



OpenGL Shading Language

Bill Licea-Kane, AMD

SIGGRAPH 2009

Overview

- What's new in 1.40?
- What's new in 1.50?
- Some details
- A “porting” example
- Cool things you can do with GLSL 1.50
- NOT covered today - new ARB extensions related to shading
- And of course, Trivia

Version 1.40

- **Added**

- Uniform Blocks
- Layouts
- Rectangle Textures
- Texture Buffers
- gl_InstanceID
- Optional gl_Position
- Built-in function inverse()

- **Removed to “ARB_compatibility”**

- gl_ClipVertex
- Built-in vertex shader inputs
- Built-in uniforms
 - (exception - depth range parameters)
- Built-in varyings
- Built-in two sided color
- Built-in ftransform()
- Fixed function vertex processing
- Fixed function fragment processing

Version 1.50

- **Added**

- Profiles
- Geometry Shaders
 - layout for geometry shader in and out
 - 2D arrays deleted
- Uniform blocks generalized to interface blocks
- layout for gl_FragCoord
- multi-sample texture support
- gl_PrimitiveID available in fragment shaders
- determinant() built-in
- Vertex shader in can be arrays
- Vertex and geometry out may be struct
- Fragment and geometry in may be struct
- Interface limits
 - gl_MaxVertexOutputComponents
 - gl_MaxGeometryInputComponents
 - gl_MaxGeometryOutputComponents
 - gl_MaxFragmentInputComponents

- **Deprecated**

- gl_MaxVaryingComponents

ARB_compatibility -> profiles

```
#version 140 // core

#version 140 // core (belt)
#extension GL_ARB_compatibility : disable // (suspenders)

#version 140 // core
#extension GL_ARB_compatibility : enable // + compatibility

#version 150 // core

#version 150 core // core

#version 150 compatibility // core
// + compatibility
```

Interface Block & Layout

- Added in 140
- Generalized in 150

```
layout-qualifieropt interface-qualifier block-name  
{  
    member-list;  
} interface-nameopt;
```

Interface Qualifier

```
layout-qualifieropt interface-qualifier block-name  
{  
    member-list;  
} interface-nameopt;
```

Interface-qualifier:

```
uniform /* added in 140 */  
in      /* added in 150 */  
out     /* added in 150 */
```

Layout Qualifier - Uniform

```
layout-qualifier uniform;  
layout-qualifieropt uniform declaration;  
layout-qualifieropt uniform block-name  
{  
    member-list;  
} interface-nameopt'
```

uniform layout:

```
shared  
packed  
std140  
row_major  
column_major
```

Layout Qualifier – In (Geometry)

layout-qualifier in;

in layout:

points

lines

lines_adjacency

triangles

triangles_adjacency

Layout Qualifier – In (Fragment)

```
layout-qualifier in vec4 gl_FragCoord;  
in layout:  
    origin_upper_left  
    pixel_center_integer
```

*Note – this layout can be applied to the built-in gl_FragCoord
Default (unqualified) is origin at lower left
and pixel centers at half-pixels*

Layout Qualifier – Out (Geometry)

```
layout-qualifieropt out;
```

```
out layout:
```

```
points
```

```
line_strip
```

```
triangle_strip
```

```
max_vertices = integer-constant
```

Interface Block Examples

```
layout( std140 ) uniform TransformBlock {  
    mat4 ModelViewMatrix;  
    mat4 ModelViewProjectionMatrix;  
    mat3 NormalMatrix;  
};
```

```
layout( std140 ) uniform SkinningBlock {  
    layout( row_major ) mat4x3 BoneMatrix[10];  
} Arm[5]; /* starting in 150 */
```

```
layout( origin_upper_left, pixel_center_integer )  
    in vec4 gl_FragCoord;
```

Interface Block Examples

```
layout( points ) in;  
layout( triangle_strip, max_vertices = 4 ) out;  
  
in vertexIn {  
    vec4 position;  
    vec3 tangentNormal;  
    vec3 biNormal;  
}  
vertex[];  
  
out vertexOut {  
    vec4 position;  
    vec3 normal;  
    vec2 texCoord;  
}  
};
```

Multisample Texture Support

New Samplers:

sampler2DMS
isampler2DMS
usampler2DMS

New Built-in Functions:

ivec2 textureSize (sampler2DMS sampler)
ivec2 textureSize (isampler2DMS sampler)
ivec2 textureSize (usampler2DMS sampler)

*vec4 texelFetch (sampler2DMS sampler, ivec2 P,
int sample)*
*ivec4 texelFetch (isampler2DMS sampler, ivec2 P,
int sample)*
*uvec4 texelFetch (usampler2DMS sampler, ivec2 P,
int sample)*

Multisample Texture Array Support

New Samplers:

sampler2DMSArray
isampler2DMSArray
usampler2DMSArray

New Built-in Functions:

ivec2 textureSize (sampler2DMSArray sampler)
ivec2 textureSize (isampler2DMSArray sampler)
ivec2 textureSize (usampler2DMSArray sampler)

vec4 texelFetch (sampler2DMSArray sampler, ivec3 P, int sample)
ivec4 texelFetch (isampler2DMSArray sampler, ivec3 P, int sample)
uvec4 texelFetch (usampler2DMSArray sampler, ivec3 P, int sample)

“Porting” a “simple” shader to “core”

```
varying vec3 ECNormal;  
void main ()  
{  
    gl_Position = ftransform();  
    ECNormal = gl_NormalMatrix * gl_Normal;  
    gl_ClipVertex = gl_ModelViewMatrix * gl_Vertex;  
}
```

“Porting” a “simple” shader to “core”

```
varying vec3 ECNormal;  
void main ()  
{  
    gl_Position = ftransform();  
    ECNormal = gl_NormalMatrix * gl_Normal;  
    gl_ClipVertex = gl_ModelViewMatrix * gl_Vertex;  
}
```

“Porting” a “simple” shader to “core”

```
varying vec3 ECNormal;  
void main ()  
{  
    gl_Position = ftransform();  
    ECNormal = gl_NormalMatrix * gl_Normal;  
    gl_ClipVertex = gl_ModelViewMatrix * gl_Vertex;  
}
```

“Porting” a “simple” shader to “core”

```
varying vec3 ECNormal;  
void main ()  
{  
    gl_Position = ftransform();  
    ECNormal = gl_NormalMatrix * gl_Normal;  
    gl_ClipVertex = gl_ModelViewMatrix * gl_Vertex;  
}
```

“Porting” a “simple” shader to “core”

```
varying vec3 ECNormal;  
void main ()  
{  
    gl_Position = ftransform();  
    ECNormal = gl_NormalMatrix * gl_Normal;  
    gl_ClipVertex = gl_ModelViewMatrix * gl_Vertex;  
}
```

“Porting” a “simple” shader to “core”

```
varying vec3 ECNormal;  
void main ()  
{  
    gl_Position = ftransform();  
    ECNormal = gl_NormalMatrix * gl_Normal;  
    gl_ClipVertex = gl_ModelViewMatrix * gl_Vertex;  
}
```

“Porting” a “simple” shader to “core”

```
#version 150 core
```

```
varying vec3 ECNormal;
```

```
void main ()  
{  
    gl_Position = ftransform();  
    ECNormal = gl_NormalMatrix * gl_Normal;  
    gl_ClipVertex = gl_ModelViewMatrix * gl_Vertex;  
}
```

“Porting” a “simple” shader to “core”

```
#version 150 core
```

```
out vec3 ECNormal;
```

```
void main ()  
{  
    gl_Position = ftransform();  
    ECNormal = gl_NormalMatrix * gl_Normal;  
    gl_ClipVertex = gl_ModelViewMatrix * gl_Vertex;  
}
```

“Porting” a “simple” shader to “core”

```
#version 150 core
in  vec4 MCVertex;
in  vec3 MCNormal;
out vec3 ECNormal;
```

```
void main ()
{
    gl_Position = ftransform();
    ECNormal = gl_NormalMatrix * MCNormal;
    gl_ClipVertex = gl_ModelViewMatrix * MCVertex;
}
```

“Porting” a “simple” shader to “core”

```
#version 150 core
in  vec4 MCVertex;
in  vec3 MCNormal;
out vec3 ECNormal;

uniform transformBlock {
    mat4 MVMatrix;
    mat4 MVPMatrix;
    mat3 NormalMatrix;
};

void main ()
{
    gl_Position = ftransform();
    ECNormal = NormalMatrix * MCNormal;
    gl_ClipVertex = MVMatrix * MCVertex;
}
```

“Porting” a “simple” shader to “core”

```
#version 150 core
in  vec4 MCVertex;
in  vec3 MCNormal;
out vec3 ECNormal;
invariant gl_Position;
uniform transformBlock {
    mat4 MVMatrix;
    mat4 MVPMatrix;
    mat3 NormalMatrix;
};

void main ()
{
    gl_Position = MVPMatrix * MCVertex;
    ECNormal = NormalMatrix * MCNormal;
    gl_ClipVertex = MVMatrix * MCVertex;
}
```

“Porting” a “simple” shader to “core”

```
#version 150 core
in  vec4 MCVVertex;
in  vec3 MCNormal;
out vec3 ECNormal;
invariant gl_Position;
uniform transformBlock {
    mat4 MVMatrix;
    mat4 MVPMatrix;
    mat3 NormalMatrix;
};
uniform vec4 ClipPlane[8];
void main ()
{
    gl_Position = MVPMatrix * MCVVertex;
    ECNormal = NormalMatrix * MCNormal;
    vec4 ecVertex = MVMatrix * MCVVertex;
    for ( int i=0; i<8; i++ )
        gl_ClipDistance[i] = dot( ClipPlane[i], ecVertex );
}
```

“Porting” a “simple” shader to “core”

```
#version 150 core
in  vec4 MCVVertex;
in  vec3 MCNormal;
out vec3 ECNormal;
invariant gl_Position;
layout( std140 ) uniform transformBlock {
    mat4 MVMatrix;
    mat4 MVPMatrix;
    mat3 NormalMatrix;
};
uniform vec4 ClipPlane[8];
void main ()
{
    gl_Position = MVPMatrix * MCVVertex;
    ECNormal = NormalMatrix * MCNormal;
    vec4 ecVertex = MVMatrix * MCVVertex;
    for ( int i=0; i<8; i++ )
        gl_ClipDistance[i] = dot( ClipPlane[i], ecVertex );
}
```

Cool things you can do with GLSL 1.50

The addition of:

- **Framebuffer Objects (added in OpenGL 3.0)**

opened new pixel recirculation paths!

The addition of:

- **Transform Feedback (added in OpenGL 3.0 and GLSL 1.30)**
- **Uniform Buffer Objects (added in OpenGL 3.1 and GLSL 1.40)**
- **Geometry Shaders (added in OpenGL 3.2 and GLSL 1.50)**

opens up several new vertex recirculation paths!

Example - basic matrix operations

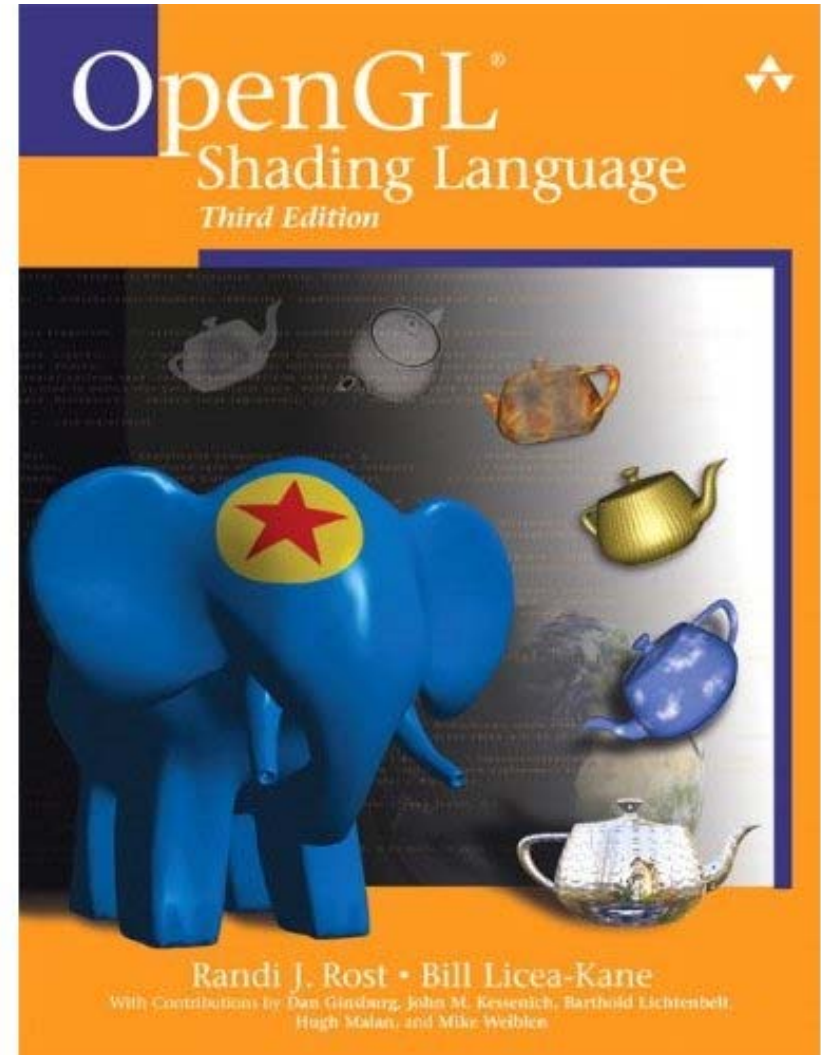
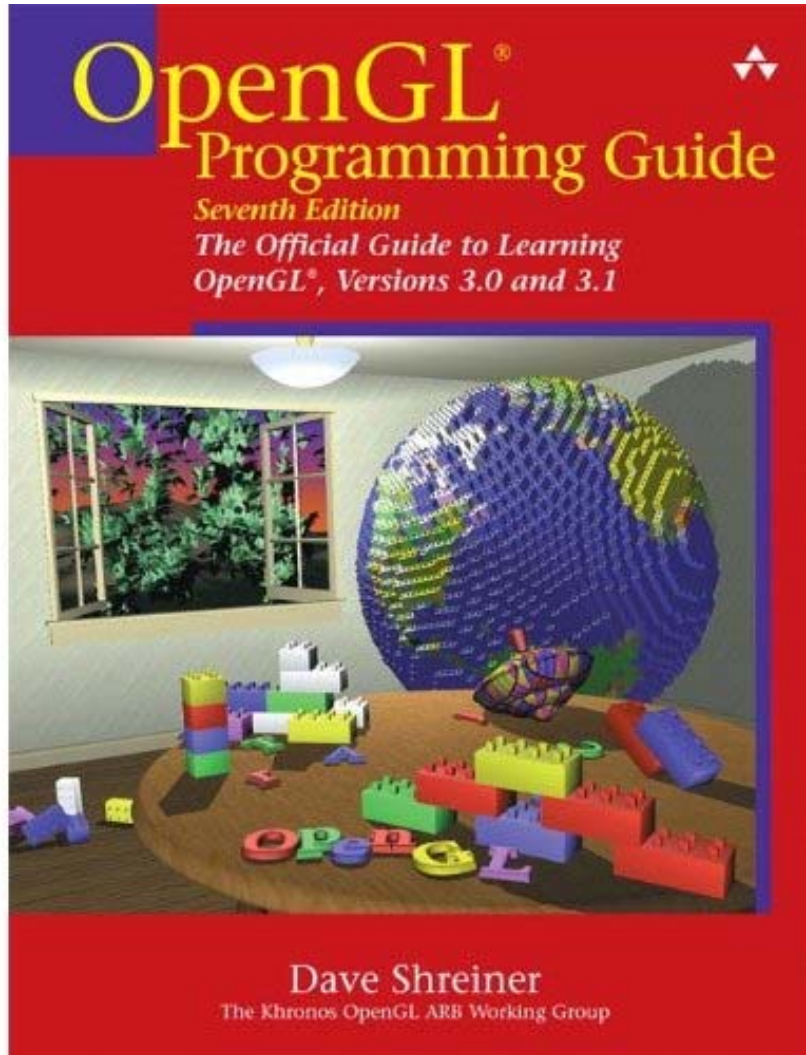
Caveat!

- It's simple
- It may not be a win
- But it is somewhat cool
- Generalizing is where the **real** cool is

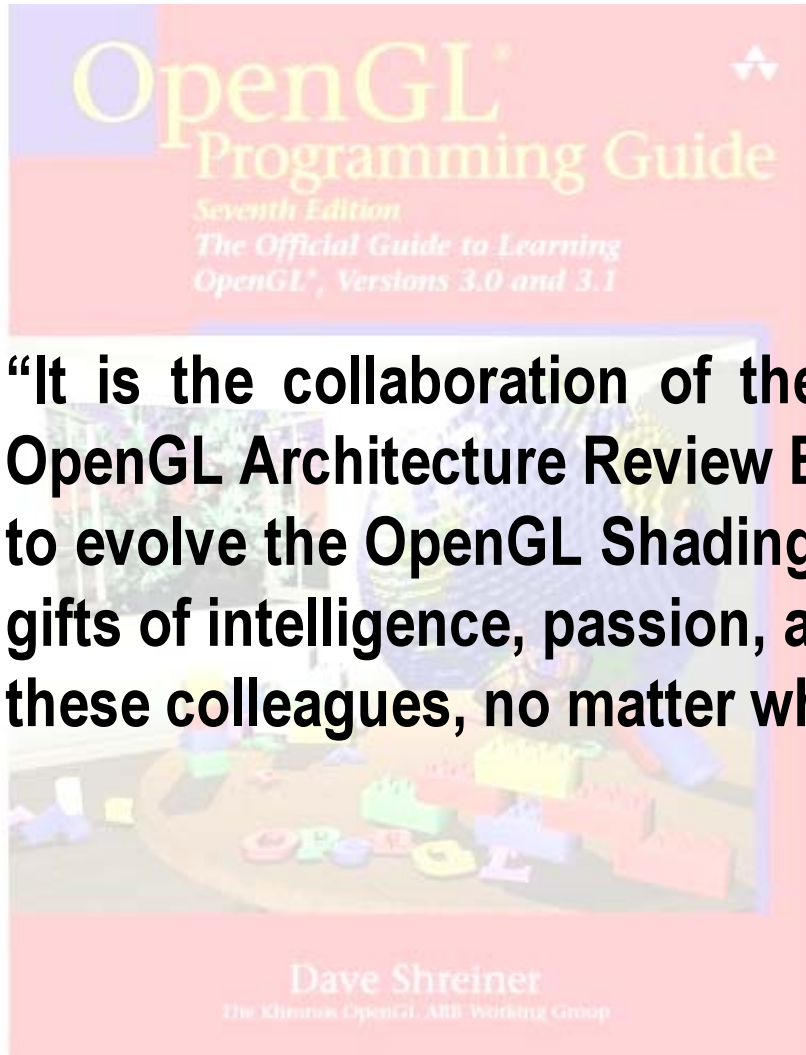
Transform Input Matrices to Useful Matrices

```
#version 150 core
// layout( std140 ) uniform transformBlock {
//     mat4 MVMatrix;
//     mat4 MVPMatrix;
//     mat3 NormalMatrix;
// };
uniform mat4 MVMatrix; // ModelView Matrix
uniform mat4 PMatrix; // Projection Matrix
out mat4    MV;        // ModelViewMatrix
out mat4    MVP;       // ModelViewProjection
out mat4x3  N;         // Normal
void main()
{
    MV = MVMatrix;
    MVP = PMatrix * MVMatrix;
    mat4 MVIT = transpose( inverse( MV ) );
    N = mat3x4( MVIT );
}
```

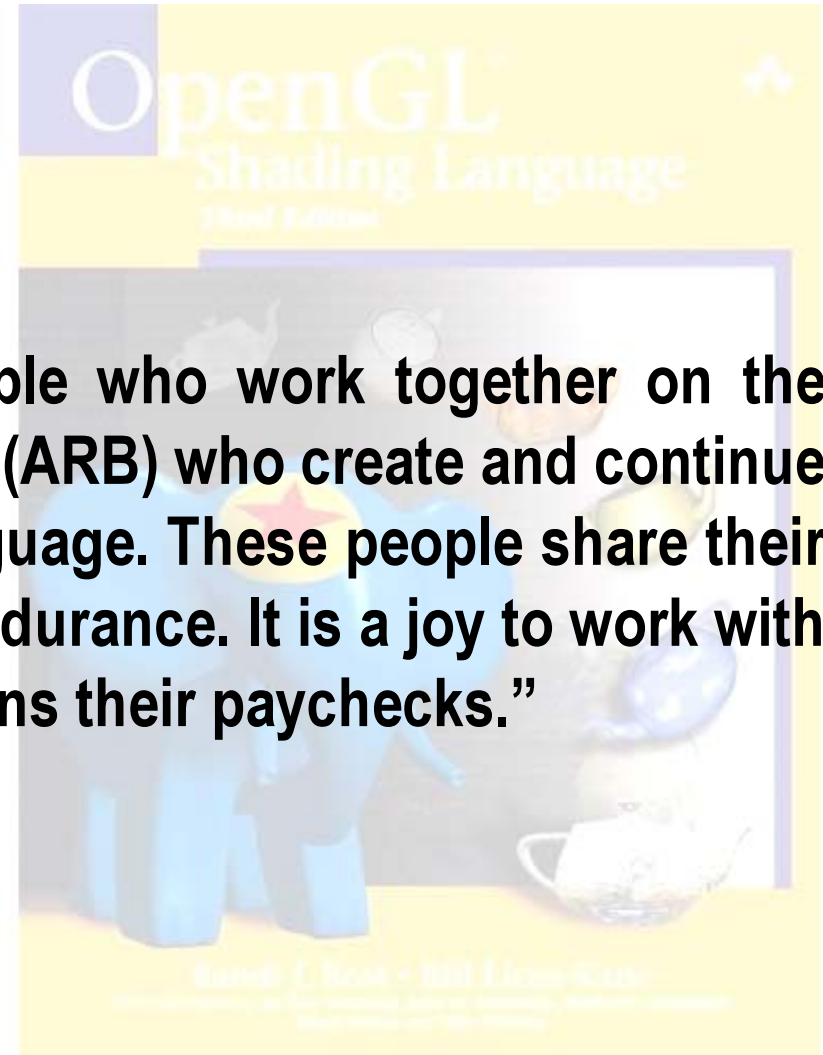
Brief shameless plug



Afterword



“It is the collaboration of the people who work together on the OpenGL Architecture Review Board (ARB) who create and continue to evolve the OpenGL Shading Language. These people share their gifts of intelligence, passion, and endurance. It is a joy to work with these colleagues, no matter who signs their paychecks.”



And now for something completely different...

TRIVIA!

(NOLA edition)

**Please answer
in the form of
a question.**

Category

Local Geography

Local Geography

**This was *NOT*
built on
Canal Street.**



Local Geography

The pronunciation
of this word:
“Tchoupitoulas”

Category

Local Cuisine

Local Cuisine

**The “trinity,”
a variation
or adaptation of
mirepoix.**

Local Cuisine

The “holy trinity.”

Category

Celebrations

Celebrations

The day before Mardi Gras.

Celebrations

The colors of Mardi Gras or Carnival.

Category

“Current Events”

“Current Events”

Originally Spanish,
then French,
it is now:
“Lagniappe.”