

Paramétrer le moteur de rendu SunFlow

avec Sweet Home 3D version 4.0 et supérieur

Introduction

Sweet Home 3D utilise le moteur [SunFlow](#) pour le rendu aux deux meilleurs niveaux de qualité.

Depuis la version 4.0 de Sweet Home 3D, il est possible de paramétrer ce moteur de rendu à l'aide de propriétés lues depuis un fichier *.properties* ou depuis des propriétés système.

Le fichier ressource [PhotoRenderer.properties](#) (révision 1.2) contient les valeurs par défaut des paramètres de rendu :

1. Paramètres initiaux

- [Qualité 3](#)

```
lowQuality.antiAliasing.min=0
lowQuality.antiAliasing.max=0
lowQuality.globalIllumination=default
lowQuality.diffusedBounces=1
lowQuality.causticsPhotons=0
lowQuality.shininessShader=default
lowQuality.normalLens.focusDistance=250.
lowQuality.normalLens.radius=1
```

- [Qualité 4](#)

```
highQuality.antiAliasing.min=1
highQuality.antiAliasing.max=2
highQuality.globalIllumination=default
highQuality.diffusedBounces=1
highQuality.causticsPhotons=0
highQuality.shininessShader=default
highQuality.normalLens.focusDistance=250.
highQuality.normalLens.radius=1
```

2. Paramètres additionnels

- [Qualité 3](#)

```
lowQuality.filter=box
lowQuality.samplerAlgorithm=bucket
```

- [Qualité 4](#)

```
highQuality.filter=blackman-harris
highQuality.samplerAlgorithm=bucket
```

Paramétrage à l'aide de lignes de commande

Pour la [version Jar exécutable](#) de Sweet Home 3D, il faut prendre les paramètres du fichier *PhotoRenderer.properties* en rajoutant devant ceux-ci
-D*com.eteeks.sweethome3d.j3d.PhotoRenderer* pour définir des propriétés système.

Ce qui donne comme exemple de propriété modifiable :

-D*com.eteeks.sweethome3d.j3d.PhotoRenderer.levelQuality.property=value*

Dans un terminal ou une console, rajouter entre java et -jar */path/to/SweetHome3D-version.jar* les lignes des propriétés à modifier commençant par -D et séparées par un espace.

Paramétrage à l'aide de fichier

Pour la [version installateur](#) de Sweet Home 3D, il faut prendre les paramètres du fichier *PhotoRenderer.properties* en rajoutant devant ceux-ci
-D*com.eteeks.sweethome3d.j3d.PhotoRenderer* pour définir des propriétés système.

Ce qui donne comme exemple de propriété modifiable :

-D*com.eteeks.sweethome3d.j3d.PhotoRenderer.levelQuality.property=value*

Vous pouvez prendre les lignes des propriétés que vous souhaitez modifier et les mettre dans un fichier :

- Sous Windows, nommer ce fichier texte *SweetHome3D.l4j.ini* dans le même dossier que *SweetHome3D.exe* .
- Sous Linux, éditez le script de lancement SweetHome3D en ajoutant après *exec "\$PROGRAM_DIR"/jre1.6.0_43/bin/java* les lignes commençant par -D et les séparer par un espace.

Paramétrage à l'aide de plug-in

Le plug-in [Paramètres avancées de rendu photo](#) permet de modifier les valeurs des propriétés directement dans Sweet Home 3D.

Pour l'installer, il suffit de double-cliquer dessus (à partir de Java 1.6) sauf pour la [version portable](#) de Sweet Home 3D où il faut le placer dans le dossier *data/plugins*.

La [version Java Web Start](#) de Sweet Home 3D **n'autorise pas** l'utilisation de celui-ci.

A l'ouverture du plug-in, il affiche :

- soit les valeurs par défaut des propriétés lues depuis la ressource *PhotoRenderer.properties* (bouton *valeurs par défaut* grisé)
- soit les valeurs des propriétés système lues depuis un fichier *SweetHome3D.l4j.ini* ou un script de lancement Linux (bouton *valeurs par défaut* activé).

En l'occurrence, les valeurs modifiées ne sont valables que pendant l'utilisation de Sweet Home 3D.

1. Paramètres initiaux

a) antiAliasing.min et antiAliasing.max

Antialiasing minimum et maximum utilisés pour calculer l'image ($aAmin \leq aAmax$).

Des valeurs élevées peuvent augmenter le temps de rendu.

b) globalIllumination

Algorithme d'illumination globale : "default" ou "path".

"default" utilise l'occlusion ambiante en mode visite virtuelle.

"path" prend beaucoup plus de temps à calculer mais donne une vue plus réaliste. Il faut utiliser un ciel sans texture en visite virtuelle sinon le rendu sera sombre. Un éclairage puissant peut donner un rendu éblouissant.



GI default, dB 1



GI path, dB 2

c) diffusedBounces

Rebonds maximum effectuées par les rayons lumineux quand GI "path" est utilisé.

L'augmentation de cette valeur ralentit grandement le processus de rendu.

Pour une valeur de 0, le rendu est contrasté c'est-à-dire des ombres plus prononcées quel que soit le type d'illumination globale. Il faut donc absolument un éclairage.



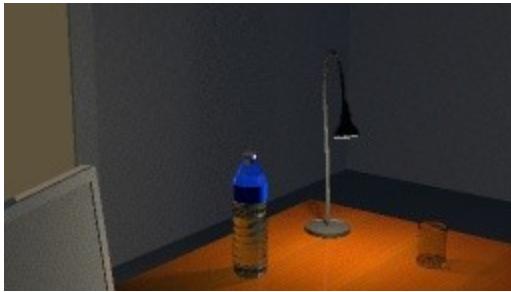
GI default ou path, dB 0

d) causticsPhotons

Nombre de photons caustiques, avec 0 ne produit pas de caustiques.

Des valeurs élevées peuvent augmenter le temps de pré-calcul de l'image.

Avec une valeur de 500000, on peut voir certains effets si l'icône lune est active.



cP 500000, icône soleil



cP 500000, icône lune

e) shininessShader

Shader utilisé pour rendre les matériaux brillants : "default", "glossy" ou "silk".

"default" utilise le shader *silk* (*soyeux*) au niveau de qualité élevé et en mode de vue virtuelle, *glossy* (*brillant*) dans le cas contraire.



silk



glossy

f) normalLens.focusDistance

Distance de mise au point en cm lorsque la profondeur de champ est utilisée.

g) normalLens.radius

Rayon de l'objectif lorsque la profondeur de champ est utilisée.

Une valeur plus grande produit une profondeur de champ plus étroite.



focusDistance 250, radius 1



focusDistance 250, radius 3

2. Paramètres additionnels

a) filter

Filtre utilisé pour contrôler l'image sur-échantillonnée : "box", "triangle", "gaussian", "mitchell", "catmull-rom", "blackman-harris", "sinc", "lanczos" ou "bspline".

b) samplerAlgorithm

Algorithme de l'échantillonneur utilisé par le moteur de rendu : "bucket", "fast" ou "ipr".