



S O C L A B

SOCIETIES OF COMPUTATION LABORATORY

OPEN RENDER BUMP

USER MANUAL

OPEN RENDER BUMP

USER MANUAL

QUICK START

1. **Installation.** To install Open Render Bump (ORB) on your computer, unzip `orb.zip` into a directory of your choice (remember to extract the file *with* sub-directories).
2. **Resolution.** Set your desktop to 32bit color.
3. **Console.** ORB is operated by means of a *console*. You can toggle between the console and the 3D view by pressing the '~' key. Type 'help' to get a listing of available commands.¹
4. **Examples.** If you just want to test ORB without your own models, there are a few example models and related configuration files² included in the ORB distribution. Simply bring up the console and type one of the following commands:

```
exec head.cfg
```

```
exec head_hw.cfg
```

```
exec cyl.cfg
```

```
exec tech_box.cfg
```

[...study these to see how ORB can be used!](#)

5. **Navigation.** In order to navigate in the 3D view of the imported models, press '~' and use the following keys:

W, S, A, D	forward, backward, left, right
------------	--------------------------------

Q, E	down, up
------	----------

MOUSE	look
-------	------

MOUSEWHEEL	Scale view
------------	------------

SHIFT	Slow/Fast
-------	-----------

1, 2	Toggle light source FRONT/BACK, ABOVE/BELOW
------	---

6. **Exit.** ORB is exited by typing 'quit' in console.

¹ If the keyboard input doesn't work properly, or appears strange, switch to ENG (US) input locale in Windows 2000/XP.

² The configuration (.cfg) files are executed as command-based batch files.

REQUIREMENTS

- Windows 2000/XP (ORB may also work on Windows 9x, but this has not been tested yet).
- OpenGL compliant graphics card with 32 bit color, 8 bit stencil.
- ENG (US) keyboard input locale (may work with others)

CONSOLE

The Open render bump (ORB) tool is operated by means of a *console*. You can activate it at any time by pressing the '~' key.³ The available operations of ORB are accessible in run time by typing 'help' in the console.

CONSOLE NAVIGATION

UP arrow	- Command history up
DOWN arrow	- Command history down
PAGE UP	- Scroll console output up
PAGE DOWN	- Scroll console output down
INSERT	- Toggle insert/replace char
LEFT arrow	- Back in cmd string
RIGHT arrow	- Forward in cmd string
HOME	- Back to start in cmd string
END	- End of cmd string
TAB (in first cmd token)	- TAB completion among console commands
TAB (in subsequent tokens)	- TAB completion among files in (imports and exports
directories and *.cfg files)	
ENTER	- Execute command
ESCAPE	- Clear cmd string

MISC COMMANDS

Help	- (This listing)
exec <filename>	- Execute bat file
Quit	- Quit program
cg_CardInfo	- OpenGL extension and other gfx card features

IMPORT COMMANDS

ImportModel high.ext low.ext	- Import Model high.ext = high poly model low.ext = low poly model ext = ASE, OBJ, or LWO
FlipYZWhen3dsImport on/off	- Flip 3DS coords to "normal" (default = off)
NegZWhenLWOImport on/off	- Flip Lightwave coords to "normal" (default = on)

³ If the keyboard input doesn't work properly, or appears strange, switch to ENG (US) input locale in Windows 2000/XP.

RENDER BUMP COMMANDS

RenderBump dir/name	- Raycast version
RenderBumpMulti dir/name	- Raycast version (1 map per material)
RenderBumpHW dir/name	- Hardware version
RenderBumpHWMulti dir/name	- Hardware version (1 map per material)

SETTINGS

Scale f	- Scale model in view (no scaling is 1.0, half size is 0.5)
MultiSample on/off	- 5 samples per pixel (better results, but 5x slower, default=off)
TangentSpace on/off	- Use tangent space (default = on)
MultiPassErrorCorrection on/off	- Toggle error correction passes on/off
ExpandFilter on/off	- Expands pixels to empty areas after Renderbump (default=on, recommended!)
VerticalFlip on/off	- Vertical flip maps when saving to disk (default=on)
FlipYInNormalMap on/off	- Flip Y when writing normalmap to disk, I.e. Normal(x,y,z) = Norma(x,-y,z) (Doom3, nVidia Photoshop plugin uses this convention) Tangent space only
SetRayHitDistance f	- Ray hit distance (1.0 = lowpoly triangle bounding sphere radius) also NUMPAD +/- in 3D view
AntiAlias on/off	- Antialias filter (default=on)
BackBufferOnly on/off	- nVidia low end (Geforce4MX, etc) or driver bug fix, use this if generated map is black with text (obvious error)
DispmapFormat value	- Displacement map output format (default = 1) 0 = No output 1 = Grayscale signed 2 = Grayscale unsigned (not implemented yet) 3 = RGB 4 = IEEE float 32 bit
DiffuseFormat value	- Diffuse map output format (default = 3) 0 = No output 1 = Highpoly vertex color 2 = Lowpoly vertex color 3 = Highpoly material color 4 = Lowpoly material color 5 = Highpoly texture sample 6 = Lowpoly texture sample

VIEWING AND INFO

PrintModelInfo	- Prints model information
ShowNormalColors	- Shows normal colors on DOT3 low poly model (*)
ShowTangentSpaceVectors	- Show tangent space vectors at each vertex (*) (* only after renderbump)
ShowDiffuse on/off	- Show diffuse on DOT3 model
ShowRayDistanceHull on/off	- Toggle displace of ray distance (white pulsating thing)
ShowOnlyModel value	- Show only model 0 = All models 1 = High poly 2 = Low poly 3 = DOT3 lit low poly 4 = Goraud shaded low poly

EXTRA COMMANDS

NUMPAD 1/4	- Scan "debug-tri" among high- and low-poly model (marked as white triangle in two leftmost models)
NUMPAD +/-	- Increase/decrease Ray-hit-distance
CollapseToOneGroup	- Collapse a multi-group model to one group/mesh
ExportMs3dModel	- Export LowPoly to Milshape3D format (.MS3D)
AddNormals normal.tga height.tga strength	- Add a heightmap to a normalmap with strength between 0.0 - 25.0
HeightmapToNormalmap height.tga strength	- Convert heightmap to normalmap

ORB AS A PREVIEWER

```
ViewModel model.ext - Import Model for viewing (ext = ASE/OBJ/LWO)
SetViewDiffusemap i dir/filename.tga - Set diffuse (for ViewModel) (i = mat-index)
SetViewNormalmap i dir/filename.tga - Set normal map (for ViewModel) (i = mat-index)
```

IMPORTING MODELS

ORB currently supports three different formats: 3DS ASCII export (ASE), Wavefront (OBJ), and Lightwave (.LWO). The import functions of ORB accept standard triangle data, i.e., it reads vertices, texture coordinates, vertex colors, meshes/groups, smoothing groups, and materials. Import is done like this, (*note that highpoly should always come first*):

```
ImportModel dir/highpoly.ase dir/lowpoly.ase
```

```
ImportModel dir/highpoly.obj dir/lowpoly.obj
```

```
ImportModel dir/highpoly.lwo dir/lowpoly.lwo
```

Once a model is imported in ORB it will appear in the tool's 3D view as one high-poly and one low-poly (see Figure 1). By pressing NUMPAD 1/4 you can scan through all the triangles in the low-poly model and see which triangles in the high-poly model that they match. By pressing NUMPAD +/- keys, you may scale the *ray hit distance* setting. The ray hit distance is shown as a pulsating white hull (not included in Figure 1).

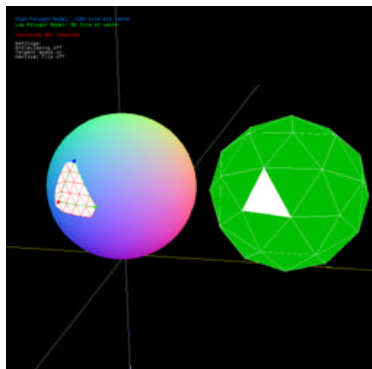


Figure 1. An imported model (high-poly and low-poly).

RENDER BUMP

There are currently four different renderbump commands: `Renderbump`, `RenderbumpMulti`, `RenderbumpHW`, and `RenderbumpHWMulti`. `Renderbump` and `RenderbumpMulti` outputs vertex color map (*_d.tga), normal map (*_local.tga), displacement map (*_disp.tga), and a displacement info text file (*_disp.txt). `RenderbumpHW` and `RenderbumpHWMulti`, however, only output normal maps.

`Renderbump dir/name` - This command renders one bump map using the texture mapping supplied in `lowpoly.ext`. (see Figure 2). This command uses the ray-casting version of the `Renderbump` algorithm, which is significantly slower than the hardware version, but produces better results in most cases. **BEST QUALITY!**

`RenderbumpMulti dir/name` - This command is similar to `Renderbump` except it renders one map per material found in `lowpoly.ext`. **BEST QUALITY!**

`RenderbumpHW dir/name` - This command renders one bump map using the texture mapping supplied in `lowpoly.ext`. (see Figure 2). This command uses the hardware version of the `renderbump` algorithm.

`RenderbumpHWMulti dir/name` - This command is similar to `RenderbumpHW` except it renders one bump map per material found in `lowpoly.ext`.

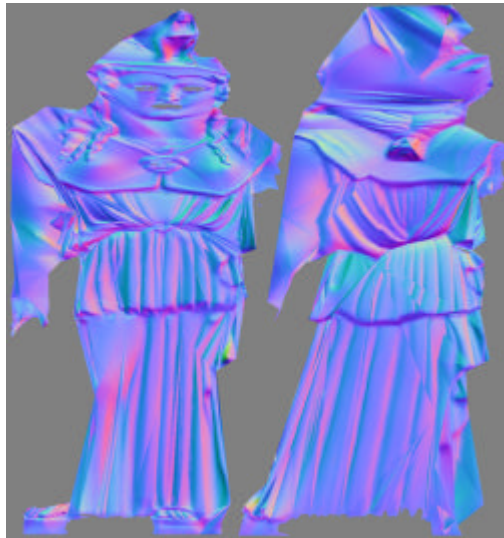


Figure 2. Texture mapping imported in low-poly model.

During `renderbump` execution, the result is displayed on screen. Once it is completed, two new models will appear in the tool's 3D view. As previously mentioned, the leftmost is the highpoly, the middle-left is the lowpoly. The middle-right is the result of the low-poly model with the newly rendered bump map using DOT3 bump-mapped per-pixel lighting. The rightmost model is a Goraud-shaded low-poly model (for comparison). (See Figure 4).

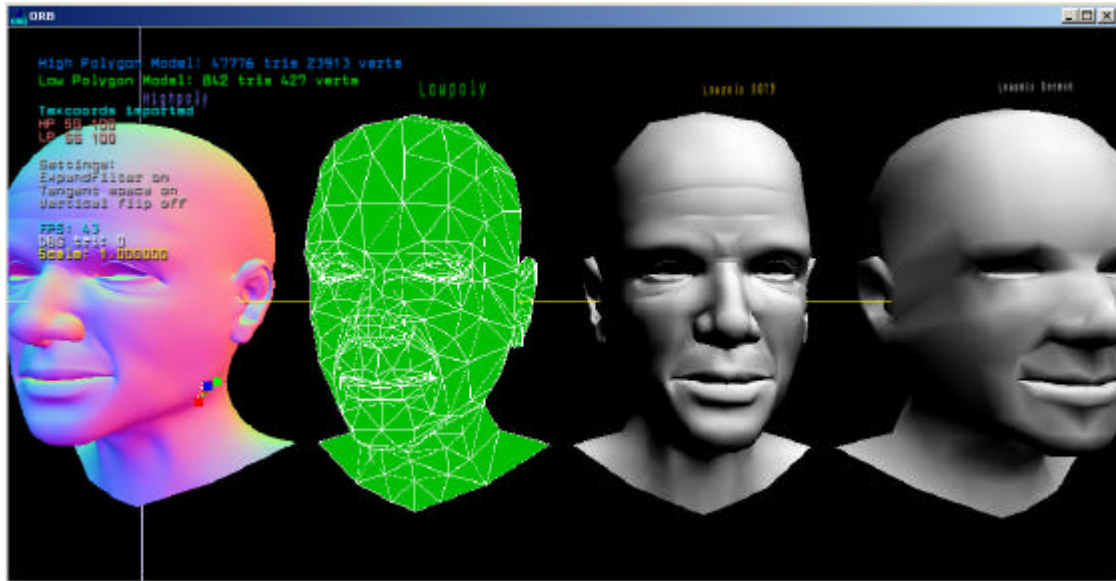


Figure 4. ORB's 3D view. (From left to right) The high-polygon model, the low-polygon model, the low-polygon model with DOT3 per-pixel lighting using the rendered bump map, and a Goraud-shaded version of the low-polygon model. (model by Derick Dressel www.BrainFaucet.com)

HINTS AND TROUBLE SHOOTING

- Texture coordinates set in low-poly.
- Smoothing groups should be set in high-poly *and* low-poly models. (Check **HP SG** and **LP SG** text in 3D view to see the percentage of triangles that have smoothing groups set. Remember that smoothing group 0 or off is ok, but that should be used for flat surfaces only).

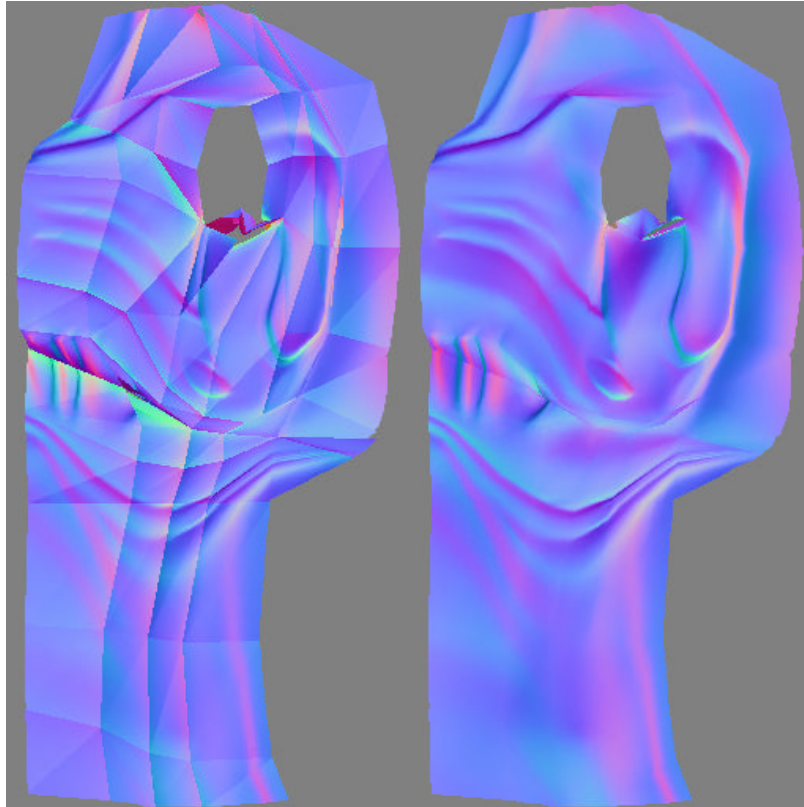
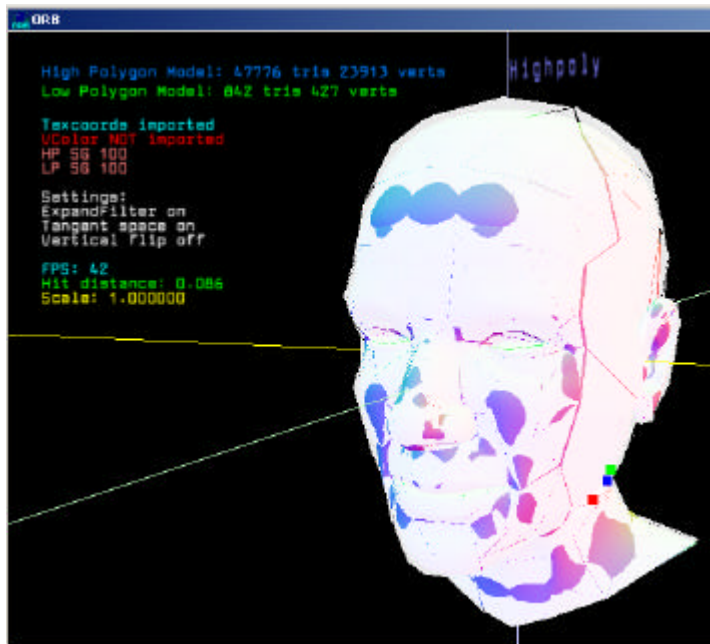


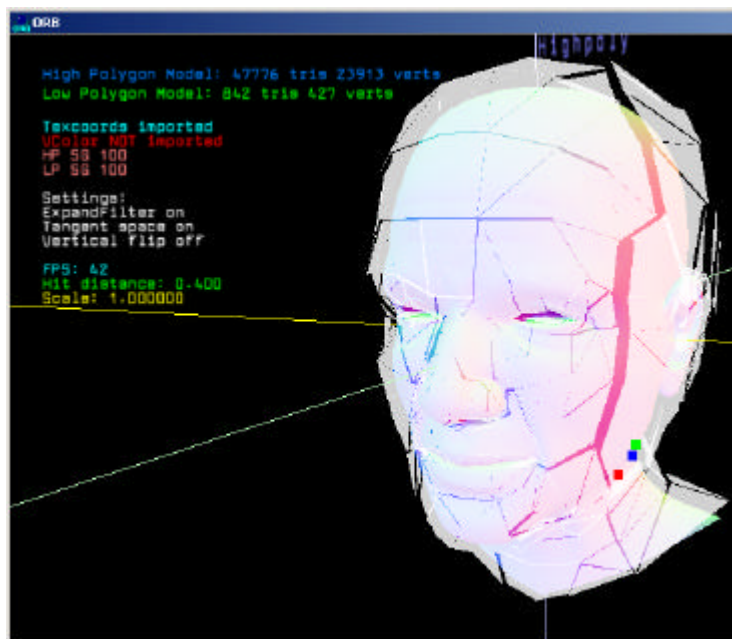
Image: This is the same model with smoothing groups off (left) and smoothing groups on (right). (*model by pestilence*).

- You must also make sure that vertices are welded, at least if they belong to smooth surfaces.
- On models with fine detail, use **Renderbump** and the setting '**MultiSample on**', it will produce significantly higher quality than standard **RenderbumpHW** algorithm. Use the HW for testing and then raycast for the final production.
- Model positioning. The high-poly and low-poly model should be in the same object space positions; triangle matching from low- to high-poly is made in object space. It should be centered around origo (0,0,0).
- Make sure all normals are pointing "outward", i.e. all vertices have counter-clockwise winding in the triangles. Mirroring objects in 3DS Max can flip normals and winding. Try using Reset X-form utility to fix this.

- Remember to tune the **ray-hit-distance** so that it is **large enough** to contain the high poly geometry and **small enough** not to hit other parts of the model. Use [NUMPAD +/-](#).

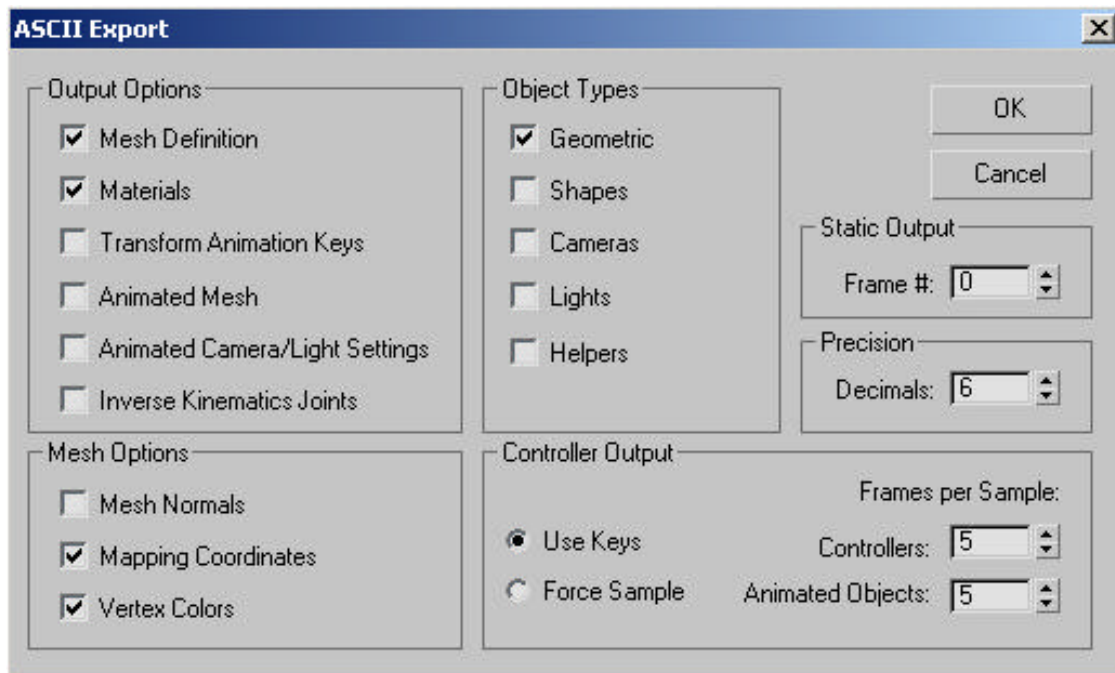


Ray hit distance setting is too low at **0.086** for this model.



Ray hit distance is good at **0.4** for this model.

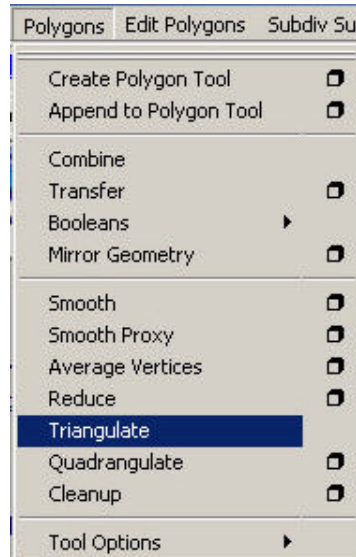
- RenderbumpHW command: The window must be visible when rendering a bump map.
- Renderbump command: The window must be visible during index buffer rendering, ORB will warn you if it isn't.
- Change screen resolution AND output map size in settings.ini (remember, both width and height must be power of 2, i.e. 512, 1024, otherwise results cannot be viewed in 3D view after Renderbump)
- For users with nVidia low-end card (Geforce2Go, Geforce4MX, etc) or old drivers, type 'BackBufferOnly on' in autoexec.cfg and everything should work fine.
- Known limitation: it has been reported that Renderbump doesn't work with Geforce4MX, this will hopefully be fixed in the future.
- .ASE format import is rather simple and do not handle all of ASE, it may crash if unexpected data is encountered, try exporting only basic geometry and materials, like this:



UV-data is read from mapping channel 1 only, all other channels are disregarded.

- ORB supports 1 level of **"Multi/Sub-Object"** material type in 3DS Max. I.e. you can have multi/sub object materials containing materials, but NOT a multi/sub-object material containing another multi/sub-object material.

- Import of **Maya** .OBJ files is also simple and doesn't handle the complete OBJ format. ORB expects triangles (3 vertices/face) and won't accept polygons (4 or more vertices / face). So tell Maya to use triangles. Unfortunately, OBJ doesn't export vertex colors. You'll get white in the diffuse output (if you choose vertex channel).



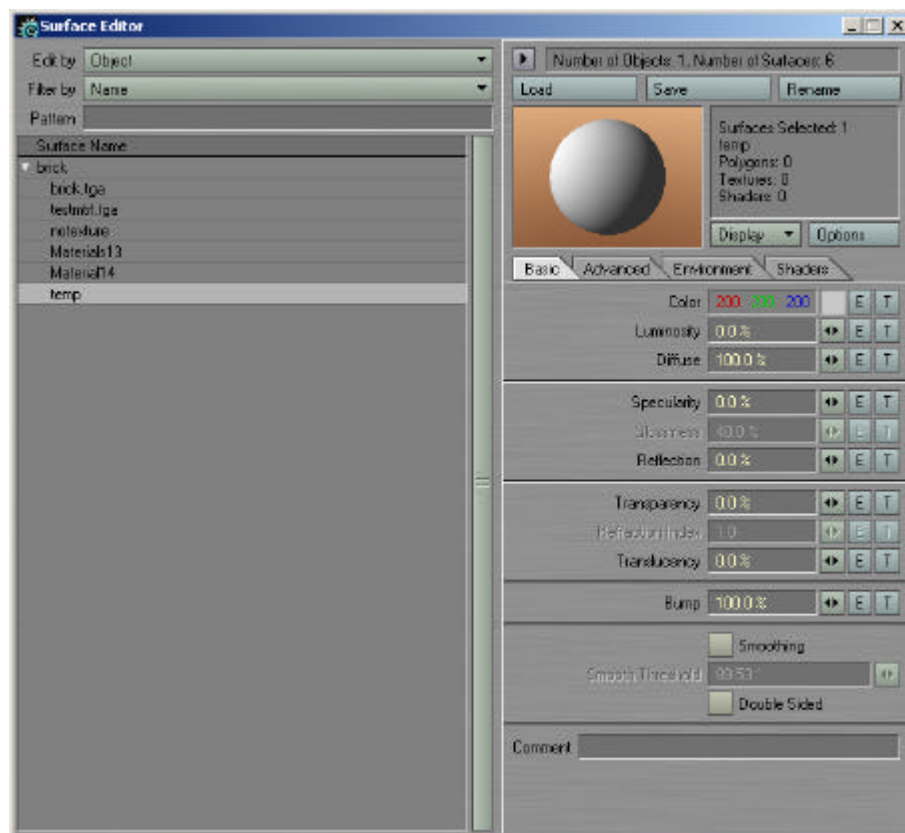
- Import of **Lightwave 3D** .LWO files is also simple and doesn't handle the complete LWO format. ORB expects triangles (3 vertices/face) and won't accept polygons (4 or more vertices / face). Before saving a lwo file for use with ORB, do this:
 - Construct -> Convert -> Freeze
 - Construct -> Subdivide -> Triple
- You can convert the 3DS MAX and Lightwave's coordinate systems to that in ORB if you want, set the variables before doing import:
 - 3DS MAX: "FlipYZWhen3dsImport on"
 - Lighwave: "NegZWhenLWOImport on"
- **Error correction.** There are two kinds of error correction (holes, misses in ray casting), both can be toggled on/off in the console or .cfg bat files.
 - **ExpandFilter**= stretches out pixels after map is rendered
 - **MultiPassErrorCorrection** = Runs multiple passes (5 passes) with increasing error margin

- **DiffuseFormat.** There are 7 sources for diffuse color output:

- 0 = (no output)
- 1 = Highpoly vertex color
- 2 = Lowpoly vertex color
- 3 = Highpoly material color
- 4 = Lowpoly material color
- 5 = Highpoly texture sample
- 6 = Lowpoly texture sample

The input texture must be TGA and is loaded like this:

- **OBJ:** In the .mtl file, the line **map_Kd texture_path** tells ORB where to find the file.
- **ASE:** In the material list, ORB looks for the ***BITMAP "filename.tga"** entry.
- **LWO:** For this format, a simple solution was made, the name of the surface is the path/filename, and .tga should be included, like this:



- **Diffuse output notes:**
 - Examples of valid paths (for all formats) are:
 - test.tga
 - C:\temp\test.tga
 - director/test.tga
 - ../directory/test
 - Material UV-Tiling and UV-Offset is read for ASE format only.
- **Displacement map format.** There are currently three displacement map formats implemented:
 - **Grayscaled signed:**
 - 0 = max neg displacement
 - 127 = zero displacement
 - 255 = max positive displacement
 - **RGB**
 - Red channel = positive displacement
 - Green channel = negative displacement
 - Blue channel = (unused)
 - **IEEE Float 32bit**
 - This file starts with two unsigned shorts (width, then height), then the image is just as array of 32bit floats with the actual geometry displacements for each pixel.
- **Displacement info file (.txt).** The different values for *max negative displacement*, *zero displacement*, and *max positive displacement* is written to a file called **filename_disp.txt**
- **Displacement scaling.** Different rendering packages and shaders have different settings. Therefore, ORB gives you an option how to scale the different displacements; there are currently two modes which are set with the console command **DispmanMode**.
 - **DispmanMode 0**

- This is the fixed scaling mode, in the case of grayscale signed, max negative displacement is 0, zero displacement will always be 127, and max positive is 255
- **DispmapMode 1**
 - This is the varying scaling mode, in the case of grayscale signed, max negative is 0, **zero displacement is varied depending** on the difference between max positive and max negative. Example: If max negative displacement is -0.5, and max positive is +1.5, we will get these values
 - Max neg in file = 0 (black), in geometry = -0.5
 - Zero disp in file = 64 (dark gray, in geometry = 0.0)
 - Max pos in file = 255 (white), in geometry = +1.5
- **Command line arguments**
 - `-exec script.cfg` = Auto-run script.cfg (after autoexec.cfg)
 - `-width 1024` = Custom width
 - `-height 512` = Custom height

CONTACT

Please, send any information concerning bug reports, feedback on functionality, requests for features, etc. to:

Christian Seger

Societies of Computation Laboratory (SOCLAB),

Department of Software Engineering and Computer Science,

Blekinge Institute of Technology,

Box 520, Ronneby - Sweden.

christian.seger@bth.se