

OPEN CLONK

BACK TO THE ROCKS

Artist's Guide

By Charles "Ringwaul" Spurrill

Getting Started

To create 3D models used by OpenClonk, you'll need to have a few things in advance. This guide will assume the reader has a basic understanding of how to create 3D models.



Blender – The modelling tool which will be used to create 3D models. Other 3D graphics applications may have the ability to export into the OGRE format used by OpenClonk, but only Blender will be covered in this guide.

Download:

<http://www.blender.org/download/get-blender/>

OGRE Exporter – A plugin for Blender which allows 3D models to be exported in a format usable by OpenClonk. Follow the installation instructions at the download page.



Download:

<http://www.ogre3d.org/tikiwiki/tiki-index.php?page=Blend>

OGRE XML Converter – Optional converter from .xml OGRE format to .mesh format. Meshes will load much faster than .xmls in game. (recommended)

Download:

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

Guidelines for Modelling

There are some guidelines which must be followed for 3D models to operate properly inside the OpenClonk engine. Not following these guidelines can result in 3D models not loading, errors in display of the model, or other unwanted effects.

#1 – Model on the Y/Z Plane

Models in OpenClonk appear as they do on the Y/Z plane in blender. The X-Axis goes 'into' the screen. Make sure you create models with regard to these axes.

#2 – Do not use Mesh Transformations

Mesh transformations are not handled properly by the OGRE exporter, and will cause deformation in game. If you have transformations applied to an existing model, you can remove them (select the model and press alt+g, alt+s, alt+r).

Caution: Avoid animating before removing transformations!

#3 – 3D Model must be Three-Dimensional

The 3D mesh must have length on all axes. Making a perfectly flat object on only two axes will cause it to not appear in-game. To fix this, move a vertex so the mesh has length on all axes.

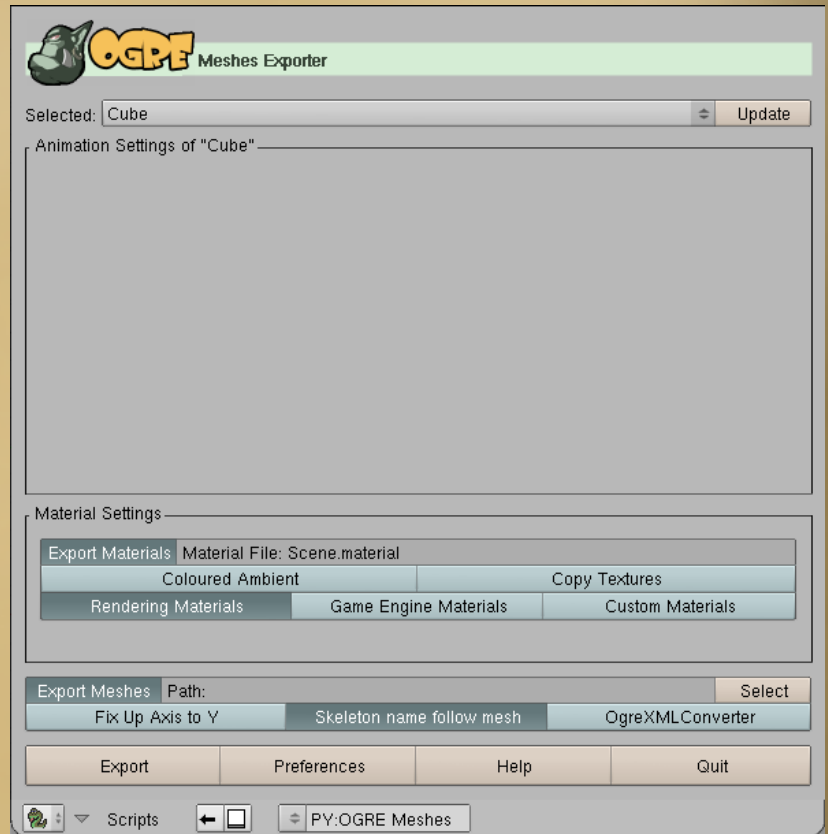
#4 – What features work with the OGRE format?

Not all of the features listed in Blender will affect an exported 3D OGRE mesh. For a full list of compatible features, visit:

<http://www.cs.ucr.edu/~macchiea/cs134/lab1/ogrehelp/ogremeshesexporter.html>

Exporting the Model

To export the model, you'll need to have a window for the OGRE exporter in Blender. Make a new pane for scripts, and click 'File/Export/OGRE Meshes'.



-The Ogre Exporter-

Models used for OpenClonk must be exported with specific options.

-“Fix Up Axis to Y” should not be enabled.

-If you have the XML converter, make sure “OgreXMLConverter” is enabled.

If you have already made manual modifications to the Scene.material file and are re-exporting the mesh and/or skeleton, make sure 'Export Materials' is not enabled.

'Path:' will be where the OGRE files are exported to. You can use a temporary directory and then copy them into your OpenClonk object definition, or you can export the OGRE files directly into the object definition.

Using the Exported OGRE Files

Once you have exported the mesh, some files will be created in the chosen directory.

Follow these steps:








-Rename the “.mesh” file to “Graphics.mesh”.

-If you did not use the XML Converter, rename the “.mesh.xml” file to ‘Graphics.mesh.xml’

(The “.mesh” and “.mesh.xml” files will be named after the object exported from blender. i.e. ‘Cube.mesh’)

-If you exported the OGRE files to a temporary directory, copy ‘Graphics.mesh’, the ‘.skeleton’ file, the ‘Scene.material’ file, and any textures used by the mesh to your object definition.

-If you have exported the OGRE files directly to your object definition, you can delete all .xml files and the .log file.

 Cube.mesh	31/08/2010 12:01 ...	MESH File	3 KB
 Cube.mesh.xml	31/08/2010 12:01 ...	XML Document	8 KB
 Cube.skeleton	31/08/2010 12:01 ...	SKELETON File	1 KB
 Cube.skeleton.xml	31/08/2010 12:01 ...	XML Document	1 KB
 MyCube.png	31/08/2010 12:01 ...	PNG image	67 KB
 OgreXMLConverter.log	31/08/2010 12:01 ...	Text Document	3 KB
 Scene.material	31/08/2010 12:01 ...	MATERIAL File	1 KB

-Exported Files-

Scene.material

Scene.material is the file which defines which materials to apply to meshes. Manually modifying Scene.material is crucial to the mesh appearing properly in game.

When you export (assuming you are using a texture) your Scene.material file will look more or less like this:

```
material MyCube
{
    receive_shadows on
    technique
    {
        pass
        {
            cull_hardware none
            ambient 0.500000 0.500000 0.500000 1.000000
            diffuse 0.640000 0.640000 0.640000 1.000000
            specular 0.500000 0.500000 0.500000 1.000000 12.500000
            emissive 0.000000 0.000000 0.000000 1.000000
            scene_blend alpha_blend
            texture_unit
            {
                texture MyCube.png
                tex_address_mode wrap
                filtering trilinear
            }
        }
    }
}
```

-If 'depth_write off' exists, delete that line.

-If you are using alpha, make sure 'scene_blend alpha_blend' exists (as shown above).

-If your mesh has visible backface(s), make sure to add 'cull_hardware none' (as shown above).

-Depending on the method you used to define alpha in blender, the 'pass' section in Scene.Material may look like this:

```
pass
{
    ambient 0.500000 0.500000 0.500000 0.000000
    diffuse 0.640000 0.640000 0.640000 0.000000
    specular 0.500000 0.500000 0.500000 0.000000 12.500000
    emissive 0.000000 0.000000 0.000000 0.000000
    ...
}
```

If so, make sure the fourth parameter on ambient, diffuse, specular, and emissive is '1.00000', not '0.000000'. (as shown in first box)