THE UNIVERSITY of NEW MEXICO

Introduction to Computer Graphics with WebGL

Ed Angel

Initializing Shaders

THE UNIVERSITY of NEW MEXICO

initShaders()

- Read shaders
- Compile shaders
- Create a program object
- Link everything together
- Link variables in application with variables in shaders
  - Vertex attributes
  - Uniform variables

THE UNIVERSITY of NEW MEXICO

Program Object

- Container for shaders
  - Can contain multiple shaders
  - Other GLSL functions

```
var program = gl.createProgram();

gl.attachShader( program, vertShdr );
gl.attachShader( program, fragShdr );
gl.linkProgram( program );
```

Reading a Shader

- Shaders are added to the program object and compiled
- Usual method of passing a shader is as a null-terminated string using the function

```
gl.shaderSource( fragShdr, fragElem.text );
```

- If shader is in HTML file, we can get it into application by `getElementById` method
- If the shader is in a file, we can write a reader to convert the file to a string

Adding a Vertex Shader

```
var vertShdr;
var vertElem =
    document.getElementById( vertexShaderId );

vertShdr = gl.createShader( gl.VERTEX_SHADER );

gl.shaderSource( vertShdr, vertElem.text );
gl.compileShader( vertShdr );

// after program object created
gl.attachShader( program, vertShdr );
```


Shader Reader

- Following code may be a security issue with some browsers if you try to run it locally
  - Cross Origin Request

```
function getShader(gl, shaderName, type) {
  var shader = gl.createShader(type);
  shaderScript = loadFileAJAX(shaderName);
if (!shaderScript) {
    alert("Could not find shader source:
        "+shaderName);
            }
}
```

Precision Declaration

- In GLSL for WebGL we must specify desired precision in fragment shaders
  - artifact inherited from OpenGL ES
  - ES must run on very simple embedded devices that may not support 32-bit floating point
  - All implementations must support mediump
  - No default for float in fragment shader
- Can use preprocessor directives (#ifdef) to check if highp supported and, if not, default to mediump

Pass Through Fragment Shader

```
#ifdef GL_FRAGMENT_SHADER_PRECISION_HIGH
  precision highp float;
#else
  precision mediump float;
#endif

varying vec4 fcolor;
void main(void)
{
     gl_FragColor = fcolor;
}
```


Error Checking 1

```
var vertElem =
 document.getElementById( vertexShaderId );
if ( !vertElem ) {
    alert( "Unable to load vertex shader "
          + vertexShaderId );
    return -1;
}
```

THE UNIVERSITY of NEW MEXICO

Error Checking 2

```
else {
vertShdr = gl.createShader( gl.VERTEX_SHADER );
 gl.shaderSource( vertShdr, vertElem.text );
 gl.compileShader( vertShdr );
    if ( !gl.getShaderParameter(vertShdr,
         gl.COMPILE_STATUS) ) {
       alert("Vertex shader failed to
         compile.  The error log is:" + "<pre>"
          + gl.getShaderInfoLog( vertShdr ) +
         "</pre>");
        return -1;
    }
 }
```