

Custom shader parameters

You can define and use own shader parameters like this:

Define the parameter in the material.xml: [Edit](#)

```
<material>
  <technique name="Techniques/skysphere.xml" />
  <parameter name="SunDir" value="0.64 0.64 0.64"/>
  <parameter name="SkyColor" value="0.9 0.5 0.5 1" />
</material>
```

Here I'm adding a Vector3/vec3 called SunDir and a RGBA color called SkyColor. Hint: there is no difference between RGB colors and Vector3/vec3 or between RGBA colors and Vector4/vec4, both are types with 3 respectively 4 floats.

In the Shader (here GLSL): [Edit](#)

```
// at the top above and outside all functions:
uniform vec3 cSunDir;
uniform vec4 cSkyColor;
...
void PS() // the parameters can also be used in the vertex shader (VS())
{
  float sun_intensity=clamp(dot(vNormal,cSunDir),0.0,1.0); // calculate a float with the dot product of the SunDir parameter and the vertex normal
  gl_FragColor=cSunColor*sun_intensity; // multiplying a vec4 / RGBA-Color with a float
}
```

In shaders the parameter names need to be preceded with a 'c'. See http://urho3d.github.io/documentation/1.32/_shaders.html .

Fandom may earn an affiliate commission on sales made from links on this page.

See http://en.wikibooks.org/wiki/GLSL_Programming/Vector_and_Matrix_Operations for an explanation of some of the shader functions.

You can also modify the parameters dynamically in you code:

```
materialSkySphere->SetShaderParameter("SunDir",Vector3(0,1,0));
```

Retrieved from "https://urho3d.fandom.com/wiki/Custom_shader_parameters?oldid=4584"

Categories:

Community content is available under [CC-BY-SA](#) unless otherwise noted.