

FX Composer Scene Commands



Scene commands are a way to extend the behavior of FX Composer to cover interactions between objects, and events that happen before/during/after the scene is rendered. A simple example might be clearing the background to a color before drawing the scene. A more complex example might render the screen to a render target texture, and then run an edge detection algorithm on it before displaying the final result.

Scene commands are a very new feature, and as such have a limited set of options currently. In version 1.5 of FX Composer, more complex interactions will be possible.

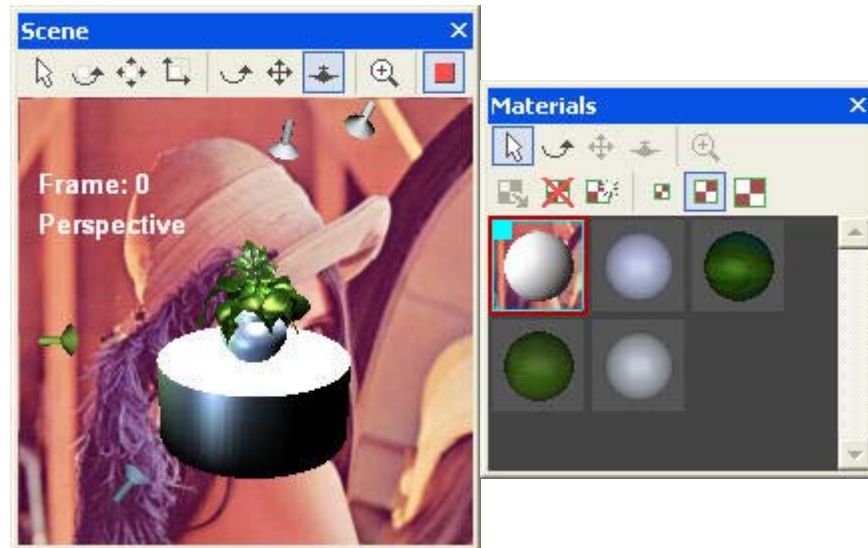
To create a scene command material, simply declare a regular .fx file, and add the XML commands to the file, either in a comment field or in an `#if 0` field (so that the parser doesn't complain when it sees them). An example is below, for the simple case of filling the backbuffer with a texture image before rendering the scene:

```
/*  
<scene_commands>  
<clear_color=".23,.23,.23"/>  
<drawquad technique="TextureClear"/>  
<drawobjects/>  
</scene_commands>
```

*/

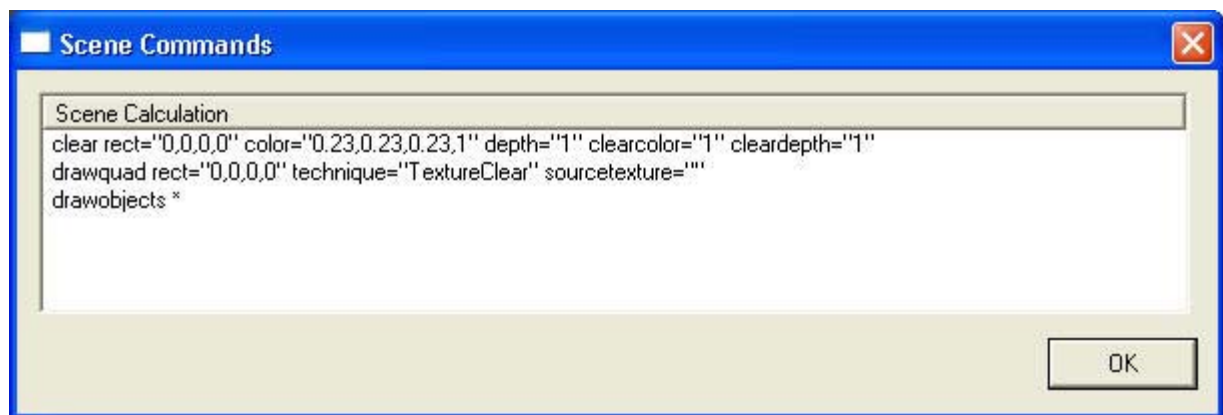
.... rest of .fx file....

Here's how this material looks in the materials panel, and on the scene:



Note that the material has a blue square at the top left, indicating that it is a scene command material. The scene shows the toothadil model, being rendered following the scene commands above. Right-clicking the material in the panel and choosing 'apply to scene' enables it, and the blue outline shows that it is the active scene command (you can only have one per scene).

Another right-click menu lets you look at the commands in a dialog ("Show scene commands"):



Note in the dialog that FX composer is also showing all the default options that weren't set in the XML, so it's possible to see the options you didn't set, and the default values FX Composer chose.

Scene command XML is delimited by:

```
<scene_commands>
```

```
... commands...
```

```
</scene_commands>
```

Any XML between these two elements is treated as a scene command. Here's a break down of the current commands:

```
<clear rect="x1,y1,x2,y2" color="r,g,b,a" depth = "fdepth" clearcolor="1 or 0" cleardepth="1 or 0"/>
```

The **clear** command causes the current rendertarget to be cleared. You can specify the depth, color, the rectangle to clear and enables for each. The default command parameters clear the whole rectangle to black, with a depth of 1.

```
<drawquad rect="x1,y1,x2,y2" technique="MyTechnique" sourcetexture=""/>
```

The **drawquad** command is used to draw a screen-space quad over the current rendertarget. You can specify the rectangle, and a technique to use. The technique refers to a technique in the .fx file containing the scene commands. In this case, FX Composer would draw a rectangle on the screen and apply the "**MyTechnique**" while drawing the quad. Since the quad is drawn at the current rendertarget, and the technique can refer to previous rendertargets, you can use this command to apply full screen effects, such as blur, edge detect, etc.

Note: The "sourcetexture" parameter is not usually required. It is for the special case of a non-power of 2 rendertarget with mipmaps. In that scenario, FX Composer will look at the source texture you specify and setup texture coordinates in TEXCOORD0 which enable sampling of the rectangular region you allocated for the texture, since the driver probably made a power of 2 texture from your request. So if you allocated a 200x200 rendertarget with mipmaps, the driver probably created a 256x256 texture for you. The texcoords to correctly sample your rendertarget would then be (200/256). This isn't a common usage though, so in general there's no reason to worry about the sourcetexture parameter.

<drawobjects/>

The drawobjects command currently has no parameters. In response to this command, FX Composer will draw all objects in the current scene graph.

Future versions of FX Composer will have 2 additional drawobjects parameters:

list="sphere01,head_mesh"

These named lists will refer to objects in the current FX Composer scene graph/project. The current drawobjects command is equivalent to list="", meaning all objects.*

material="mymaterial"

The selected material will be force-applied to all items in the draw list. This enables the entire scene to be rendered using a particular effect, or a selection of objects using a particular effect.

<settarget texture="mytexture"/>

The settarget permanently changes the rendertarget that is being drawn to. If the "texture" parameter is an empty string, then the rendertarget is reset to the main backbuffer for the scene as allocated when the window was created. If the parameter contains a string, then the current .fx file is searched for that texture name, and if the texture was allocated as a rendertarget it will be set as the new target. All future scene commands will then refer to this rendertarget.

To support rendertarget texture parameters, additional annotations are available to declare the texture. The following example shows a typical usage of these annotations:

texture mytexture

<

string usage = "RenderTarget";

int width = 0; // 0 means 'render target size'

int height = 0;

int levels = 1;

bool DepthBuffer = true;

string format = "X8R8G8B8";

>;

Note that the width/height of the texture has been declared as 0. In this case, FX Composer will maintain the rendertarget at the same size as the viewport. This is particularly useful for full screen effects. Additional annotations enable the creation of an associated depth buffer.

Future commands

To facilitate greater interaction with the FX Composer scene, additional commands will be added in version 1.5, as well as support for 'shared' texture parameters. It is

anticipated that this will enable features such as shadow buffering, etc. One obvious extension to the command set is:

<setviewpoint object="myobject"/>

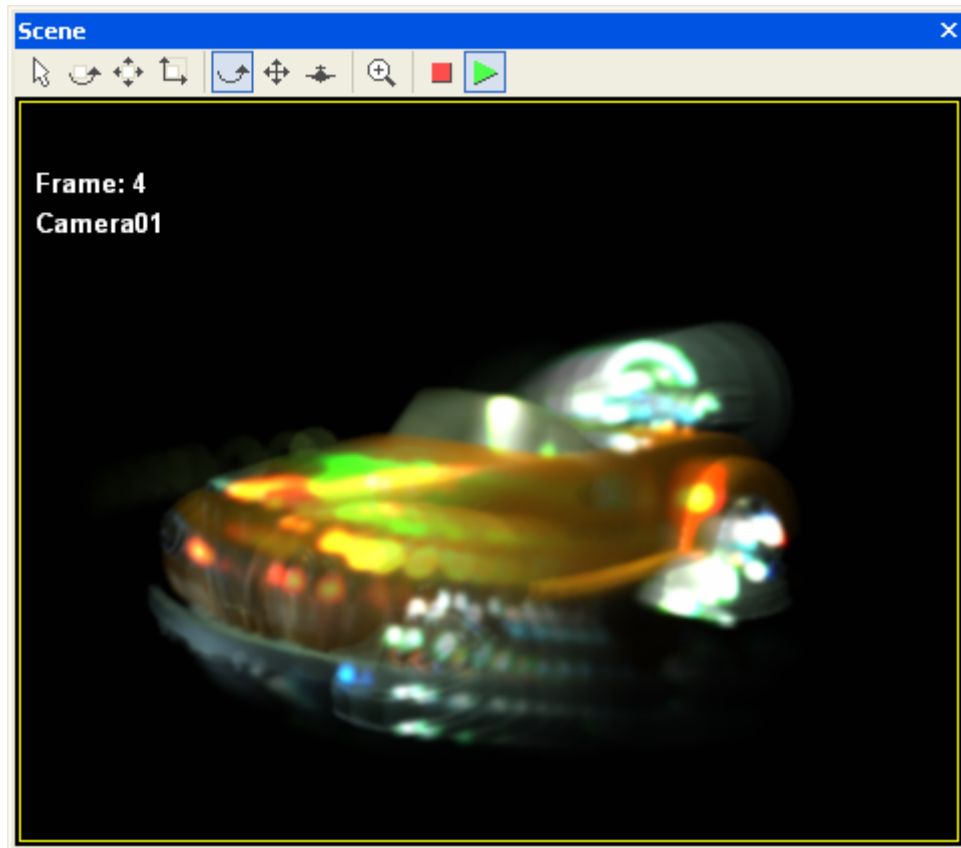
This enabling switching of the current lookat matrix to an object's point of view, such as a camera or a light.

Examples

The following examples demonstrate typical usage. Note that the .fx file contents have been omitted. See the demo projects shipping with fxcomposer in the 'fxcomposer' directory under MEDIA\ for full sources and animating demos.

Blur persistence of vision

```
<scene_commands>
<settarget texture="SceneMap"/>
<clear color="0,0,0"/>
<drawobjects/>
<settarget texture="HBlurMap"/>
<drawquad technique="GlowH" sourcetexture="SceneMap"/>
<settarget texture="FinalBlurMap"/>
<drawquad technique="GlowV" sourcetexture="HBlurMap"/>
<settarget texture="CompositeMap"/>
<drawquad technique="FinalComp" sourcetexture="FinalBlurMap"/>
<settarget texture="BlendMap"/>
<drawquad technique="Plop" sourcetexture="CompositeMap"/>
<settarget/>
<drawquad technique="Last" sourcetexture="BlendMap"/>
</scene_commands>
```



Edge Detection

```
<scene_commands>  
<settarget texture="SceneTexture"/>  
<clear color="0,0,0"/>  
<drawobjects/>  
<settarget/>  
<drawquad technique="Edges" sourcetexture="SceneTexture"/>  
</scene_commands>
```

(see image at start of page)