GLSL Programming/Blender

Blender (http://www.blender.org) is a popular open source modeling and animation tool for Windows, MacOS X, Linux, and FreeBSD that includes a game engine. (Via GameKit (https://code.google.com/p/gamekit/) it might also support game development for Android and iOS in the future.) Since meshes, textures, light sources, etc. can be edited in a graphical user interface, it is easier to define complex scenes than in an OpenGL (ES) or WebGL application. Blender supports vertex and fragment shaders in GLSL (i.e. "GLSL programs"; not to be confused with the built-in "GLSL material" or "GLSL shading").

Preliminaries

Blender can be downloaded at the Blender Foundation's download page (http://www.blender.org/download/get-blender/). Some points should be noted:

- First, Blender's Python API (which is necessary to specify GLSL programs and OpenGL states) has been dramatically changed in version 2.5 of Blender; thus, most of the tutorials and examples about GLSL programming in Blender that you might find in the internet won't work with the current version.
- Second, Blender's Python API supports only a single render pass and it is unclear to me how to set OpenGL states (other than blending and uniform variables) for specific materials or objects. Thus, the possibilities are somewhat limited.
- Third, GLSL programs are limited to Blender's game engine; thus, they are only displayed when the game is started. (This is particularly inconvenient when editing shader parameters.)

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 or Blender you should at least read through the tutorials in the "Basics" section.

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 different kinds of light sources)
- 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 and blending)
- Layers of Textures (about multitexturing)

Advanced Texturing

- Lighting of Bumpy Surfaces (about normal mapping)
- Reflecting Surfaces (about reflection mapping)

Links

The most important documentation for programming GLSL in Blender appears to be:

- http://www.blender.org/documentation/blender_python_api_2_62_release/bge.types.html#bge.types.BL_Shader (class for setting shaders)
- http://www.blender.org/documentation/blender_python_api_2_62_release/bge.logic.html#shader (built-in uniforms)
- http://www.blender.org/documentation/blender_python_api_2_62_release/bge.types.html#bge.types.KX_BlenderMaterial (class for setting the blend equation)
- http://www.blender.org/documentation/blender_python_api_2_62_release/bge.logic.html#blender-material (blending constants)
- http://www.blender.org/documentation/blender_python_api_2_62_release/bgl.html (low-level access to OpenGL states)

There is a script to export the vertex and fragment shaders of Blender's internal GLSL programs. However, these shaders (and the uniforms they use) are generated automatically; thus, they are not very readable and not well suited for learning GLSL: http://wiki.blender.org/index.php/Extensions:2.6/Py/Scripts/Game_Engine/Export_GLSL

GLSL Shaders for Image Postprocessing

- http://wiki.blender.org/index.php/Doc:2.6/Manual/Game_Engine/Logic/Actuators/2D_Filters
- http://wiki.blender.org/index.php/Dev:Source/GameEngine/2.49/VideoTexture
- http://sites.google.com/site/blendergameengine/customfilter
- http://blenderartists.org/forum/showthread.php?184102-nicer-and-faster-SSAO

< GLSL Programming