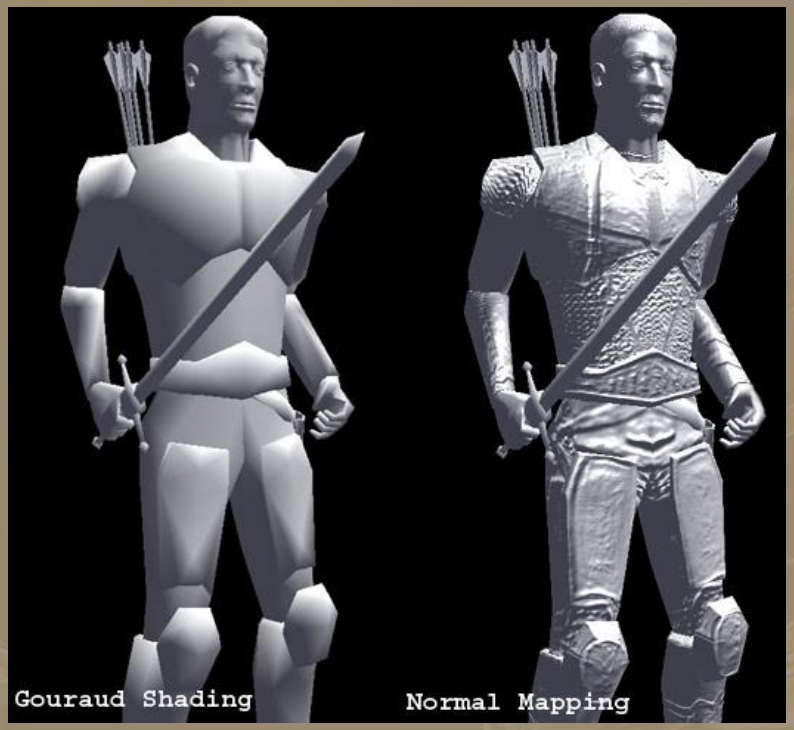
GLSL Basics

Discussion Lecture for CS418

Fall 2012

TA: Gong Chen

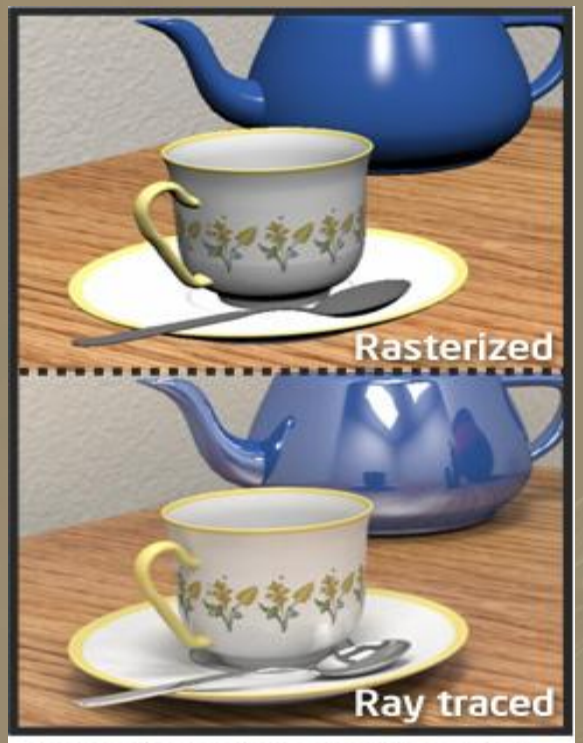
Shader Power Includes….



Normal Mapping

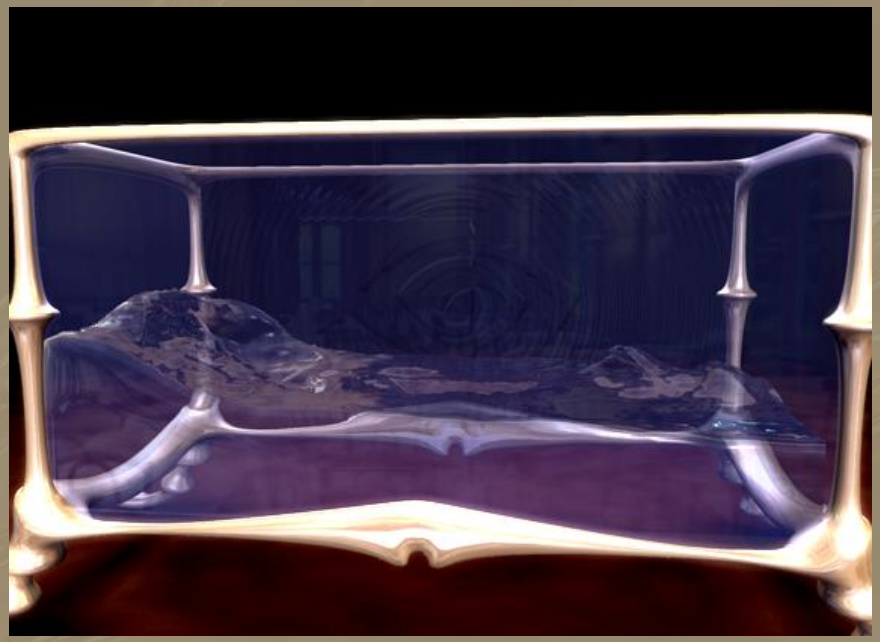
Real-Time Procedural Geometry (Geforce 8800)

Shader Power Includes….



Comparison between ray-traced and rasterized image (photo: Intel)

Real-Time Ray Tracing



Real-Time Fluid Simulation

And a lot more !

Today's Agenda

- Basic Set up
- Uniform, Attributes, and Varying
  - http://nehe.gamedev.net/article/glsl_an_introduction/25007/
- Passing Variables
  - http://www.lighthouse3d.com/tutorials/glsl-core-tutorial/communication-application-shader/
  - Please do read it. It is very helpful
- Hello Shader Example
- MP2 Q & A
- Informal Early Feedback Form

Hello-Shader

```
// Vertex Shader
void main()
{
    gl_Position = gl_ModelViewProjectionMatrix*gl_Vertex;
}
```

```
// Fragment Shader
void main()
{
    gl_FragColor = vec4(1, 0, 0, 1);
}
```

Setup Your Shader

- Windows
  - Download the new demo project posted in the discussion and resources section after Shaders
- Linux
  - Download the Hello Linux Shaders source code
- Mac
  - Use the linux source code and change the GL/glut.h to GLUT/glut.h
  - Good luck!

Variable Types : Attribute

```
attribute vec4 velocity;
void main()
{
   gl_Position = gl_Vertex + velocity*0.1;
}
```

- Attribute variable :
  - Change at each vertex
  - Usually positions, normals, texture coordinates
  - Read Only

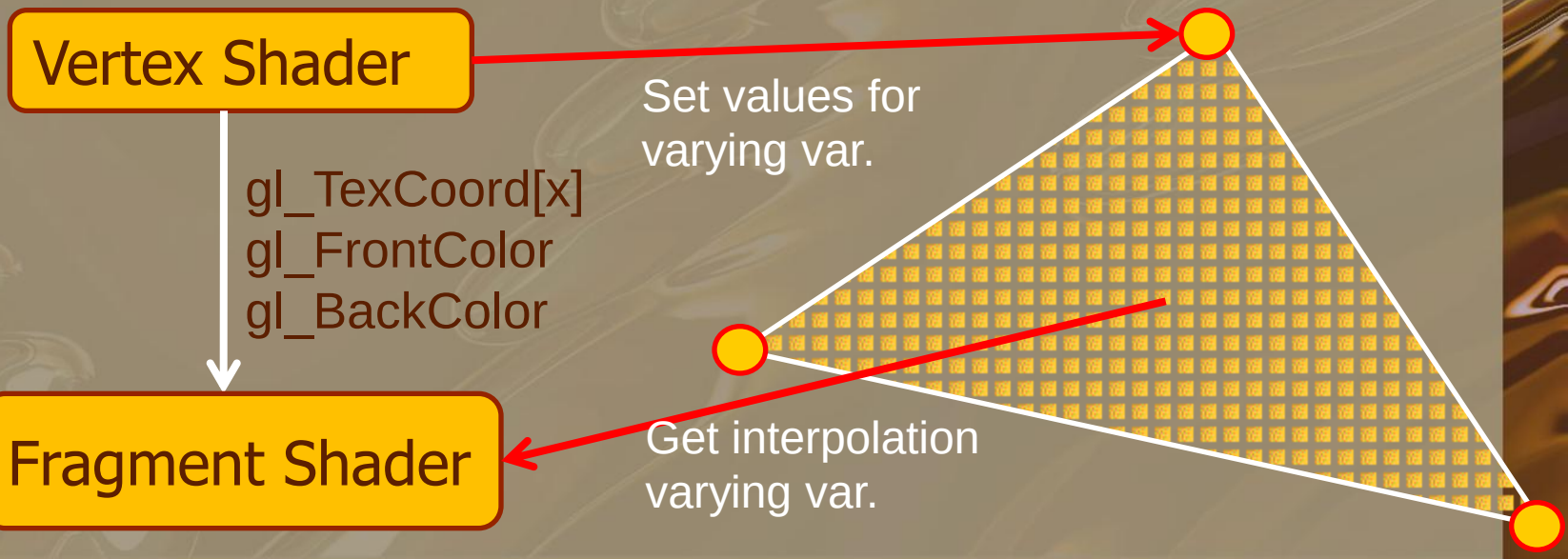
Variable Types : Uniform

```
attribute vec4 velocity;
uniform float time;
void main()
{
   mat4 m = gl_ModelViewProjectionMatrix;
   gl_Position = m*gl_Vertex + velocity*time;
}
```

- Uniform variable :
  - Values are the same for all vertices
  - Available for both vertex/fragment shader
  - Usually settings, light positions, gravity…
  - Read Only

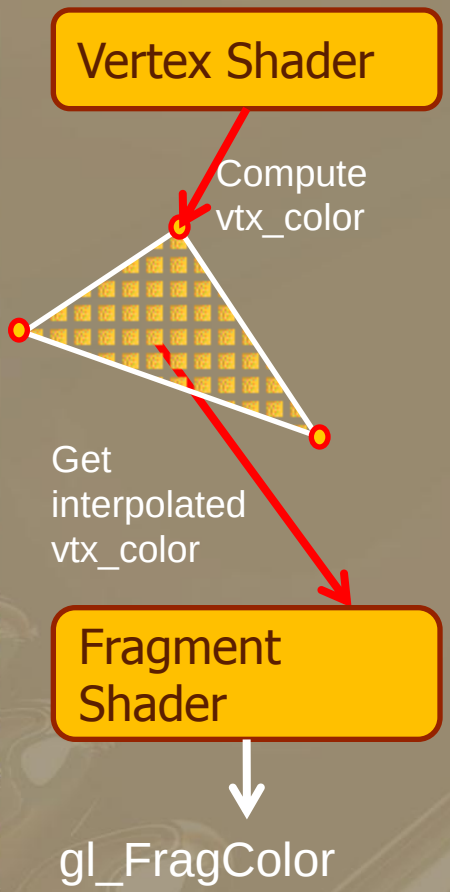
Variable Types : Varying

- Varying variable :
  - Passing data from Vertex shader to Fragment shader.
  - Data that will be interpolated during rasterization.
  - To create user defined varying variables, first declare in vertex shader then declare it again in pixel shader to access.



Variable Types : Varying

A Simple Gouraud Shading

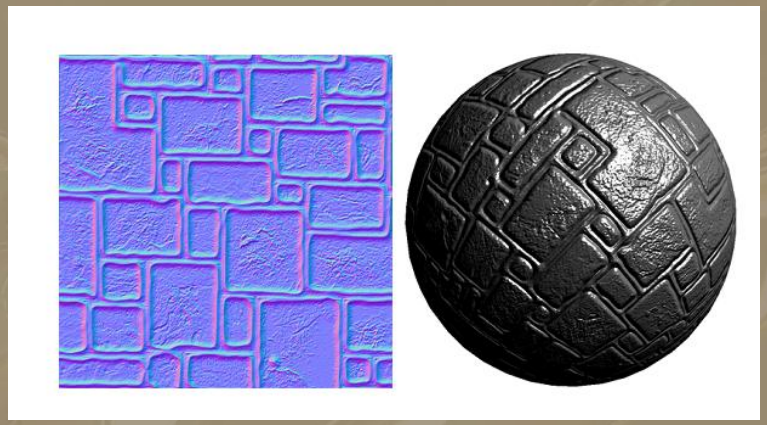



```
// Vertex Shader Program
varying vec4 vtx_color; //to fragment shader
void main()
{
   gl_Position = gl_Vertex;
   vtx_color = dot(gl_Normal,LightDir);
}
```

```
// Fragment Shader Program
varying vec4 vtx_color; // from vertex shader
void main()
{
    gl_FragColor =  vtx_color; //output vtx_color
}
```

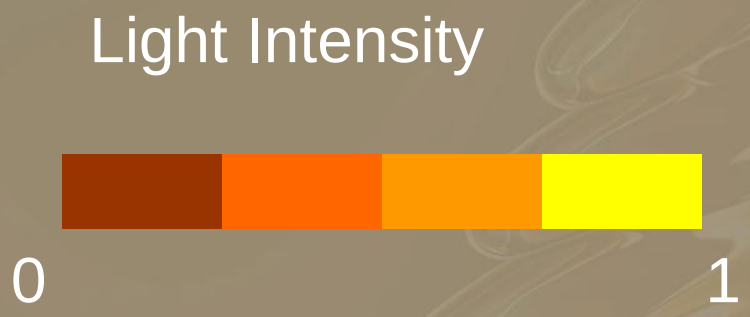
Setting Variables

- OpenGL already set up some default variables for you.
  - Uniform : gl_LightSource, gl_***Matrix, etc
  - Attributes : gl_Vertex, gl_Normal, etc.
- Define your own variable
  - http://www.lighthouse3d.com/tutorials/glsl-core-tutorial/communication-application-shader/
  - They declare attribute as in because of a newer GLSL version



Examples

- **Toon Shading**
  - **Use fixed color to replace original shading color.**



Q&A

- Try it earlier in case of any issues.