Configure the SunFlow rendering engine

with Sweet Home 3D version 4.0 and higher

Introduction

Sweet Home 3D uses the SunFlow engine for rendering the two best levels of quality.

Since version 4.0 of Sweet Home 3D, it is possible to set this rendering engine using properties read from a *.properties* file or from system properties.

The PhotoRenderer.properties resource file (revision 1.2) contains the default values of render settings :

1. **Initial parameters**
   - Quality 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
   - Quality 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. **Additional parameters**
   - Quality 3
     lowQuality.filter=box
     lowQuality.samplerAlgorithm=bucket
   - Quality 4
     highQuality.filter=blackman-harris
     highQuality.samplerAlgorithm=bucket

Setting using command lines

For the Jar executable version of Sweet Home 3D, take settings of *PhotoRenderer.properties* file adding before them *-Dcom.eteks.sweethome3d.j3d.PhotoRenderer* to set system properties.

Giving as an example of editable property :
-Dcom.eteks.sweethome3d.j3d.PhotoRenderer.***level***Quality.*propertie*=*value*

In a terminal or console, add between java and -jar */path/to/*SweetHome3D-*version*.jar the properties lines to change starting with -D and separated by a space.

Setting using file

For the installer version of Sweet Home 3D, take settings of *PhotoRenderer.properties* file adding before them *-Dcom.eteks.sweethome3d.j3d.PhotoRenderer* to set system properties.

Giving as an example of editable property :
-Dcom.eteks.sweethome3d.j3d.PhotoRenderer.***level***Quality.*propertie*=*value*

You can take the properties lines you want to change and put them in a file :

- On Windows, name the text file *SweetHome3D.l4j.ini* in the same folder as *SweetHome3D.exe* .
- On Linux, edit the SweetHome3D launch script adding after *exec "$PROGRAM_DIR"/jre1.6.0_43/bin/java* the lines beginning with -D and separate them by a space.

Setting using plug-in

The plug-in Advanced settings of photo rendering allows to change the property values directly in Sweet Home 3D.

To install it, just double-click on it (from Java 1.6) except for the portable version of Sweet Home 3D where to place it in the folder *data/plugins*.
The Java Web Start version of Sweet Home 3D **does not allow** the use thereof.

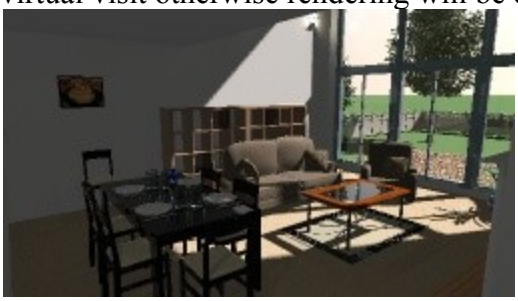
On opening the plug-in, it displays :

- either the default values of properties read from the *PhotoRenderer.properties* resource (button *Default values* grayed)
- or the values of system properties read from a *SweetHome3D.l4j.ini* file or a Linux launch script (button *Default values* enabled).

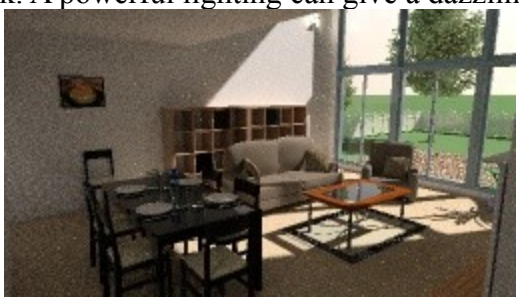
In this case, the modified values are valid only during the use of Sweet Home 3D.

1. **Initial parameters**
   a) antiAliasing.min and antiAliasing.max
      Minimum and maximum antialiasing used to compute image (aAmin ≤ aAmax).
      High values can increase rendering time.

   b) globalIllumination
      Global illumination algorithm : "default" or "path".
      "default" uses ambient occlusion in virtual visit mode.
      "path" takes much longer to compute but gives more realistic view. Use a sky without texture in virtual visit otherwise rendering will be dark. A powerful lighting can give a dazzling rendering.

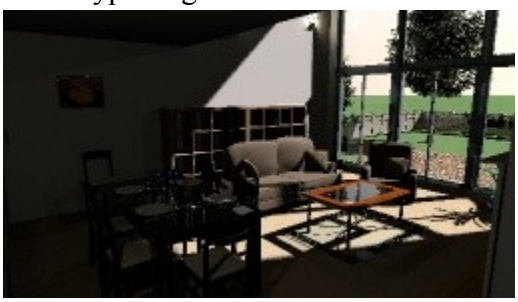


GI default, dB 1



GI path, dB 2

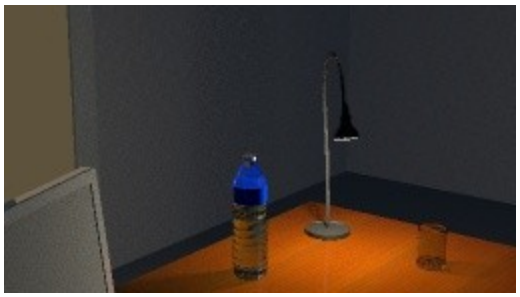
   c) diffusedBounces
      Maximum bounces done by light rays when global illumination "path" is used.
      Increasing this value greatly slows down rendering process.
      For a value of 0, the rendering is contrasted that is to say more pronounced shadows regardless of the type of global illumination. It is therefore absolutely lighting.



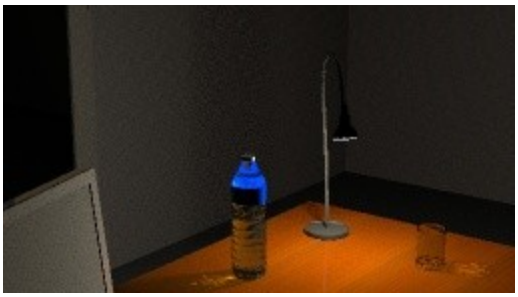
GI default or path, dB 0

d) causticsPhotons

Caustics photons count, with 0 producing no caustics.
High values can increase the time of image pre-computation.
With a value of 500000, we can see some effects if the moon icon is active.



c*P 500000, sun icon*



cP 500000, moon icon

e) shininessShader

Shader used to render shiny materials : "default", "glossy" or "silk".
"default" uses *silk* shader at high quality level and in virtual view mode, *glossy* otherwise.

silk

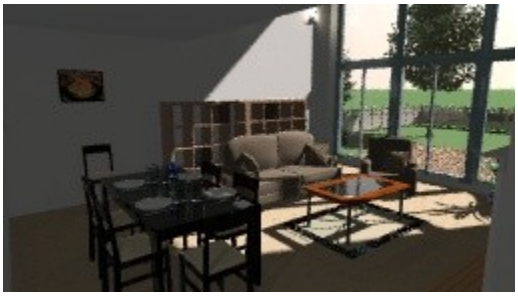
glossy

f) normalLens.focusDistance

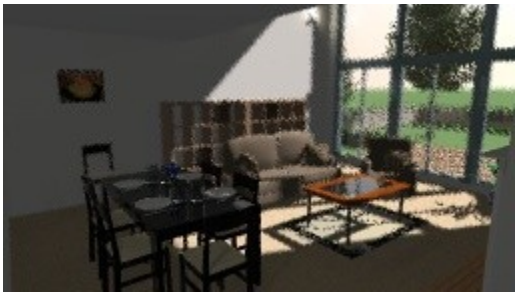
Focus distance in cm when depth of field is used.

g) normalLens.radius

Lens radius when depth of field is used.
A bigger value produces a narrower depth of field.



focusDistance 250, radius 1



focusDistance 250, radius 3

2. Additional parameters

a) filter

Filter used to control oversampled image : "box", "triangle", "gaussian", "mitchell", "catmull-rom", "blackman-harris", "sinc", "lanczos" or "bspline".

b) samplerAlgorithm

Sampler algorithm used by the renderer : "bucket", "fast" or "ipr".