

# A GPU Shader Programming Workshop

Mike Bailey  
Oregon State University

Steve Cunningham  
Grinnell College

## 1. Introduction

Shader programming is fast becoming an essential skill for computer graphics students. Using shaders, programmers have the flexibility to perform amazing vertex-by-vertex and pixel-by-pixel effects, combined with the parallel-processor performance to use shaders in interactive graphics. The emergence of shader programming is having profound effects on all areas of computer graphics including science, engineering, art, animation, and gaming. Because it is new, experience with teaching it is scarce. We taught such a class at OSU last spring, and thus have experience with teaching the topic, along with making up classroom examples and assignments. As part of that class, we developed a hands-on program called *glman*. *glman* allows students to create a shader scene description file which not only creates the 3D scene, but creates an interactive user interface to adjust shader parameters. Our experience is that *glman* is flexible enough to demonstrate and experiment with many shader concepts, and creates a fast and fun learning curve for the students.

We propose using *glman* as a basis for a SIGGRAPH Educators workshop. At the very least, it would involve presentations and live demos. But, the goal really is to obtain some number of shader-enabled graphics system loaners and teach this workshop hands-on. All attendees will be given (or can download on the web) the *glman* software, documentation, class notes, and many example files. Also, as we are currently writing a textbook on this topic, we will have very good class notes for the attendees. Thus, this workshop would be a shortcut to anyone who wants to offer such a course at their own institution.

## 2. *glman* In Action

*glman* is a program that was written to help teach the OpenGL Shading Language (GLSL). It uses an input file type called GLIB (GL Interface Bytestream), which is modeled after the style of the RenderMan Interface Bytestream (RIB). *glman* reads a GLIB file as well as one or more vertex and fragment shader files. It then creates the requested scene, activates the requested shaders, and creates sliders for user-defined global variables.

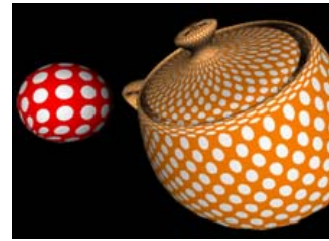
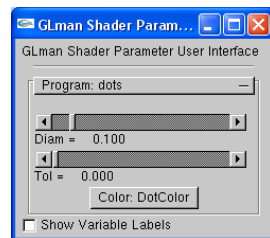
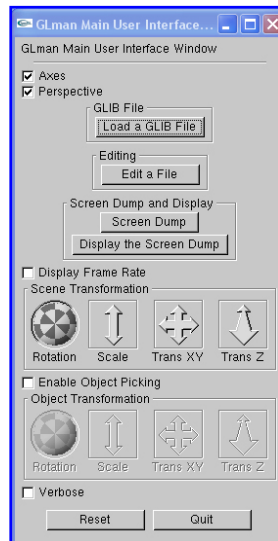


Figure 1: Procedural Dots Computed in Model Coordinates



Figure 2: Procedural Color Blurring

*glman* also provides a Perlin noise 3D texture for use in the shaders as shown here.

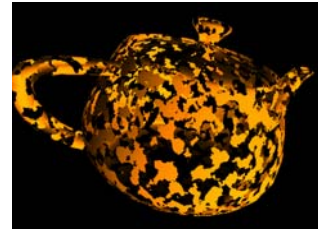


Figure 3: Noise-based erosion shader

*glman* can also be used for clever image manipulation, as shown here.



Figure 4: Interactive "Toon Rendering" for Non-photorealistic Architectural Visualization

## 3. Web Availability

The *glman* program and documentation can be obtained at:

<http://eecs.oregonstate.edu/~mjb/glman>