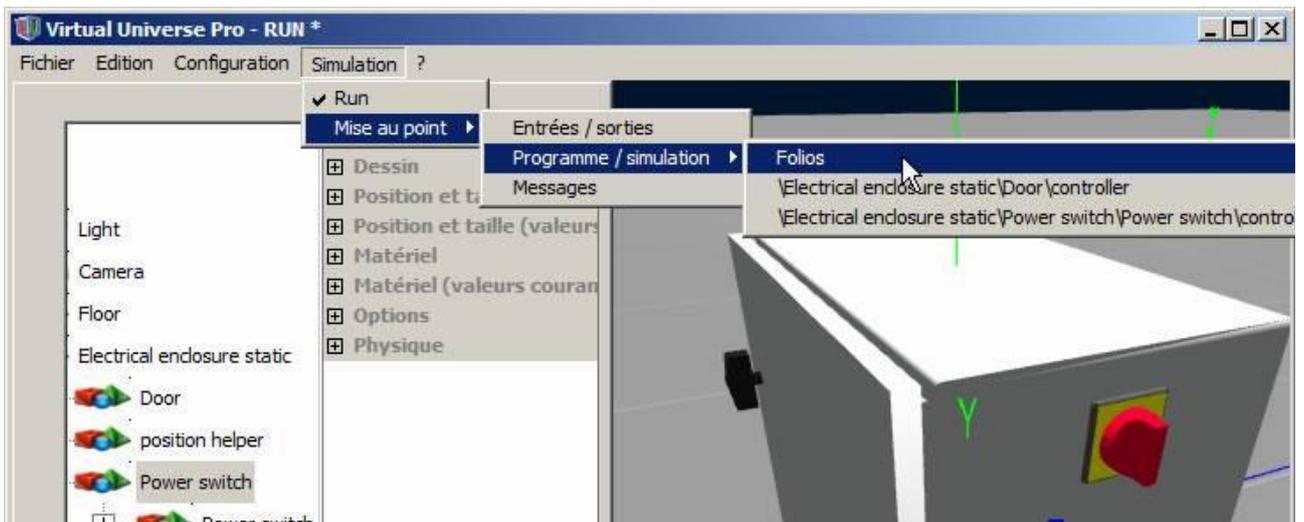






Laissez la touche ALT du clavier enfoncée pour déplacer librement l'objet sur la face du boîtier.





Les exemples se trouvant dans le sous-répertoire "Electric and pneumatic" du répertoire des exemples de VIRTUAL UNIVERSE PRO illustrent les fonctionnalités de simulation de schémas.

Définition d'interfaces pour les objets intelligents

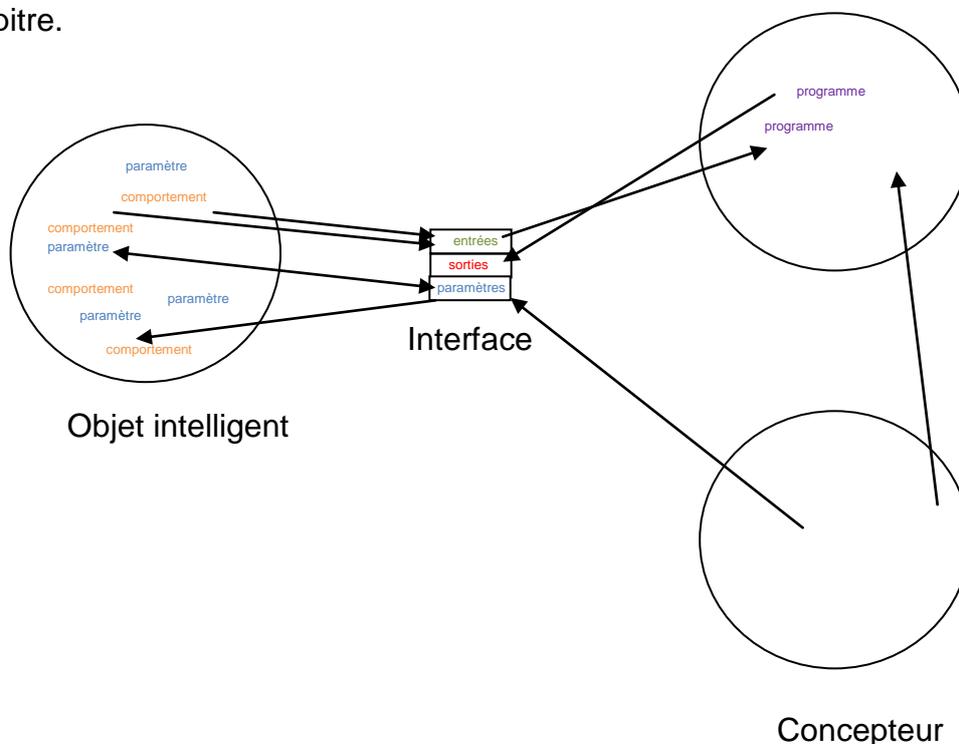
Généralités

Cette fonctionnalité permet de définir plus efficacement les entrées, sorties et paramètres d'un objet.

Lorsqu'une interface est définie pour un objet, alors il est plus facile d'accéder à ses variables et propriétés importantes. Le créateur de l'objet va en fait définir avec l'interface la liste des entrées, sorties et paramètres qu'il veut rendre accessible.

Les entrées et les sorties peuvent être utilisées dans les contrôleurs virtuels.

Rappel : dans VIRTUAL UNIVERSE PRO, les objets se composent d'un sprite 3d et de tous les objets enfants associés. Un tel objet se nomme un "objet intelligent" dans ce chapitre.



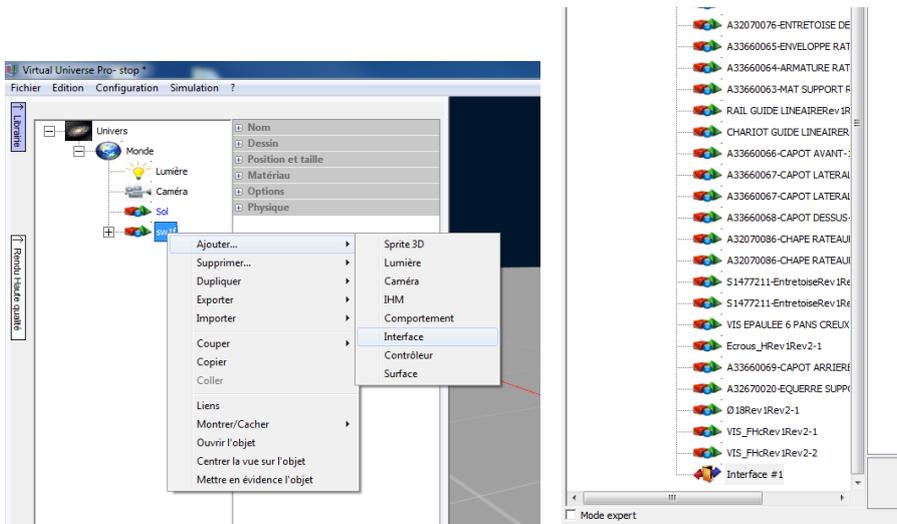
Les entrées de l'interface seront vues comme "action à faire" pour l'objet intelligent.

Les sorties de l'interface seront vues comme "résultats à fournir" pour l'objet intelligent.

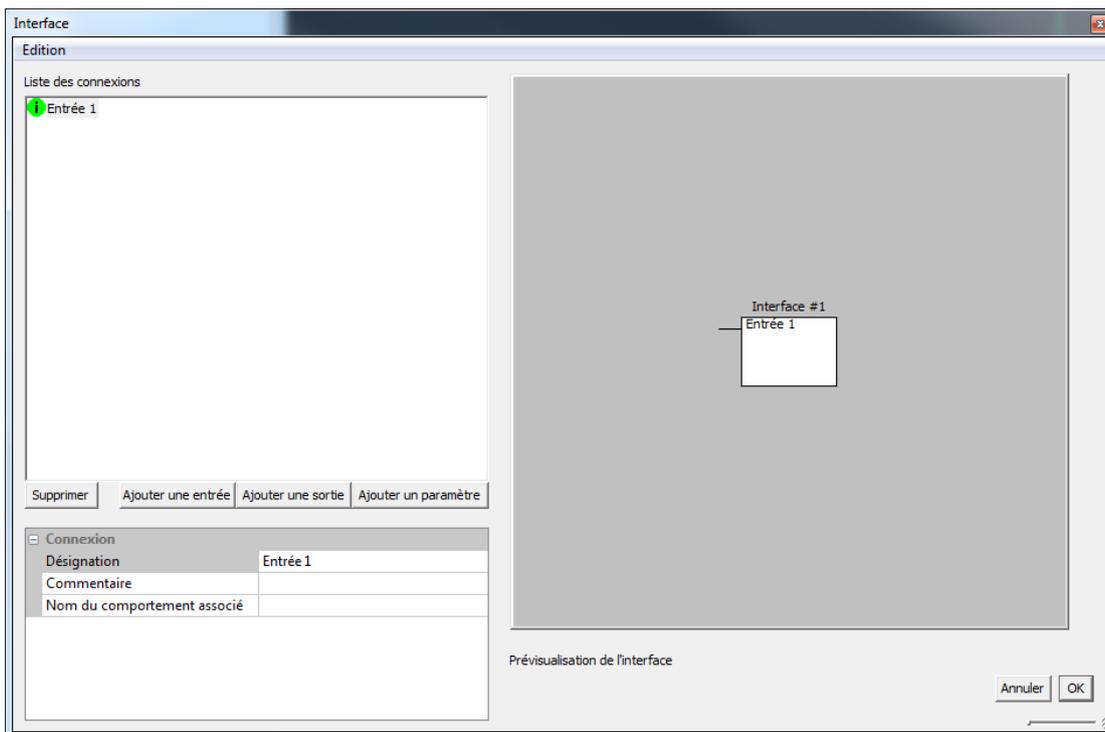
Les paramètres sont vus comme "configuration" par l'objet intelligent.

Utilisation

Un nouvel élément "interface" peut être ajouté comme enfant d'un sprite 3d.

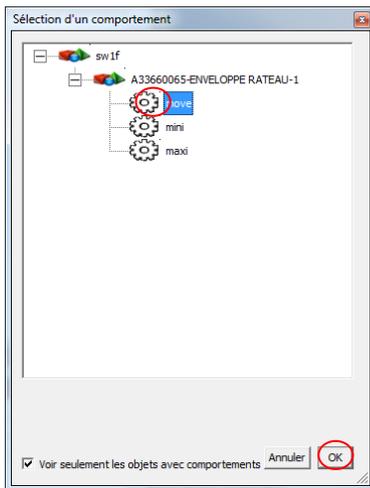
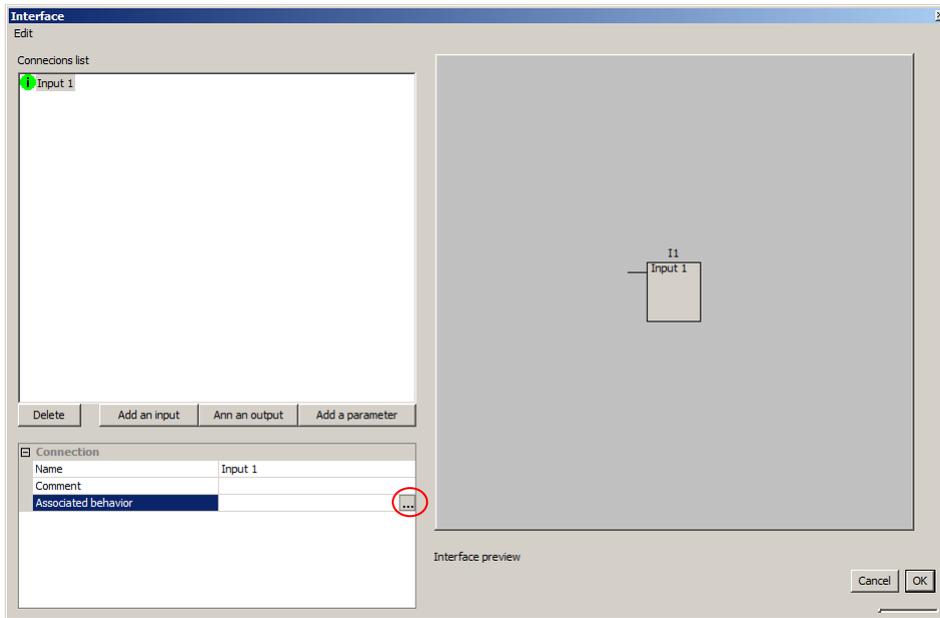


Dans cette nouvelle "Interface" on peut définir des entrées, des sorties et des paramètres :

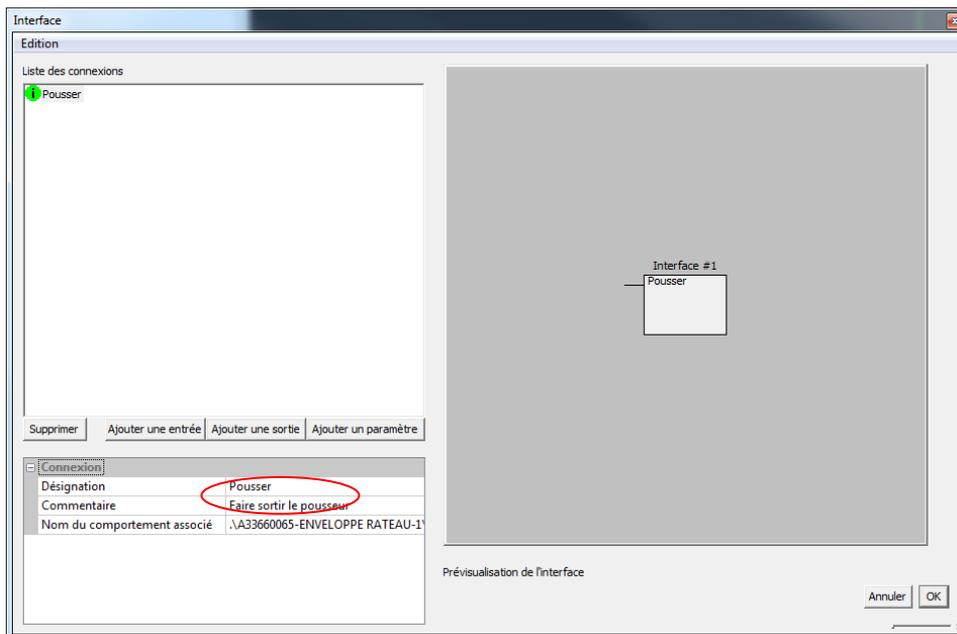


Un aperçu montre une représentation de type "bloc fonctionnel" de l'interface.

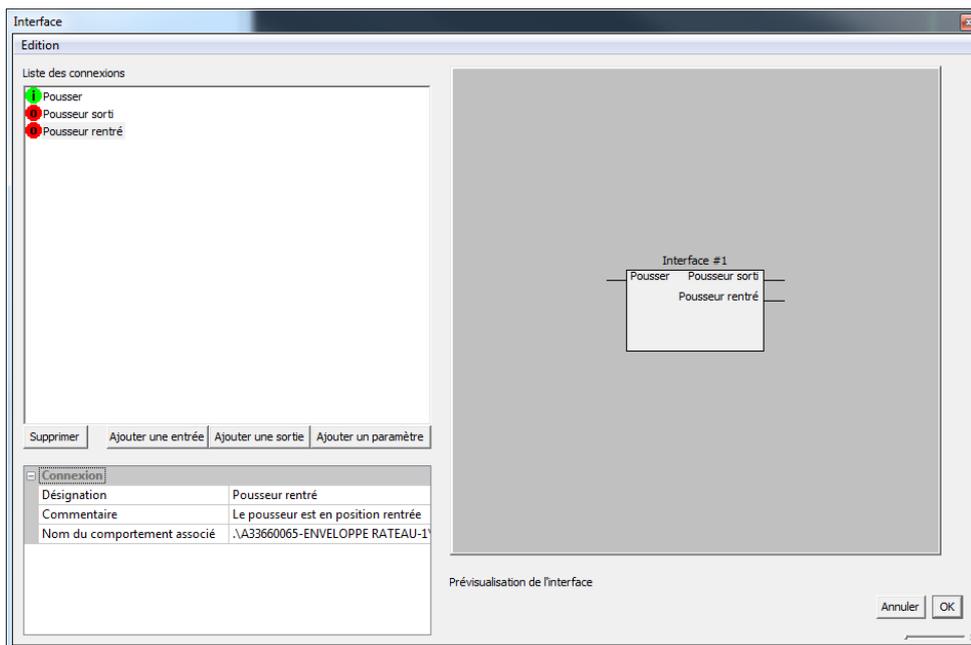
Les éléments entrées et les sorties doivent être associés à des comportements de l'objet intelligent.



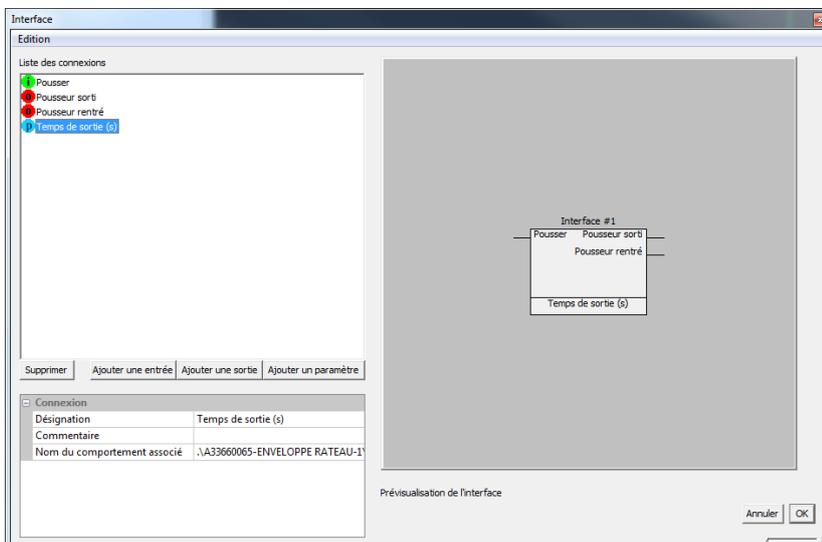
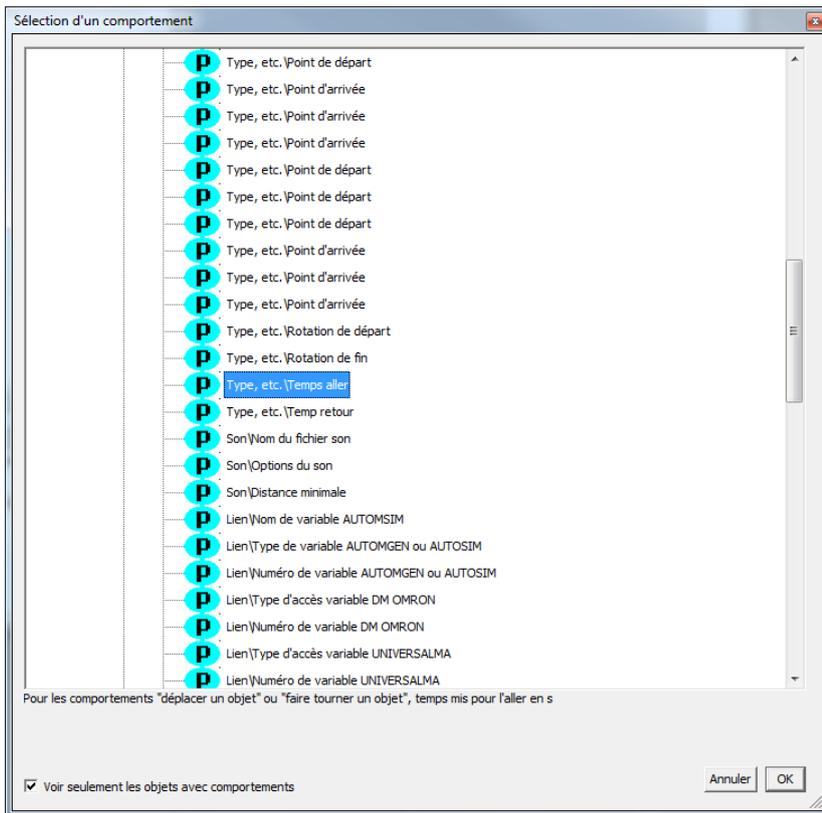
Un nom doit être associé, un commentaire peut également l'être :



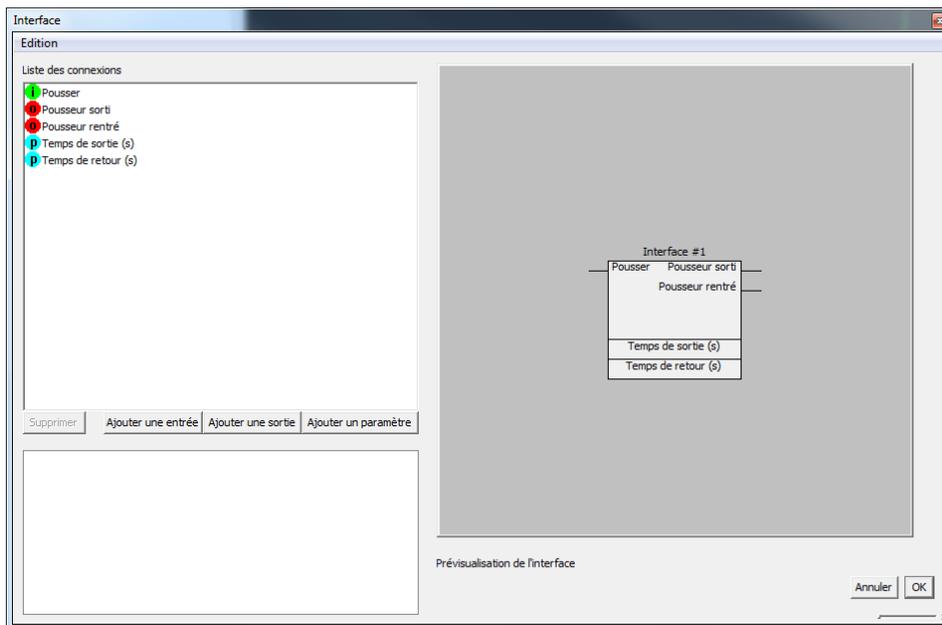
On peut de la même façon ajouter des sorties :



Les paramètres fonctionnent de la même façon excepté qu'ils sont associés à des propriétés des comportements :



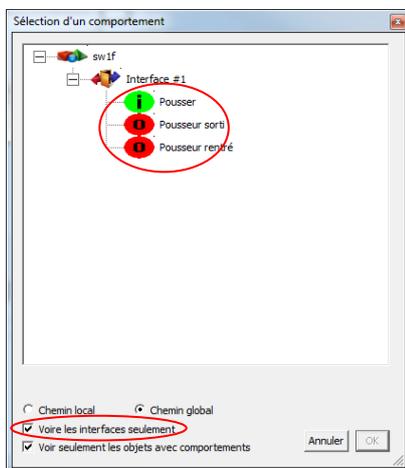
Voici notre objet intelligent avec l'interface entièrement définie :



Remarques :

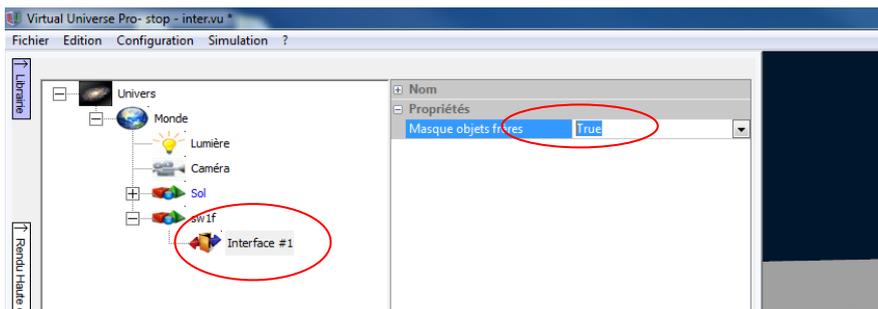
- Les objets intelligents peuvent être exportés avec leurs interfaces,
- Plusieurs interfaces peuvent être créées pour un même objet intelligent.

Accès aux entrées / sorties d'une interface depuis un contrôleur virtuel :

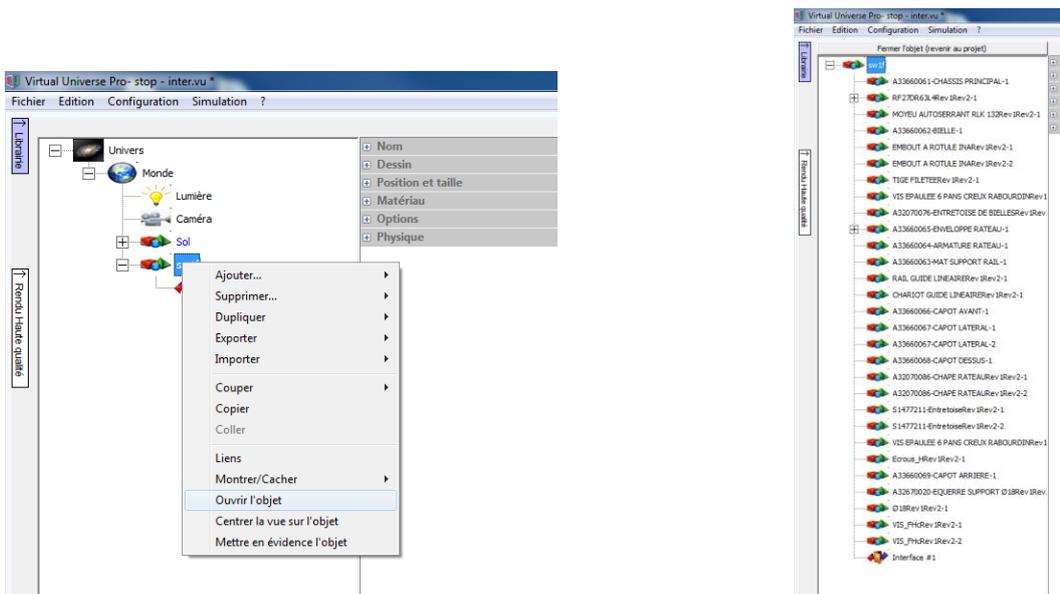


Lorsqu'on parcourt les comportements, sélectionner les entrées / sorties d'un objet intelligent possédant une interface est plus simple.

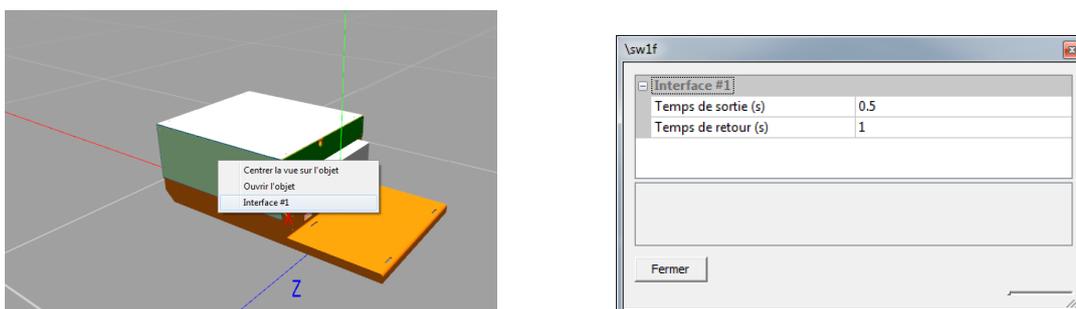
Une propriété des interfaces permet de cacher les enfants des objets intelligents.



La navigation dans l'arbre du projet s'en trouve simplifiée. Pour accéder aux sous objets, il est toujours possible d'utiliser le menu "Ouvrir l'objet" :



Pour accéder aux paramètres définis dans une interface, faire un simple clic droit sur l'objet dans la fenêtre de rendu et sélectionnez le nom de l'interface dans le menu. Une boîte de dialogue permet alors de modifier les paramètres :

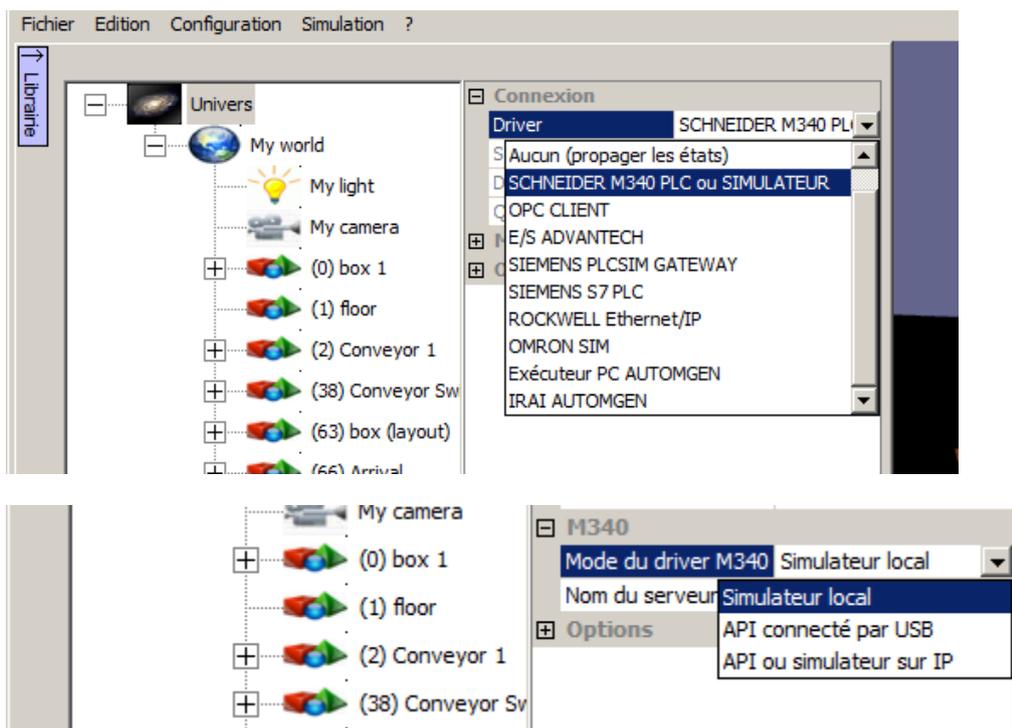


Un exemple complet se trouve dans le sous répertoire "\\Technical tips\\Interface" du répertoire des exemples.

Connexions externes

Le répertoire "PLC connection samples" du répertoire des exemples de VIRTUAL UNIVERSE PRO contient des exemples de projets VIRTUAL UNIVERSE PRO ainsi que des programmes créés avec différents ateliers logiciels constructeurs associés à ces exemples.

Connexion à un automate M340 ou au simulateur du logiciel Unity Pro de Schneider Electric



"Simulateur local" permet de communiquer avec le logiciel Unity Pro V4 ou supérieure fonctionnant sur le même PC.

"API connecté par USB" permet de dialoguer avec un automate M340 connecté au PC par un port USB

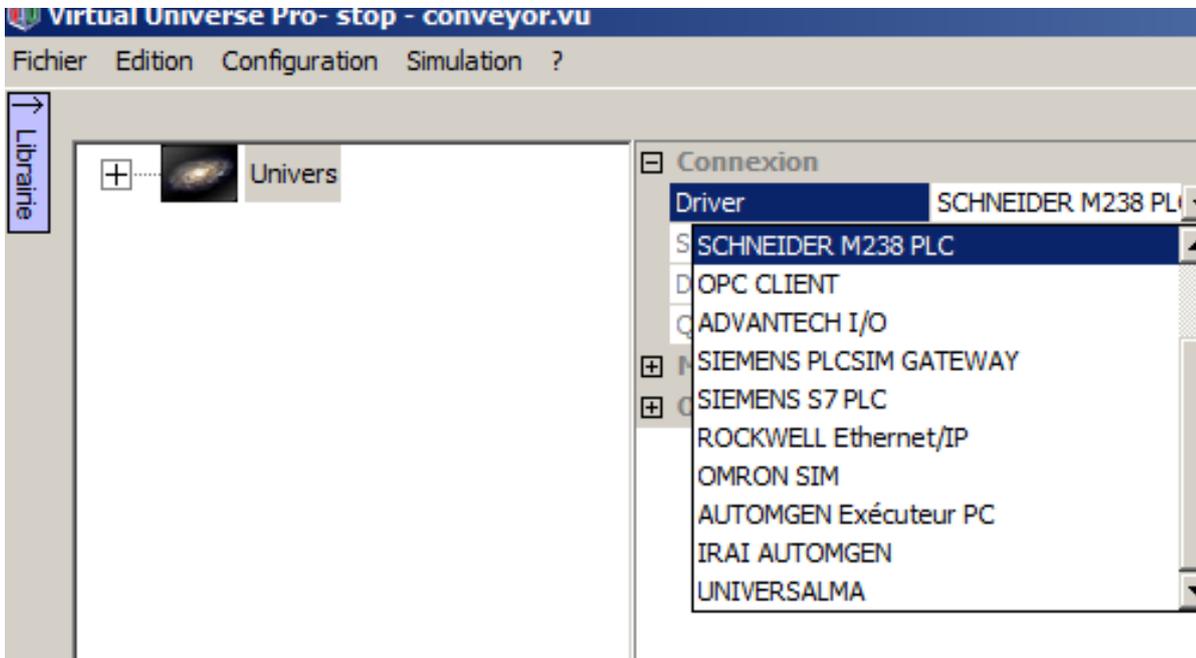
"API ou simulateur sur IP" permet de dialoguer avec un automate M340 connecté par Ethernet ou avec le logiciel Unity Pro V4 fonctionnant sur un autre PC relié par Ethernet. Pour cette option, le nom réseau ou l'adresse IP de l'automate ou du PC distant doit être renseigné sous la rubrique "Nom du serveur ou adresse IP".

Les variables utilisables dans les noms des liens sont les entrées, les sorties et les variables internes adressées sous la forme d'un bit, d'un mot de 16 bits, d'un mot de 32 bits ou d'un flottant. Par exemple : %I0.0.1, %MW10, %QW0.12, %MF15

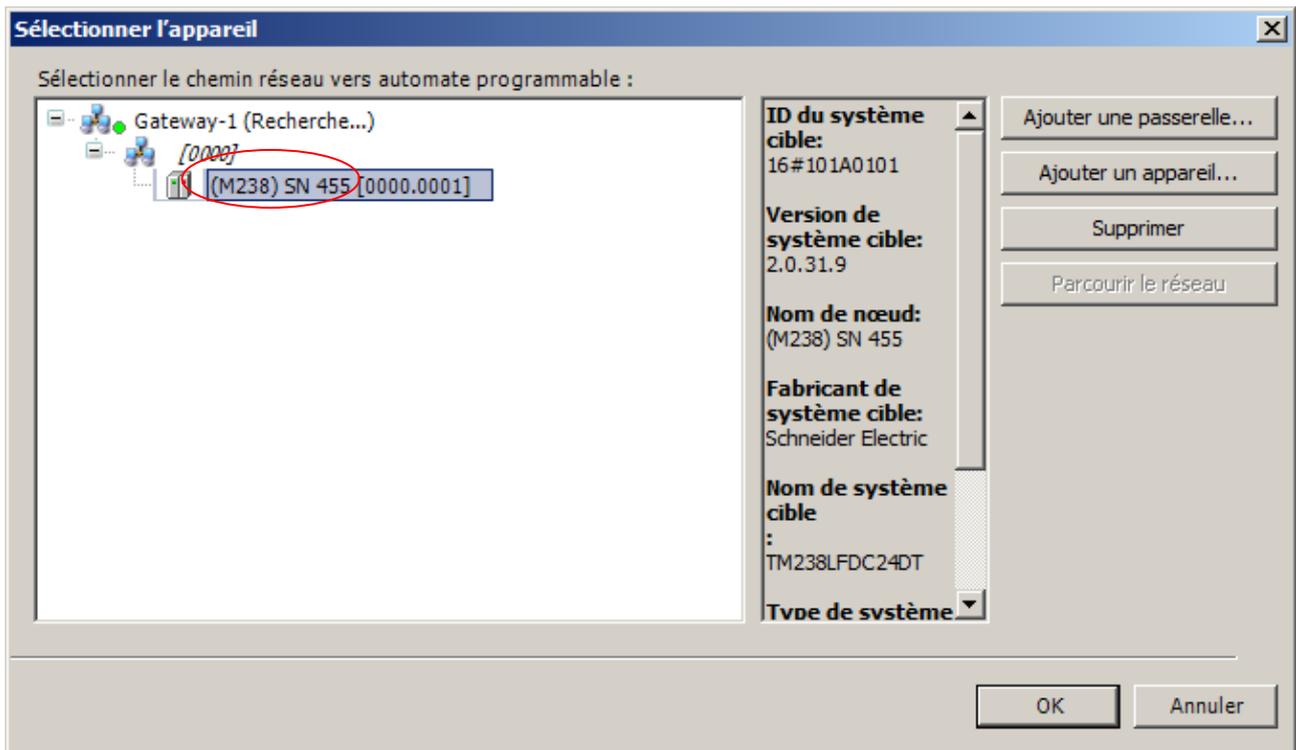
Limitation: l'émulateur du logiciel Unity Pro ne supporte pas l'accès aux variables d'E/S, il est dans ce cas obligatoire d'utiliser des variables internes pour les échanges entre Virtual Universe et Unity Pro.

Connexion à un automate m238 de Schneider Electric

Le logiciel SoMachine de Schneider Electric doit être installé sur le PC : VIRTUAL UNIVERSE PRO utilise la passerelle SoMachine gateway pour dialoguer avec l'automate. L'automate doit être connecté au PC. Le logiciel SoMachine et VIRTUAL UNIVERSE PRO peuvent être connectés en même temps à l'automate m238.



Le nom de l'automate doit être le même que celui sélectionné dans le logiciel SoMachine lors de la connexion :



Fermez ensuite la fenêtre "Configuration des symboles", recompilez l'application et transférez la dans l'automate.

La syntaxe à utiliser dans Virtual Universe pro pour référencer les variables de l'automate m238 est la suivante :

<nom d'application>.<nom de POU ou GVL pour les variables globales>.<nom de variable>

Par exemple:

Application.GVL.mavariable

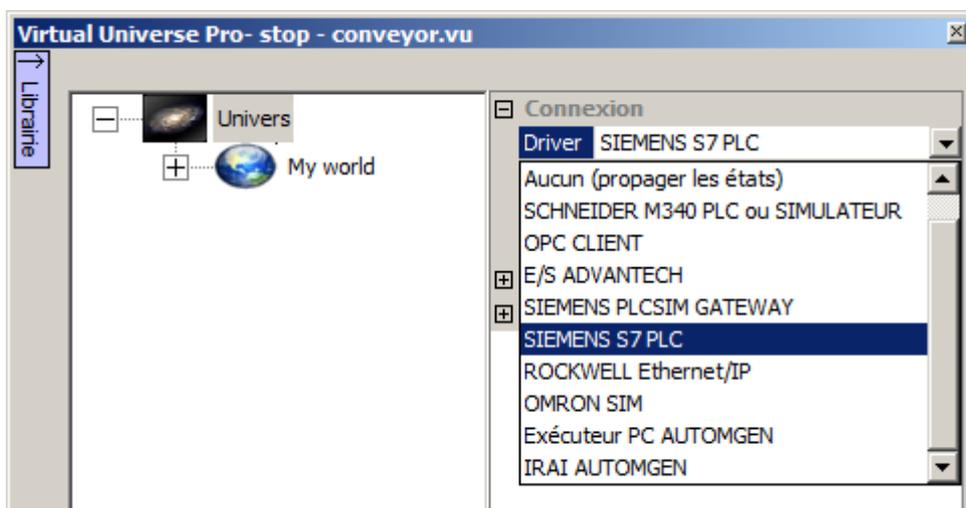
qui signifie : variable globale nommée "mavariable" de l'application "Application".

MonApplication.MonPOU.uneautrevariable

qui signifie : variable du module "MonPOU" de l'application "MonApplication" nommée "uneautrevariable".

Connexion à un automate Siemens S7-1200, S7-1500, S7-300 ou S7-400

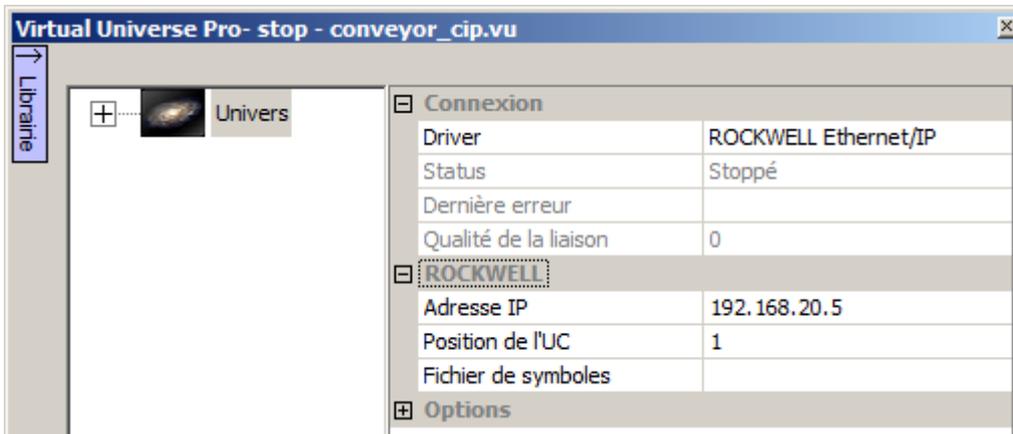
Le PC et l'automate doivent être reliés à un réseau Ethernet.



L'adresse IP de l'automate doit être renseignée sous la rubrique S7/Adresse IP. Pour les automates S7-400, la position de l'UC dans le rack de l'automate doit également être renseignée sous la rubrique S7/Position de l'UC.

Les variables utilisables dans les noms des liens sont les entrées, les sorties et les variables internes adressées sous la forme d'un bit, d'un mot de 16 bits, d'un mot de 32 bits ou d'un flottant. Par exemple : %I0.0, %MW10, %QW0, %MF15

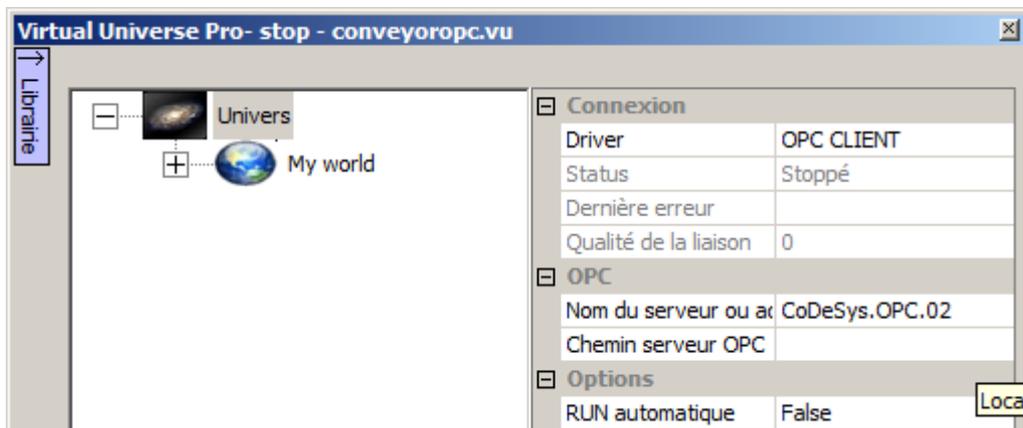
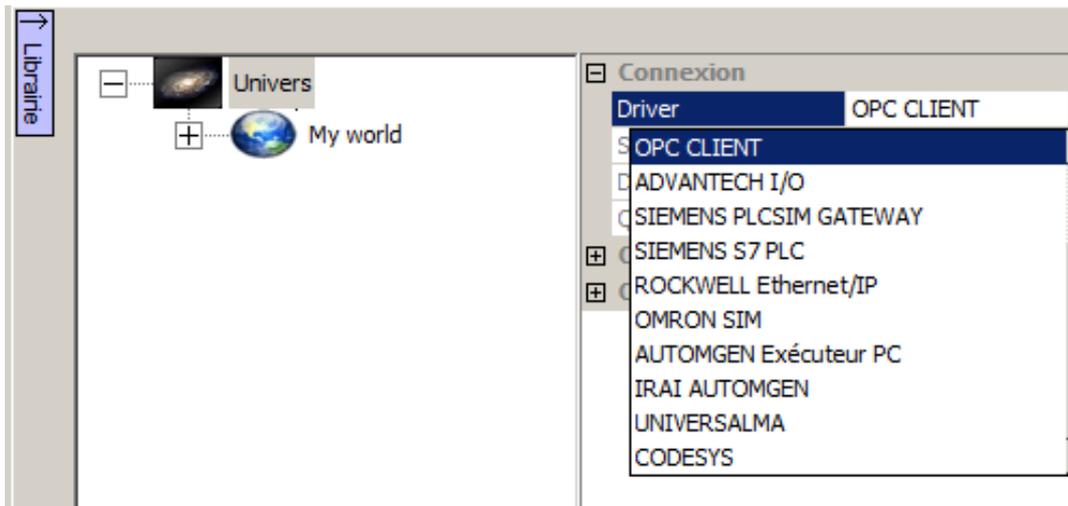
Connexion à un automate Rockwell Compact Logix, Control Logix ou à un émulateur Softlogix



L'adresse IP de l'automate ou du PC où est lancé Softlogix doit être renseignée ainsi que la position de l'UC de l'automate.

Les variables utilisables dans les noms des liens sont les symboles "Tags" utilisés dans l'application Rockwell s'exécutant dans l'automate ou l'émulateur. Pour un Tag appartenant à un programme, la syntaxe PROGRAM:<nom de programme>.<tag> doit être utilisée, par exemple : PROGRAM:MYPROGRAM:MYTAG

Connexion à un serveur OPC

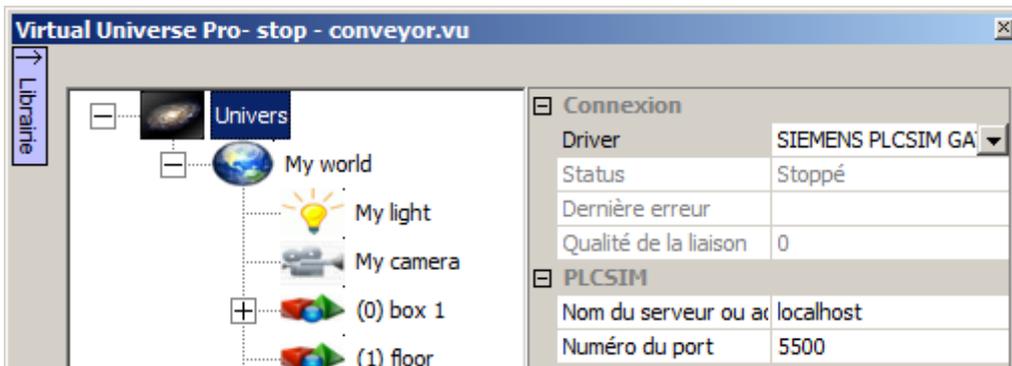
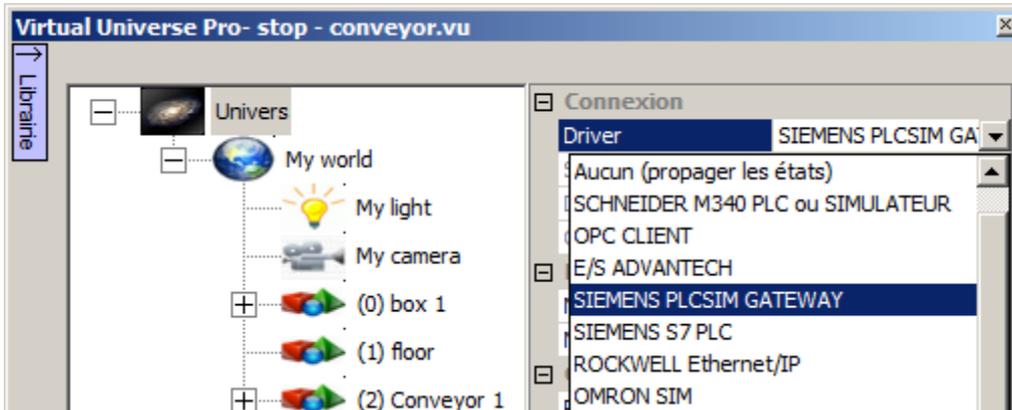


Le nom du serveur OPC doit être renseigné, si le serveur OPC se trouve sur le même PC, la rubrique OPC/Chemin serveur OPC doit être laissée vide, pour un serveur distant, le chemin réseau ou l'adresse IP doit être renseignée.

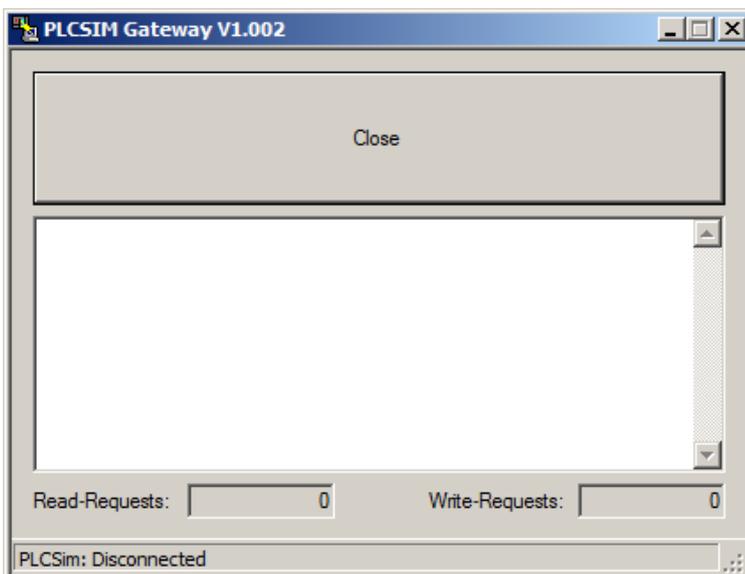
Les variables utilisables dans les noms des liens sont les noms de variables OPC.

Connexion à l'émulateur PLCSIM de Siemens

Cas de PLCSIM 5.4 SP<5



Le programme "PLCSimGateway.exe" qui se trouve dans le répertoire d'installation de VIRTUAL UNIVERSE PRO doit être lancé. La rubrique "Nom du serveur ou adresse IP" doit être renseignée avec le nom réseau ou l'adresse IP du PC où PLCSimGateway.exe est lancé (utiliser "localhost" si c'est le même PC).



Cas de PLCSIM 5.4 SP>=5

Dans ce cas, l'émulateur PLC Sim est considéré comme un automate Siemens.

Le programme NetToPlcSim.exe qui se trouve dans le répertoire d'installation de VIRTUAL UNIVERSE PRO doit être lancé, il permet de créer une passerelle permettant de dialoguer avec PLCSIM comme avec un automate S7300 ou S7400 équipé d'un coupleur Ethernet.

Le projet créé dans STEP7 doit être défini avec une CPU possédant un coupleur Ethernet.

L'interface PG/PC sélectionnée doit être PLCSIM TCP-IP.

NetToPlcSim.exe configure et active automatiquement un lien de communication entre un coupleur d'automate géré dans PLCSIM avec une adresse 192.168.0.1 et une adresse locale 127.0.0.1. VIRTUAL UNIVERSE PRO peut alors se connecter avec le driver S7 PLC en mode IP avec comme adresse 127.0.0.1.

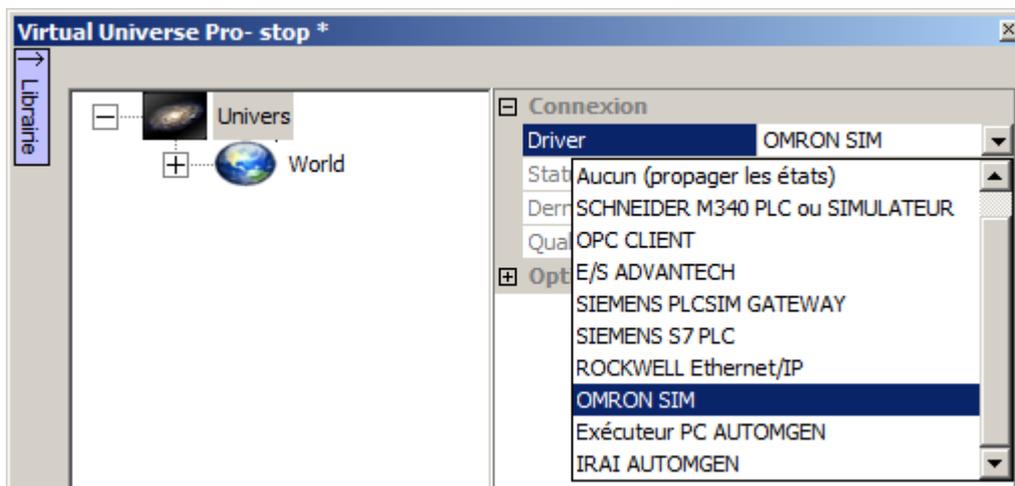
Les variables utilisables dans les noms des liens sont les entrées, les sorties et les variables internes adressées sous la forme d'un bit, d'un mot de 16 bits, d'un mot de 32 bits ou d'un flottant. Par exemple : %I0.0, %MW10, %QW0, %MF15

Connexion à l'émulateur CX-Simulator d'Omron

Le driver pour le dialogue avec l'émulateur CX-Simulator d'Omron utilise la technologie d'accès direct à la mémoire du simulateur. Cette technologie permet de lire et d'écrire les variables Omron de type DM. Si d'autres type de variables doivent être écrits ou lus, des instructions de recopie devront être ajoutées dans le programme créé avec CX-Programmer. La localisation des variables DM nécessite également de prédisposer les 2 mots DM constituant la zone d'échange avec des valeurs particulières, ces valeurs servant de signature et permettant au driver de localiser la table. La signature est utilisée uniquement à la connexion du driver. Les 2 mots de signatures représentent les mot 0 et 1 de la table, le mot suivant le mot 2, etc.

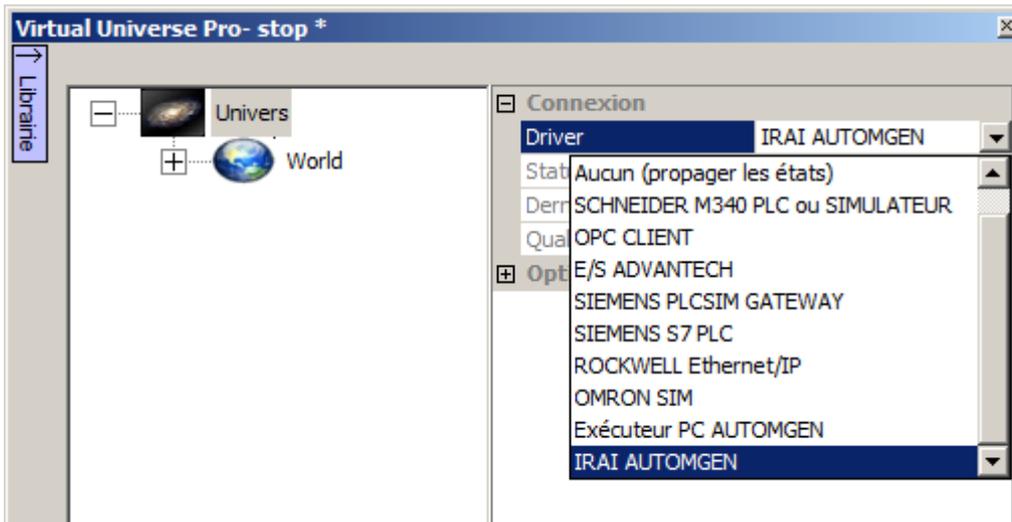
Premier mot de signature (position 0 de la table) (valeur hexa 4952)	Deuxième mot de signature (position 1 de la table) (valeur hexa 4149)	Troisième mot (position 2 de la table)			
--	---	--	--	--	--

Suivant la version de Windows utilisée, cette technique peut nécessiter une exécution de VIRTUAL UNIVERSE PRO en mode administrateur, pour lancer VIRTUAL UNIVERSE PRO en mode administrateur, cliquez avec le bouton droit de la souris sur le raccourci de lancement de Virtual Universe et sélectionnez "Exécuter en tant qu'administrateur".

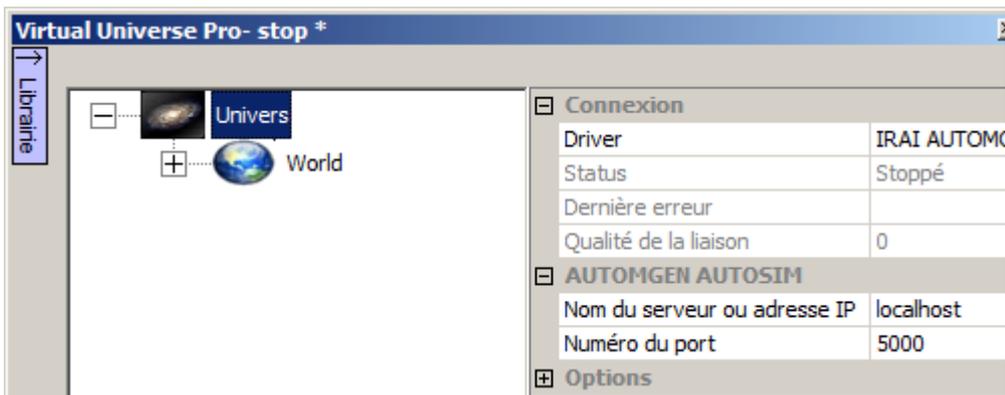


La définition des liens se fait en précisant le numéro du mot DM à partir de la table et le type d'accès. Le type d'accès précise si on accède à un des bits d'un mot DM, au mot DM entier (entier de 16 bits) ou à un groupe de 2, 4 ou 8 mots DM consécutifs considérés comme des entiers ou des flottants, pour ce dernier cas, c'est le numéro du premier mot du groupe qui est spécifié.

Connexion à AUTOMGEN



L'option Exécution/Connexion TCP-IP/Serveur dans les propriétés du projet AUTOMGEN doit être cochée.

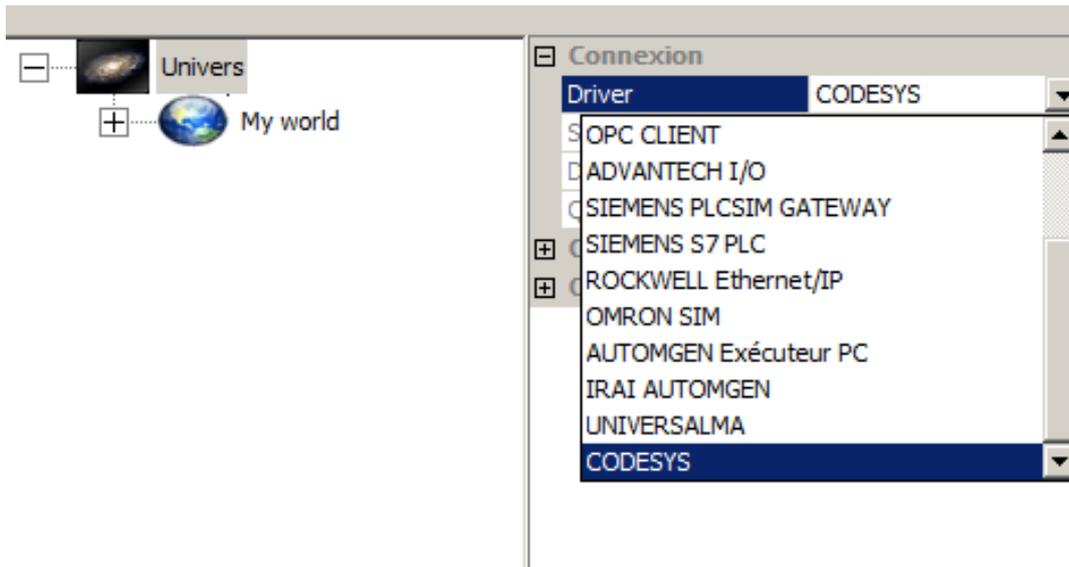


Si AUTOMGEN fonctionne sur un autre PC, le nom réseau ou l'adresse IP de ce PC distant doit être renseigné sous la rubrique Nom du serveur ou adresse IP.

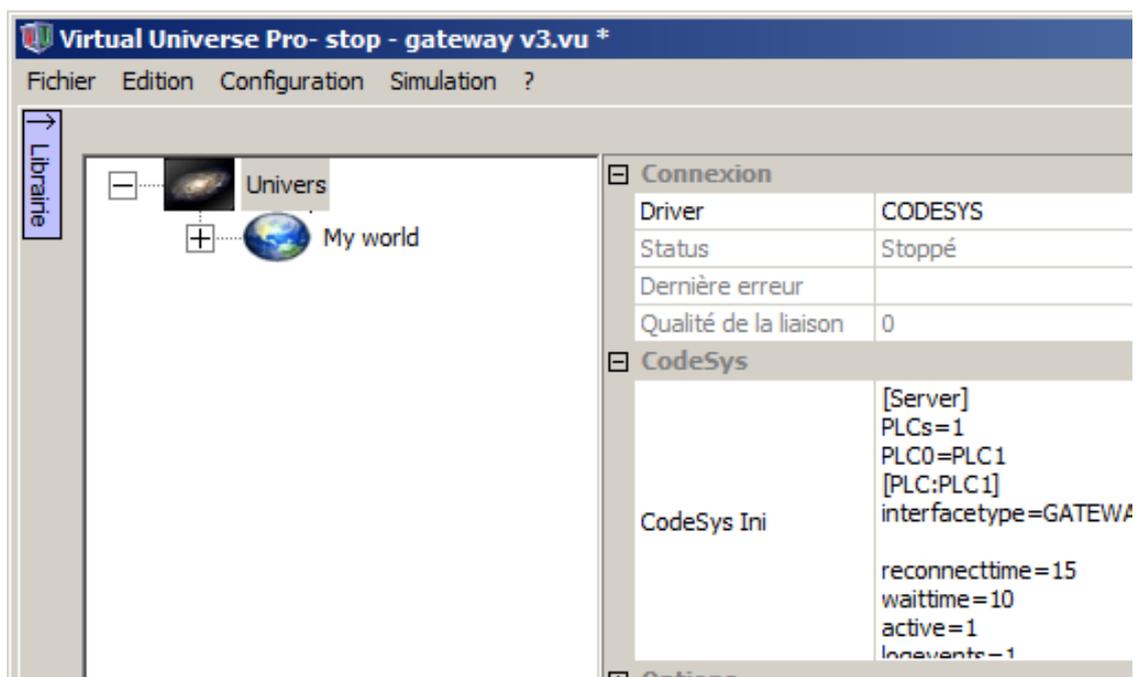
Les variables utilisables dans les liens sont l'ensemble des variables AUTOMGEN.

Le driver "Exécuteur PC d'AUTOMGEN" permet une connexion plus rapide dans le cas de l'utilisation de l'exécuteur PC d'AUTOMGEN.

Connexion à CoDeSys

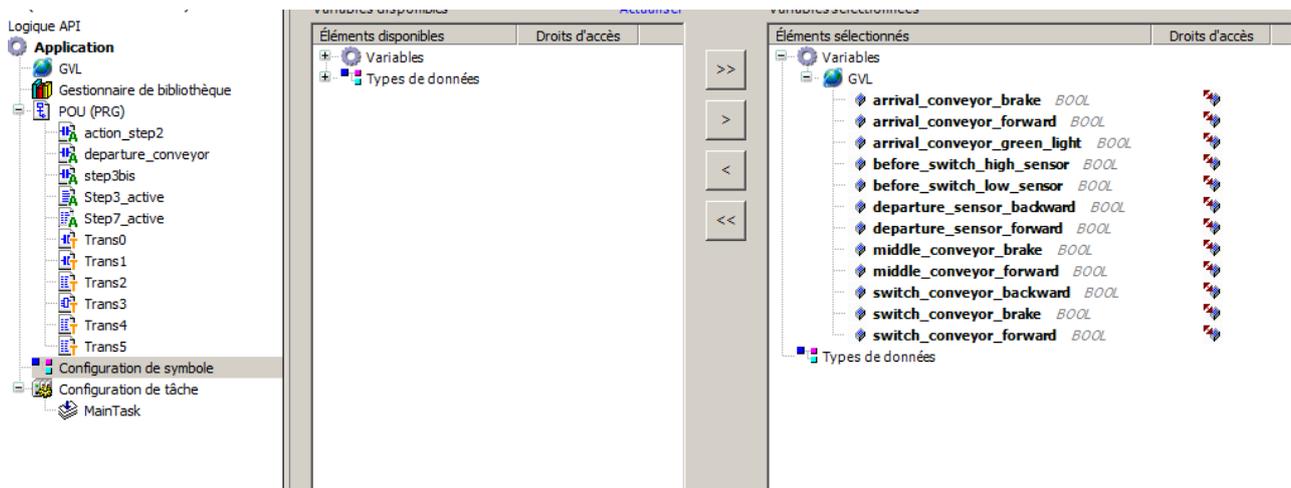


Cette connexion permet un accès aux cibles compatibles CoDeSys V2 et V3. L'élément "CodeSys Ini" doit contenir la configuration de la connexion telle que définie dans les fichiers .INI de CoDeSys.



Un exemple de connexion pour la passerelle Gateway 2 et la passerelle Gateway 3 et les exemples CoDeSys associés se trouvent dans le répertoire des exemples de VIRTUAL UNIVERSE PRO.

Les variables du projet CoDeSys auxquelles VIRTUAL UNIVERSE PRO doit pouvoir accéder doivent être placées dans un élément "Configuration de symboles" :



La syntaxe à utiliser dans Virtual Universe pro pour référencer les variables CoDeSys est la suivante :

<nom d'application>.<nom de POU ou GVL pour les variables globales>.<nom de variable>

Par exemple:

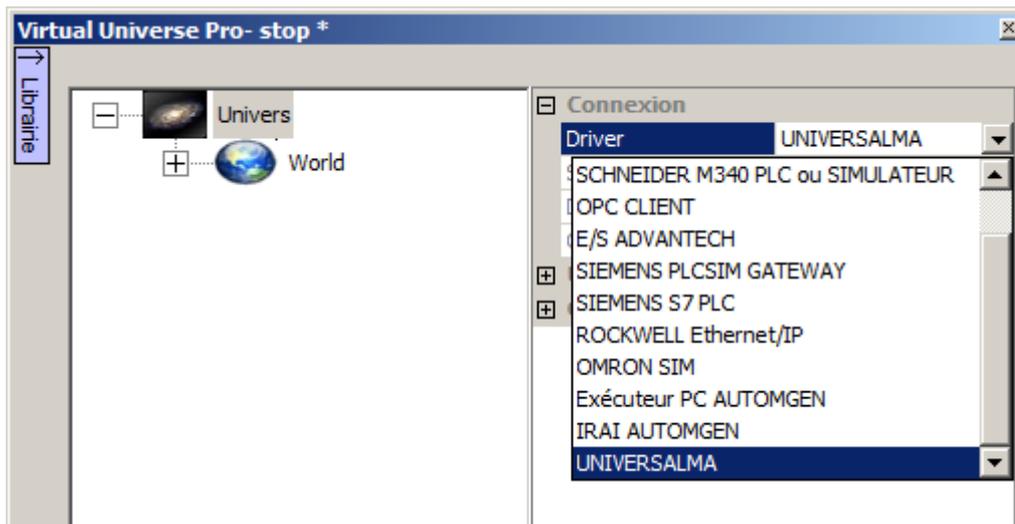
Application.GVL.mavariable

qui signifie : variable globale nommée "mavariable" de l'application "Application".

MonApplication.MonPOU.uneautrevariable

qui signifie : variable du module "MonPOU" de l'application "MonApplication" nommée "uneautrevariable".

Connexion universelle



Cette connexion permet un accès universel aux variables d'un émulateur automate fonctionnant sur le même PC. Le principe est d'identifier l'émulateur par le nom du processus représentant son fonctionnement ainsi qu'une signature permettant de localiser une table d'échange.



Le paramètre "Chaîne de recherche de processus" permet d'identifier un processus. Le premier processus trouvé incluant la chaîne dans le nom du fichier exécutable associé sera utilisé. Attention, cette chaîne est différenciée minuscules/majuscules". Le gestionnaire de tâches de Windows permet de lister les processus en cours d'exécution.

Le paramètre "Signature de la zone d'échange" permet d'identifier le début d'une zone qui sera utilisée pour lire et écrire l'état des variables entre le processus et VIRTUAL UNIVERSE PRO. Ce paramètre est spécifié sous la forme de deux digits hexadécimaux par caractère. Par exemple 52494941 représentera une succession de 4 caractères "RIIA".

Le paramètre "Mode de connexion" peut prendre les valeurs :

- "Simple" : la chaîne est simplement recherchée,
- "Evolué" : dans ce cas un mécanisme plus complexe est mis en oeuvre afin d'éviter les faux "positifs", ceci peut être utile si la chaîne recherchée est dupliquée dans le processus, ce qui peut arriver suivant sa structure interne. Dans ce mode, doit se trouver

immédiatement à la suite de la signature dans la mémoire du processus une succession de trois octets comme suit :

- 1er octet : doit être incrémenté au moins une fois par seconde par le processus,
- 2ème octet : VIRTUAL UNIVERSE PRO écrit cet octet avec différente valeur lors de la phase de connexion. Le processus doit calculer et écrire la valeur du 3ème octet en un temps inférieur à une seconde.
- 3ème octet : doit être écrit par le processus comme égal au carré du 2ème octet : 3ème octet=2ème octet x 2ème octet. Par exemple, une valeur de 3 dans le 2ème octet devra produire l'écriture d'une valeur 9 dans le 3ème octet.

Il convient dans l'émulateur automate de placer en début de la zone d'échange souhaitée les valeurs correspondant à la signature. Le plus simple est généralement de réaliser ceci en ajoutant du code au projet exécuté dans l'émulateur. L'inversion des valeurs poids faible/poids forts peut être nécessaire.

Remarques :

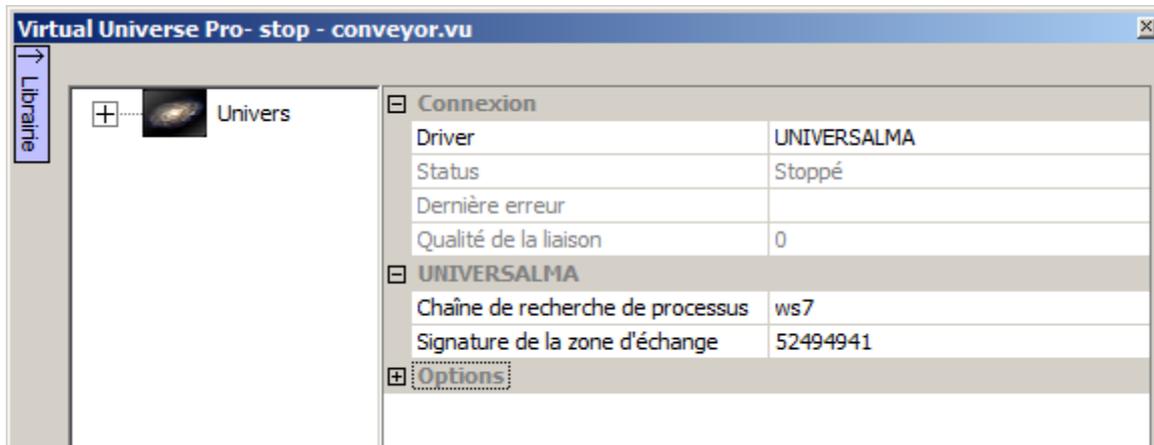
- il est recommandé d'utiliser au moins une signature de 4 caractères pour éviter une localisation erronée de la zone d'échange,
- il convient de réaliser des essais jusqu'à un fonctionnement correct.

Les liens définis dans VIRTUAL UNIVERSE PRO font ensuite référence à des positions par rapport au début de la table d'échange et le type d'accès. La position est arbitrairement précisée en nombre de mot de 16 bits. Le type d'accès précise si on accède à un des bits d'un mot, au mot entier (entier de 16 bits) ou à un groupe de 2, 4 ou 8 mots consécutifs considérés comme des entiers ou des flottants, pour ce dernier cas, c'est le numéro du premier mot du groupe qui est spécifié.

Cette technologie permet de lire et d'écrire les mots de la table d'échange. Si d'autres type de variables spécifique à chaque émulateur automate doivent être écrits ou lus, des instructions de recopie devront être ajoutées dans le programme (voir l'exemple ci-après).

Exemple, utilisation de la connexion universelle avec l'émulateur automate du logiciel WinSPS-S7 de la société MHJ-Software

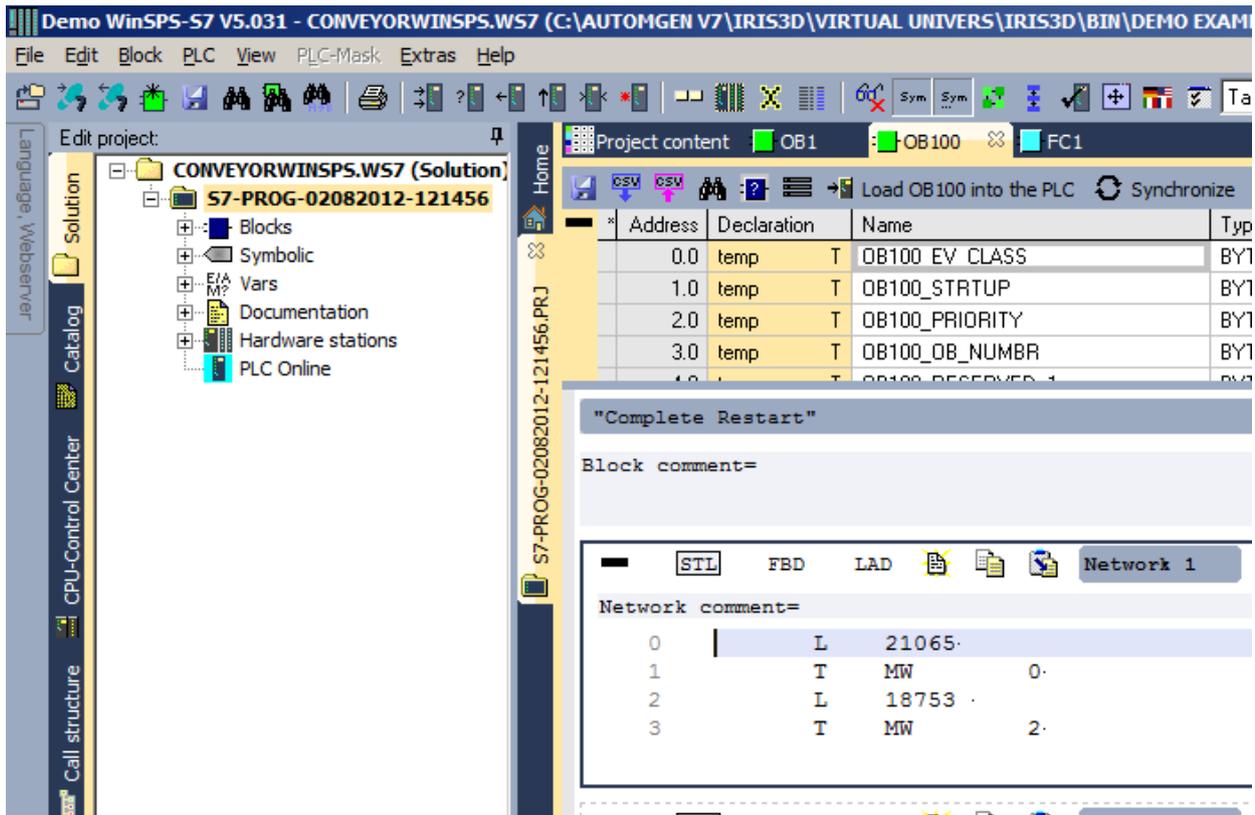
1- paramétrage de la connexion dans VIRTUAL UNIVERSE PRO



Le processus gérant la simulation des programmes automatés dans WinSPS-S7 est l'exécutable principal de l'application nommé par exemple ws7v5.exe pour la version 5. La chaîne de recherche de processus "ws7" permet d'identifier ce processus.

Arbitrairement, la signature choisi est 52494941 soit la suite de 4 caractères RIIA.

2- définition de la signature de la zone d'échange dans l'émulateur automate



La zone d'échange est arbitrairement placée à partir de la variable STEP7 MW0.

Les lignes de programme suivantes :

L 21065

T MW0

L 18753

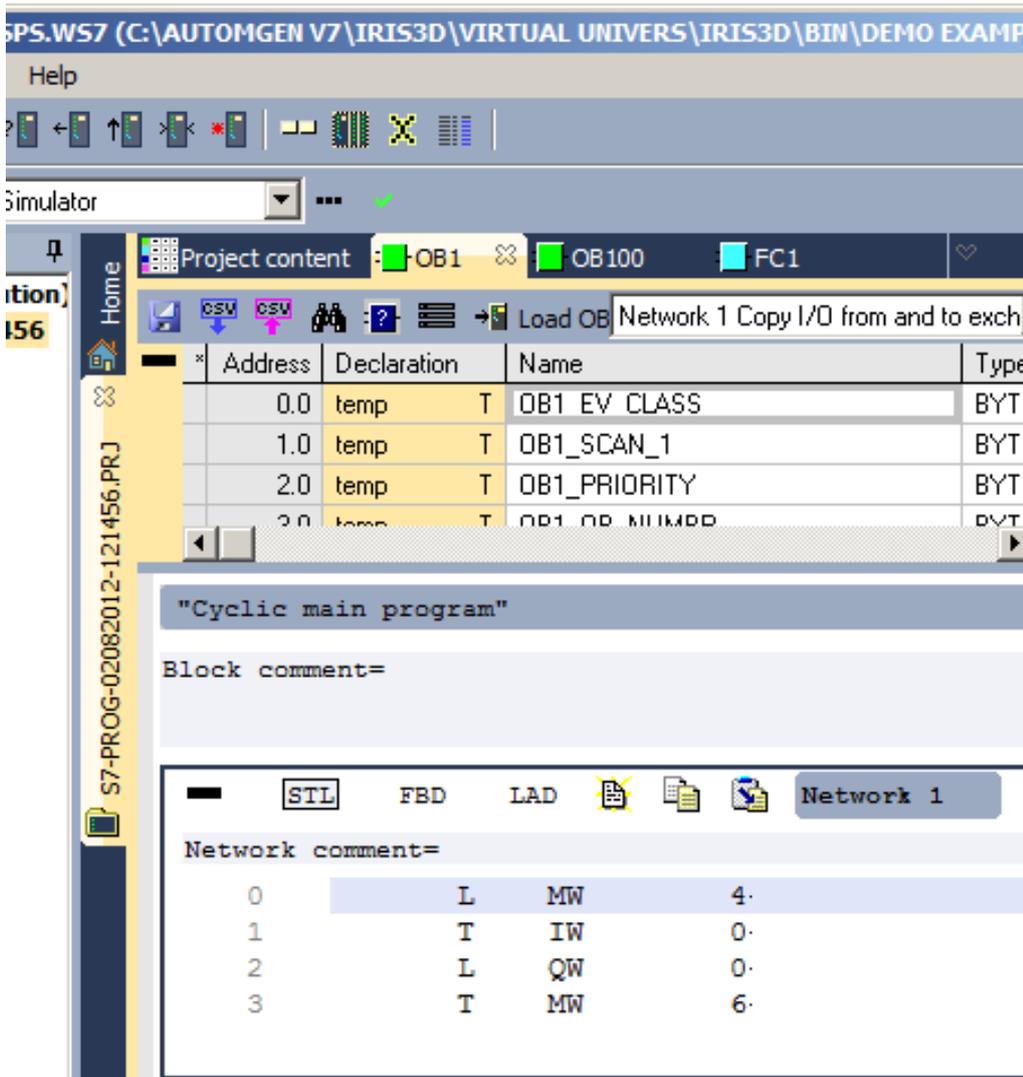
T MW2

placent les 4 caractères RIIA dans les mots MW0, MW2 (dans le cas du langage STEP7; ces deux mots occupent 4 octets consécutifs en mémoire). Cette initialisation est réalisée une seule fois au démarrage du programme automate (utilisation du bloc STEP7 OB100).

3- recopies vers et depuis la zone d'échange

Notre exemple utilise 5 entrées booléennes et 15 sorties booléennes. Arbitrairement, les entrées I0.0 à I0.4 et les sorties Q0.0 à Q0.14 sont utilisées.

Le code de recopie est le suivant :



Par commodité et pour cet exemple, les variables sont copiées par groupe de 16. Le mot MW4 correspond aux entrées du programme automate, le mot MW6 aux sorties.

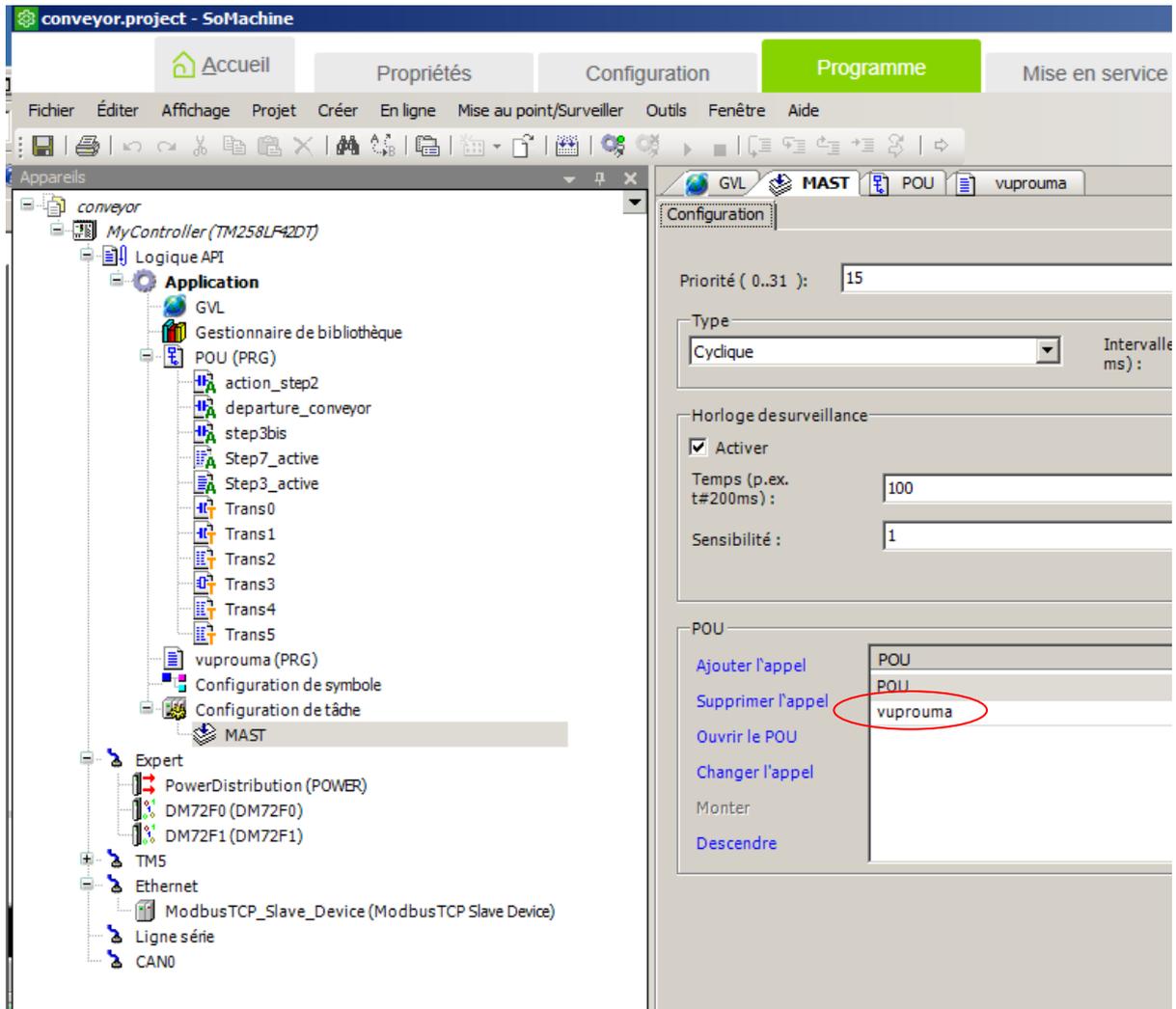
Résumé de l'utilisation de la mémoire de l'émulateur automate :

Octet 0	Octet 1	Octet 2	Octet 3	Octet 4	Octet 5	Octet 6	Octet 7
"R" 52	"I" 49	"I" 49	"A" 41				
Mot STEP7 MW0		Mot STEP7 MW2		Mot STEP7 MW4		Mot STEP7 MW6	
Mot table d'échange position 0		Mot table d'échange position 1		Mot table d'échange position 2		Mot table d'échange position 3	
				Entrées STEP7 IW0		Sorties STEP7 QW0	

3- définition des liens dans le projet VIRTUAL UNIVERSE PRO

Mise au point		
Nom	Alias	Variable externe
Conveyor 1 Rotates rollers sens 1		WORD3.0
Conveyor 1 Rotates rollers sens 2		WORD3.1
Conveyor 1 Rollers brake		WORD3.2
Bottom sensor		WORD2.0
switch on/off red light		WORD3.3
switch on/off orange light		WORD3.4
switch on/off green light		WORD3.5
Top sensor		WORD2.1
Conveyor Switch Rotates rollers sens 1		WORD3.6
Conveyor Switch Rotates rollers sens 2		WORD3.7
Conveyor Switch Rollers brake		WORD3.8
Backward sensor		WORD2.2
Forward sensor		WORD2.3
Conveyor 2 Rotates rollers sens 1		WORD3.9
Conveyor 2 Rotates rollers sens 2		WORD3.10
Conveyor 2 Rollers brake		WORD3.11
Conveyor 2 sensor		WORD2.4
switch on/off red light		WORD3.13
switch on/off orange light		WORD3.12
switch on/off green light		WORD3.14

Ajout de l'appel cyclique de ce programme :



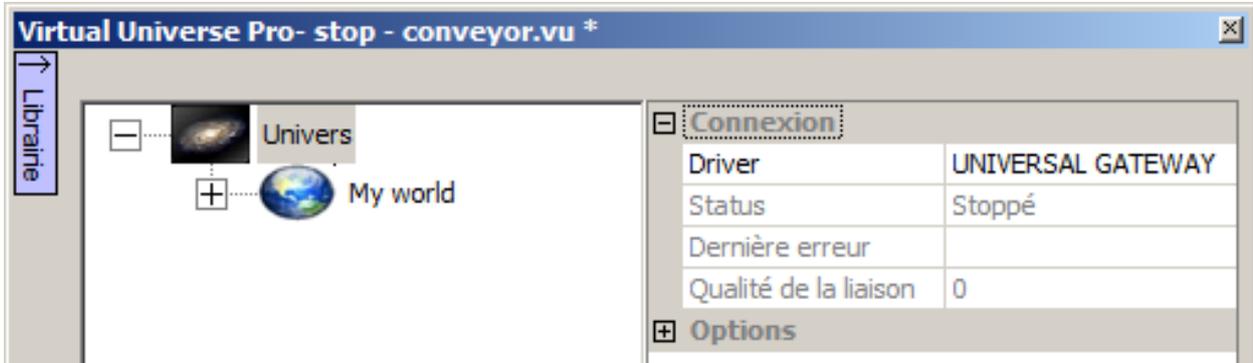
Configuration dans VIRTUAL UNIVERSE PRO



Connexion à Universal Gateway

Universal Gateway est une passerelle universelle permettant de dialoguer simultanément avec plusieurs automates de mêmes types ou de types différents.

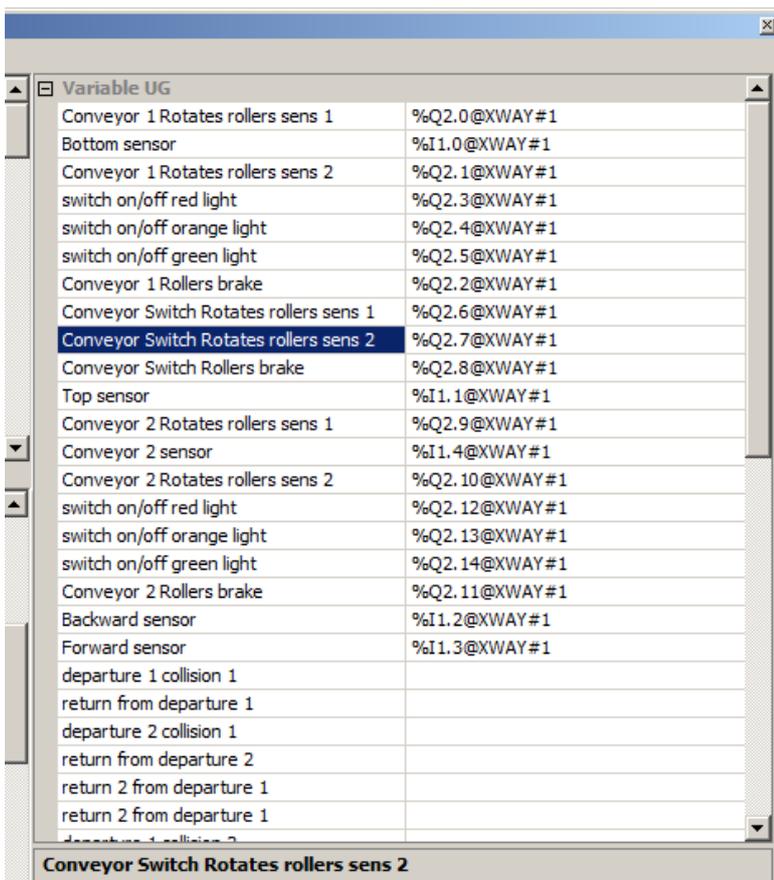
Sélectionnez "Universal Gateway" dans les propriétés de VIRTUAL UNIVERSE PRO:



Syntaxe des liens

La syntaxe générale des liens est <nom de variable>@<nom de passerelle>

Exemple de liens :



Reportez-vous au manuel de Universal Gateway pour plus d'informations.

Accès direct aux variables externes

Dans les contrôleurs virtuels, les blocs entrées et sorties du langage Fbd / Sfc peuvent utiliser une syntaxe spéciale pour accéder directement aux variables externes.

La syntaxe est : `*\<nom de la variable externe >`

Le nom de la variable externe dépend du driver.

Exemple avec un automate Siemens S7 :

