

---

# JAM asset viewer PC

## User Manual

Yannick

---

### Table des matières

Asset manager.....	2
Introduction.....	2
Contexte.....	2
Contenu.....	2
Installation.....	3
Utilisation.....	3
Format et structure de fichier SUA.....	4
Format de fichier PAK.....	5
Génération de fichier PAK.....	6
Conditions d'utilisation.....	6
Limitations.....	6
Support technique.....	7
What's new?.....	7

## Asset manager

### Introduction

Ce document présente « JAM asset viewer » qui permet de visualiser par simple drag&drop le contenu d'un fichier SUA.

### Contexte

SUGAR est un framework de développement de jeu vidéo et d'applications multimédia sur PC, consoles next-gen et appareils mobiles.

JAM est l'interface de tous les outils du framework SUGAR.

L'asset viewer est un paramétrage spécifique de JAM.

Il enchaîne automatiquement toutes les étapes du flux de production qui mène du fichier exporté SUA à l'affichage dans une vue 3D de l'ensemble des éléments contenus dans ce fichier:

- Ouverture d'une session anonyme de JAM contenant un visualiseur d'asset 3D
- Export de l'ensemble des géométries du fichier SUA dans un fichier PAK
- Affichage du contenu du fichier PAK dans le visualiseur

### Contenu

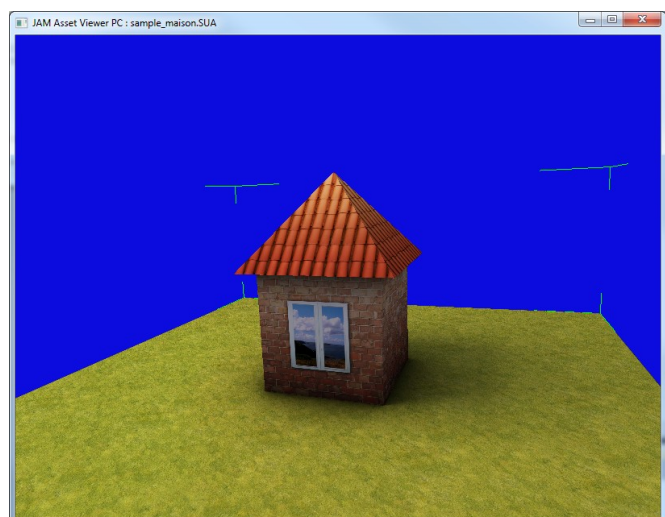
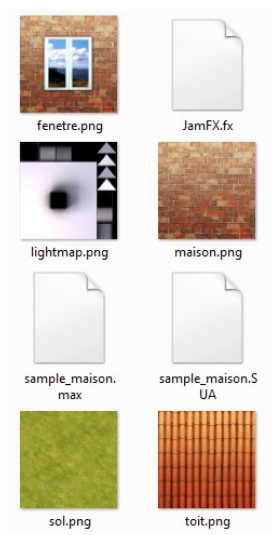
Le viewer est livré en 3 packages séparés: JAM, Assemblies et Samples.

Le package « **JAM** » contient la dernière version en date de JAM.

A décompresser sur n'importe quel disque local.

Le package « **Assemblies** » doit être décompressé au même endroit que JAM. Il ne contient que des dll third party indispensables au fonctionnement de JAM.

Le package « **Samples** » contient des exemples qui regroupent les fichiers sources et les fichiers exportés.



### Installation

Décompresser le package « JAM » sur sa machine.

Décompresser dans le même repertoire le package « Assemblies ».

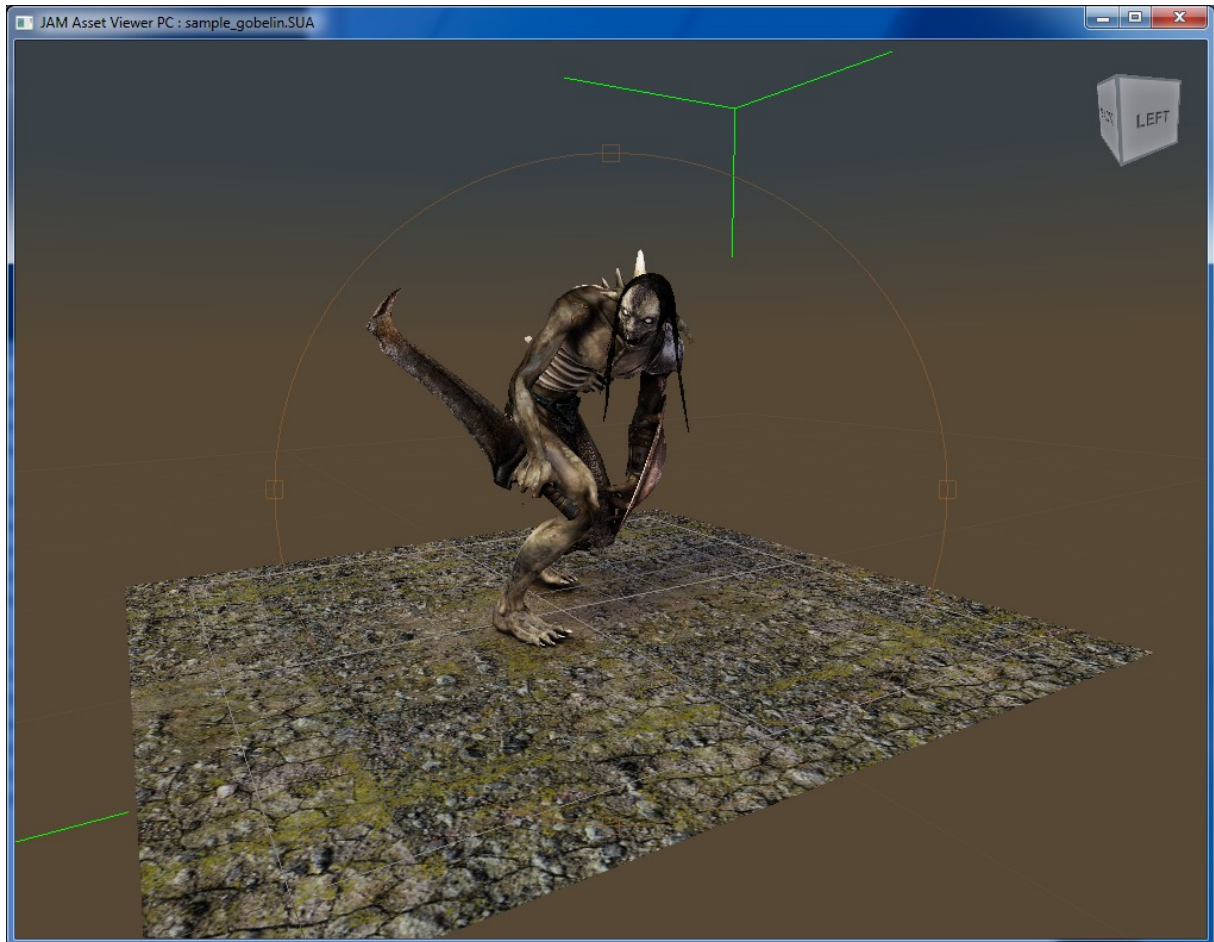
Exécuter jam.wsf pour lancer l'application.

## Utilisation

Pour visualiser un fichier de type SUA dans le viewer, il faut le glisser-déposer (drag & drop ) dans la fenêtre de visualisation.

Le viewer génère alors un fichier pak dans le même répertoire que le SUA à partir du sua puis charge le pak pour l'afficher.

La fenêtre de visualisation accepte aussi les fichiers de type PAK, par glisser-déposer (drag & drop ) qu'elle affiche directement.



Les fonctions disponibles sont:

- tourner autour de la scène en maintenant le bouton enfoncé tout en déplaçant la souris.
- Un « zoom » avec la molette.

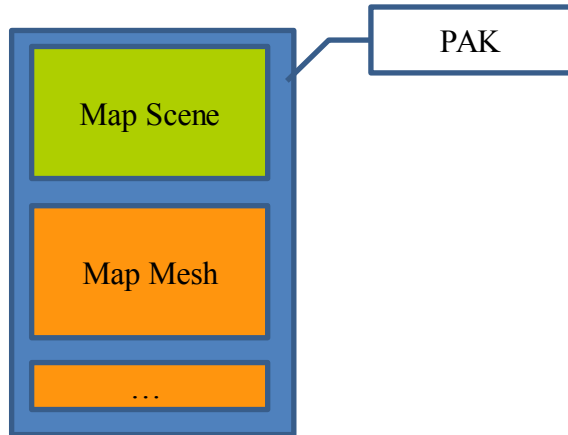
## Format et structure de fichier SUA

Pour une description détaillée du format SUA, se référer à la documentation du plugin d'export MAX.

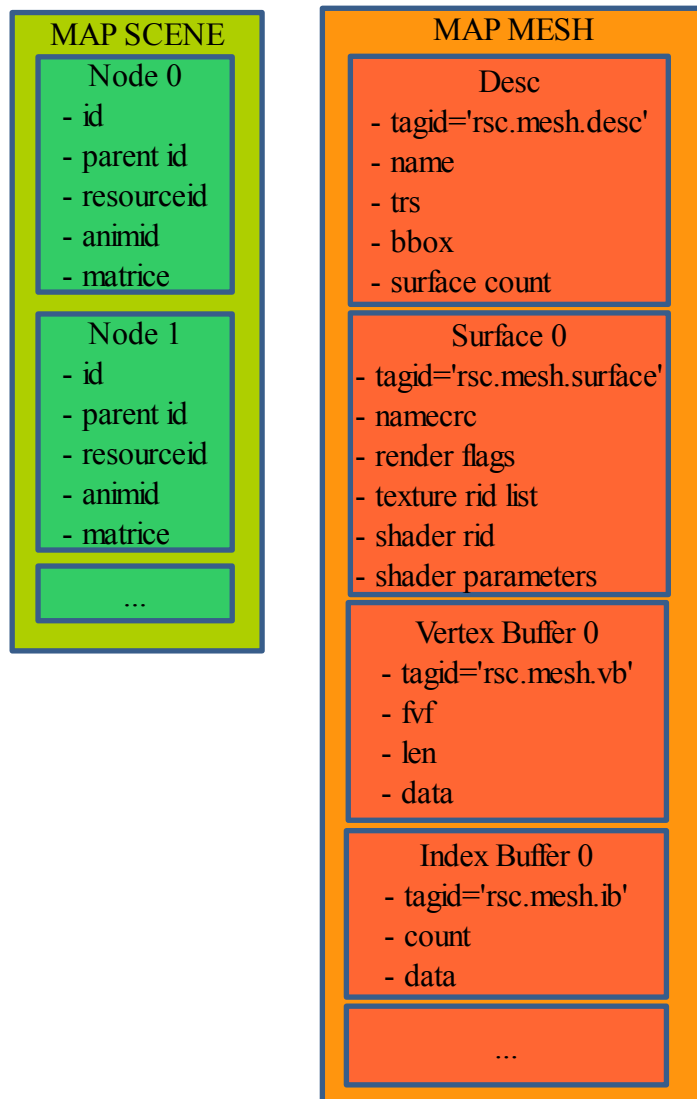
## Format de fichier PAK

Le fichier PAK est composé d'un ensemble de ressources binaires qui possèdent un id unique (rid).

Chaque ressource binaire est définie en tant que map qui peut être de plusieurs types: mesh, texture...



Une map est composée et définie entièrement par un ensemble de tags.



Une map mesh contient un tag descripteur unique. Il nous renseigne sur le nombre de surfaces qu'il contient.

La surface est la plus petite unité affichable par le viewer, elle nous renseigne sur la matière, les textures, et les vertex et index buffers d'une sous partie du mesh.

Pour chacune des surfaces d'un mesh le viewer s'attend à trouver un tag de type surface, un tag de type vb et un tag de type ib à la suite.

Une map de type scene contient exclusivement des tags de type node.

## Génération de fichier PAK

Il est possible de créer des maps et de les ajouter dans un pak entièrement en lua.

Chaque tag binaire est représenté par une table en lua.

Une map binaire est représentée par une table en lua qui contient l'ensemble des tables de ses tags.

La transformation d'un fichier SUA en ressource d'un fichier PAK se fait à l'aide de scripts en LUA.

Le point d'entrée est dans node\_asset\_man.lua dans la fonction `GeneratePackFromSuaFile()`.

Une map de type scène est générée et insérée dans le PAK. Sa génération nous rend la liste des géométries à générer.

On parcourt cette liste pour générer des maps de type mesh à insérer dans le PAK.

La génération des meshes nous rend une liste l'ensemble des fichiers de références (textures, shader, ...) à binariser et à ajouter au PAK.

Pour binariser une table lua qui correspond à une map et mettre à jour un pak avec il faut suivre ces différentes étapes :

`bm = su.util.binarizeMap( myMesh )` renvoie un objet qui contient la map binaire.

`pak = su.util.openPak( fPath )` crée un objet pak via lua

`pak:startUpdating()` Prépare le pak pour l'update

`pak:updateContent( bm )` update le pak avec la map binaire

## Conditions d'utilisation

La version actuelle permet d'afficher des PAK contenant des scènes ayant un nombre quelconque de géométries, elles-même constituées d'un nombre quelconque de surfaces. Les multi-matériaux sont pris en compte.

## Limitations

On peut utiliser 2 types de matériaux:

- Les matériaux standards avec une map de diffuse
- Les matériaux de type FX en utilisant le shader « JamFX.fx »

Ce FX gère une DiffuseMap et une LightMap et trois techniques différentes :

- DiffuseMap
- DiffuseMap + LightMap utilisant les même UV
- DiffuseMap + LightMap avec des UV différents

JAM cherche les fichiers sources dans le répertoire d'origine, et en cas d'échec dans le même répertoire que le SUA.

Deux objets ou plus ne doivent pas porter le même nom dans le modeleur.

Un mesh exporté sans UV n'est pas chargé dans JAM.

Il n'y a pas de tri en alpha sur les objets.

Il serait préférable d'appliquer un ResetXForm avant de faire les exports.

Le nombre de triangles supportés pour le moment est de 21845 par surface au maximum (mais pas de limite du nombre de surface par mesh).

## Support technique

Pour tout rapport d'anomalie, demande d'information technique, veuillez créer un ticket dans la base de bug à l'adresse suivante:  
(en attente d'installation)

En attendant, pour une utilisation au sein de Axyz, merci de passer par Bruno Plantier qui fera suivre à atOnce.

## What's new?

23/03/2011:

- Indication de la restriction: 21845 triangles par surface.
- Correction du bug de génération de nodes et mesh (artefact de rotation)
- Correction du bug de render state dans JamFx.fx (affichage des UV de la diffuse en « repeat » dans 3DS Max)
- Ajout du cube cliquable dans le viewer.
- Ajout du dessin de l'arcball.
- Ajout du Drag&Drop du .pak sur le viewer.
- Génération du PAK dans le même répertoire que le SUA.

16/03/2011:

- la restriction des sources et des exécutables sur les drives C: ou D: n'existe plus.
- La livraison se fait désormais en 3 packages: Jam, Assemblies et Samples
- Ajout d'une grille de visualisation
- Correction d'un bug de manipulation de l'objet à la souris
- Correction d'un bug d'affichage pour des géométries ayant plus de 2 niveaux hiérarchiques.