

GLSL Programming/GLUT

In this section we present how to implement the GLSL concepts using C/C++, with GLUT as the basic OpenGL wrapper.

We will rely on the base code template built step-by-step in the [OpenGL Programming wikibook](#), for loading textures, models, as well as passing vertices and uniforms to the shaders. Almost all of the coding will occur in the shaders.

Note: Android users may want to have a look at the [GLUT wrapper](#).

When re-implementing obsoleted OpenGL 1.x features (such as lighting) using shaders, we'll try to use similar variable names, so that people familiar with the previous version can easily upgrade their knowledge.

Tutorials

Note that the tutorials assume that you read them in the order in which they are presented here, i.e. each tutorial will assume that you are familiar with the concepts and techniques introduced by previous tutorials. If you are new to GLSL you should at least read through the tutorials in the “Basics” section.

A number of these tutorials are already available for [Blender](#) and [Unity](#) - if you can port them, feel free to contribute!

Basics

- [Minimal Shader](#) (about game logic, Python scripts, and shaders)
- [RGB Cube](#) (about varyings)
- [Debugging of Shaders](#) (about attributes)
- [Shading in View Space](#) (about uniforms)

Transparent Surfaces

- [Cutaways](#) (about discarding fragments and triangle-face culling)
- [Transparency](#) (about blending)
- [Order-Independent Transparency](#) (about order-independent blending)
- [Silhouette Enhancement](#) (about transforming normal vectors)

Basic Lighting

- [Diffuse Reflection](#) (about per-vertex diffuse lighting and multiple light sources of different kinds)
- [Specular Highlights](#) (about per-vertex lighting)
- [Two-Sided Surfaces](#) (about two-sided per-vertex lighting)
- [Smooth Specular Highlights](#) (about per-pixel lighting)
- [Two-Sided Smooth Surfaces](#) (about two-sided per-pixel lighting)
- [Multiple Lights](#) (about for-loops for handling multiple light sources)

Basic Texturing

- [Textured Spheres](#) (about texturing a sphere)
- [Lighting Textured Surfaces](#) (about textures for diffuse lighting)
- [Glossy Textures](#) (about gloss mapping)
- [Transparent Textures](#) (about using alpha textures for discarding fragments, alpha testing, and blending)
- [Layers of Textures](#) (about multitexturing)

Textures in 3D

- [Lighting of Bumpy Surfaces](#) (about normal mapping)
- [Projection of Bumpy Surfaces](#) (about parallax mapping)
- [Cookies](#) (about projective texture mapping for shaping light)
- [Light Attenuation](#) (about texture mapping for light attenuation and lookup tables in general)
- [Projectors](#) (about projective texture mapping for projectors)

Environment Mapping

- [Reflecting Surfaces](#) (about reflection mapping)
- [Curved Glass](#) (about refraction mapping)
- [Skyboxes](#) (about rendering of environment maps as background)
- [Many Light Sources](#) (about image-based lighting)

Variations on Lighting

- [Brushed Metal](#) (about anisotropic specular reflection)
- [Specular Highlights at Silhouettes](#) (about the Fresnel factor for specular reflection)
- [Diffuse Reflection of Skylight](#) (about hemisphere lighting)
- [Translucent Surfaces](#) (about diffuse and forward-scattered transmission of backlight)
- [Translucent Bodies](#) (about diffuse lighting with reduced contrast and transmission of diffuse backlight at silhouettes)
- [Soft Shadows of Spheres](#) (about rendering the umbra and penumbra of spheres)
- [Toon Shading](#) (about non-photorealistic rendering)

Non-Standard Vertex Transformations

- [Screen Overlays](#) (about a direct transformation from object space to screen space)
- [Billboards](#) (about view-aligned projection of objects)
- [Nonlinear Deformations](#) (about vertex blending)
- [Shadows on Planes](#) (about projecting shadows onto planes)
- [Mirrors](#) (about mirroring objects and stenciling with the alpha buffer)

Unless stated otherwise, all example source code on this page is granted to the public domain.

Retrieved from "https://en.wikibooks.org/w/index.php?title=GLSL_Programming/GLUT&oldid=2389272"

This page was last edited on 10 August 2012, at 12:44.

Text is available under the Creative Commons Attribution-ShareAlike License.; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy.