



# OpenGL Shading Language

**Bill Licea-Kane**  
**Chair OpenGL Shading Language Workgroup**  
**AMD**

# OpenGL Shading Language

- **In a Sentimental Mood**
- **Back to the Future**
- **Hints/Kinks**

# In a Sentimental Mood - 20 years ago!

- **April 1991 - Iris GL**
  - SGI and Microsoft (and IBM)

# In a Sentimental Mood - 20 years ago!

- **April 1991 - Iris GL**
  - SGI and Microsoft (and IBM)
- **April 1991 - ACE (Advanced Computing Environment)**
  - Compaq, Digital Equipment Corporation, Microsoft, MIPS, SCO
  - Many other members including SGI

# In a Sentimental Mood - 20 years ago!

- **April 1991 - Iris GL**
  - SGI and Microsoft (and IBM)
- **April 1991 - ACE (Advanced Computing Environment)**
  - Compaq, Digital Equipment Corporation, Microsoft, MIPS, SCO
  - Many other members including SGI
  - Scott McNealy, Sun Microsystems
    - "Those who can, do. Those who can't, consort." [1]

[1] Los Angeles Times, April 13, 1991, Sun Microsystems Aims at New Market Segment, Jonathan Weber

# In a Sentimental Mood - 20 years ago!

- **April 1991 - Iris GL**
  - SGI and Microsoft (and IBM)
- **April 1991 - ACE (Advanced Computing Environment)**
  - Compaq, Digital Equipment Corporation, Microsoft, MIPS, SCO
  - Many other members including SGI
  - Scott McNealy, Sun Microsystems
    - "Those who can, do. Those who can't, consort." [1]
  - Scott Cook, Intuit
    - "There has never been a technology product fostered by a coalition that has been successful." [2]

[1] Los Angeles Times, April 13, 1991, Sun Microsystems Aims at New Market Segment, Jonathan Weber

[2] Los Angeles Times, April 10, 1991, Consortium Seeks to Develop Faster Desktop Computer, Jonathan Weber

# 20 years ago!

- **September 1991**

- SILICON GRAPHICS OPENS IRIS GRAPHICS LIBRARY FOR GENERAL LICENSING
  - Compaq, Digital Equipment Corporation, Intel and Microsoft [1]

- **June 30, 1992**

- The OpenGL Graphics System: A Specification, Version 1.0
  - "This document describes the OpenGL graphics system: what it is, how it acts, and what is required to implement it." [2]
- OpenGL does not provide a programming language.

...

Programmability would conflict with keeping the API close to the hardware and thus with the goal of maximum performance." [3]

[1] SGI press release, 1991-09-17, [www.sgistuff.net/hardware/graphics/documents/iris-gl-announce.txt](http://www.sgistuff.net/hardware/graphics/documents/iris-gl-announce.txt)

[2] The OpenGL Graphics System: A Specification, Version 1.0 June 30, 1992, Segal, Akeley

[3] The Design of the OpenGL Graphics Interface, 1994, Segal, Akeley

# First question about Shading....

bi...@entropy.sps.mot.com (William C. Archibald) writes:

|>

|> A couple of quick queries:

|>

|> 1)Quite a few modern rendering techniques and renderers support

|> shader function calls at arbitrary points across a surface

|> being rendered. Most marble textures, for example, are

|> rendered this way with Perlin-ish noise functions.... Is there any way

|> to do this kind of thing in OpenGL?

Not with the current API. One could extend OpenGL in this manner...

comp.graphics.opengl, 1993-10-13, Kurt Akeley



# Brief Extension Digression

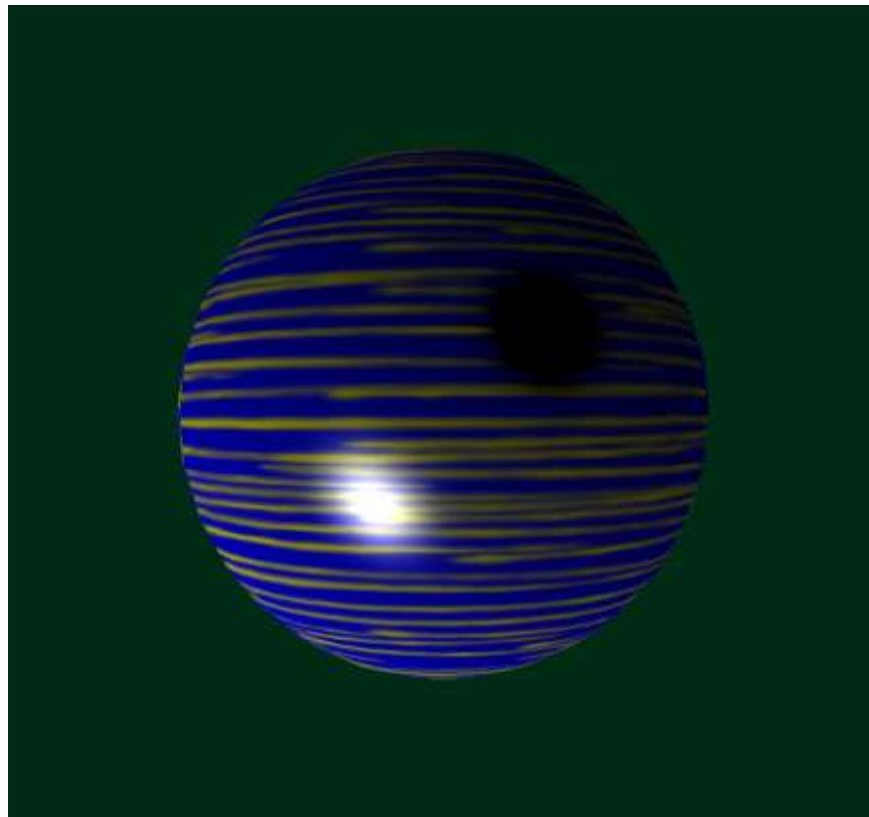
- **“One disadvantage of using procedure calls on such a fine grain is that it may result in poor performance if procedure calls are costly....**

**“If it turns out that fine-grained procedure calls are too expensive, then it may be necessary to add a few popular block formats to the OpenGL API or to provide a mechanism for defining such formats.” [1]**

- **OpenGL 1.1**
  - InterleavedArrays (meh, “popular block formats” – just a helper function)
  - Vertex Arrays (successful)

[1] The Design of the OpenGL Graphics Interface, 1994, Segal, Akeley

# Mechanism, Not Policy



# Digression – meanwhile, at the movies

- **“Shading is an important part of computer imagery, but shaders have been based on fixed models to which all surfaces must conform. As computer imagery becomes more sophisticated, surfaces have more complex shading characteristics and thus require a less rigid shading model.” [1]**

[1] Shade Trees, SIGGRAPH 1984, Cook

# Digression – meanwhile, at the movies

- **“Shading is an important part of computer imagery, but shaders have been based on fixed models to which all surfaces must conform. As computer imagery becomes more sophisticated, surfaces have more complex shading characteristics and thus require a less rigid shading model.” [1]**
- **“We introduce the concept of a Pixel Stream Editor. This forms the basis of an interactive synthesizer for designing highly realistic Computer Generated Imagery.” [2]**

[1] Shade Trees, SIGGRAPH 1984, Cook

[2] An Image Synthesizer, SIGGRAPH 1985, Perlin

# Digression – meanwhile, at the movies

- **“Shading is an important part of computer imagery, but shaders have been based on fixed models to which all surfaces must conform. As computer imagery becomes more sophisticated, surfaces have more complex shading characteristics and thus require a less rigid shading model.” [1]**
- **“We introduce the concept of a Pixel Stream Editor. This forms the basis of an interactive synthesizer for designing highly realistic Computer Generated Imagery.” [2]**
- **“A shading language provides a means to extend the shading and lighting formulae used by a rendering system.” [3]**

[1] Shade Trees, SIGGRAPH 1984, Cook

[2] An Image Synthesizer, SIGGRAPH 1985, Perlin

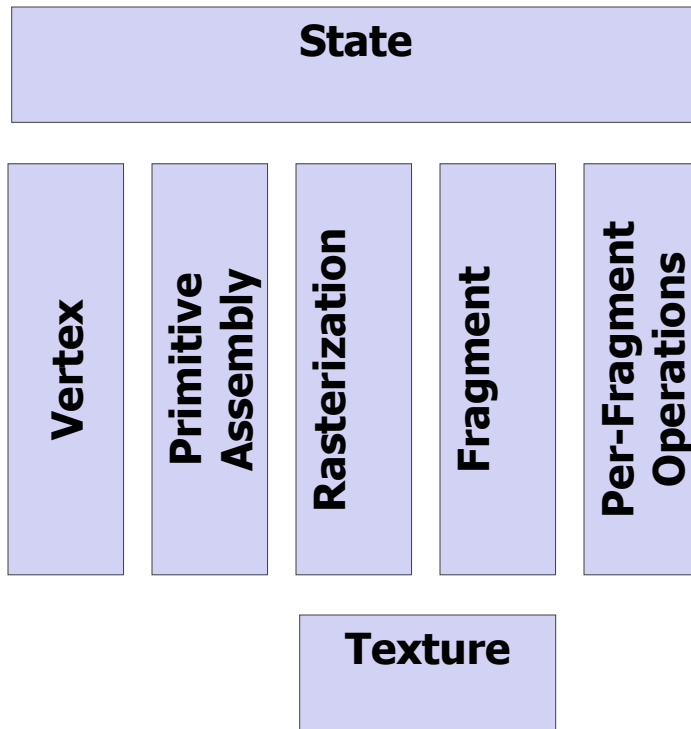
[3] A Language for Shading and Lighting Calculations, SIGGRAPH 1990, Hanrahan, Lawson

# In a Sentimental Mood – Bottom Line

- “Off-line” rendering evolved from “fixed function” to shaders
- OpenGL evolving as well

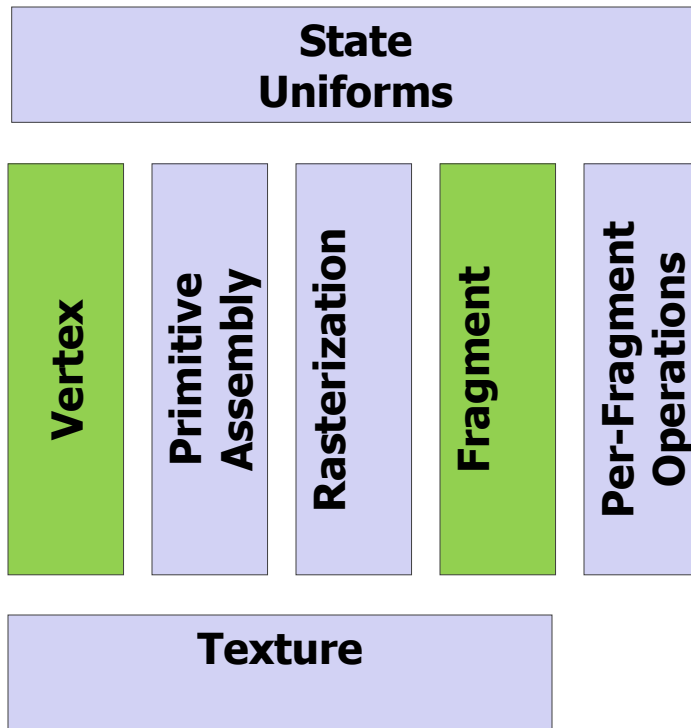
# Back to the Future

# OpenGL 1.x

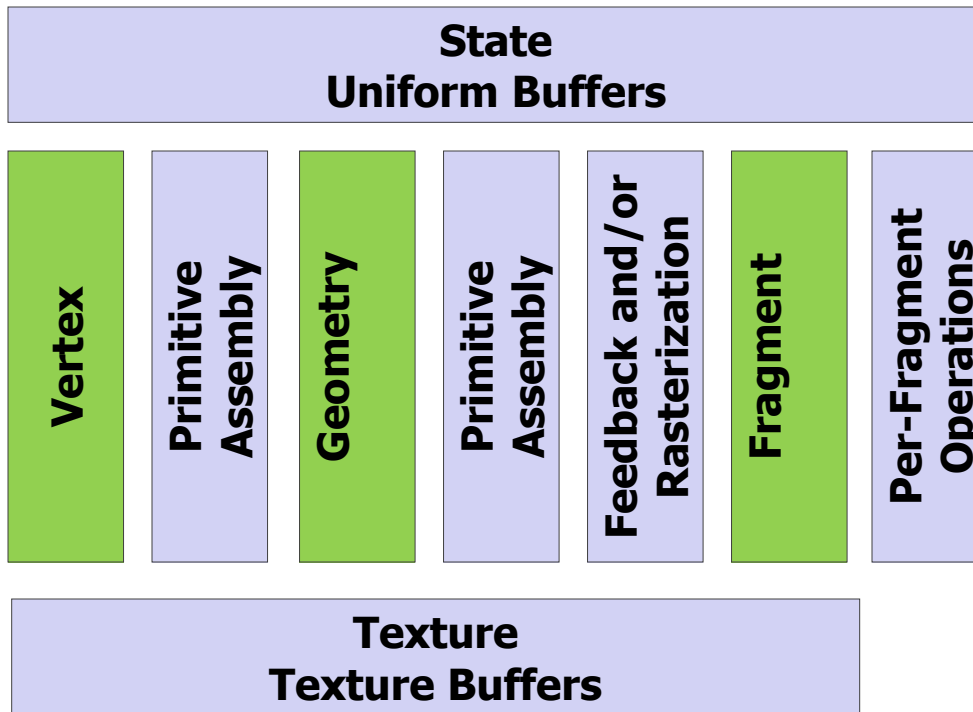




# OpenGL 2.x



# OpenGL 3.x



# OpenGL 4.x

**State  
Uniform Buffers**

**Vertex**

**Primitive  
Assembly**

**Tessellation  
Control**

**Primitive  
Assembly**

**Tessellation**

**Tessellation  
Evaluation**

**Primitive  
Assembly**

**Geometry**

**Primitive  
Assembly**

**Feedback and/or  
Rasterization**

**Fragment**

**Per-Fragment  
Operations**

**Texture  
Texture Buffers**

# Hints/Kinks

# “Dusty Deck” shaders

- **Yes, they work**
- **But...**

# “Dusty Deck” shaders

- Yes, they work
- But...

```
varying float Texcoord;  
sampler1D idMap;  
//...  
vec2 texel = texture1D( idMap, Texcoord ).xy;  
texel      = floor( texel * 255.0 + 0.125/256.0 );  
int i      = int( dot( texel, vec2( 1.0, 256.0 ) ) );
```

# “Dusty Deck” shaders

- Yes, they work
- But...

```
varying float Texcoord;  
sampler1D idMap;  
//...  
vec2 texel = texture1D( idMap, Texcoord ).xy;  
texel      = floor( texel * 255.0 + 0.125/256.0 );  
int i      = int( dot( texel, vec2( 1.0, 256.0 ) ) );
```

# “Dusty Deck” shaders

- Yes, they work
- But...

```
in    float Texcoord;  
sampler1D  idMap;  
//...  
vec2 texel = texture1D( idMap, Texcoord ).xy;  
texel      = floor( texel * 255.0 + 0.125/256.0 );  
int i      = int( dot( texel, vec2( 1.0, 256.0 ) ) );
```



# “Dusty Deck” shaders

- Yes, they work
- But...

```
in    float Texcoord;
sampler1D  idMap;
//...
vec2 texel = texture1D( idMap, Texcoord ).xy;
texel      = floor( texel * 255.0 + 0.125/256.0 );
int i      = int( dot( texel, vec2( 1.0, 256.0 ) ) );
```

# “Dusty Deck” shaders

- Yes, they work
- But...

```
in    float Texcoord;  
isampler2D idMap;  
//...  
int i    = texture1D( idMap, Texcoord ).x;
```

# “Dusty Deck” shaders

- Yes, they work
- But...

```
in    float Texcoord;  
isampler2D idMap;  
//...  
int i    = texture1D( idMap, Texcoord ).x;
```

# “Dusty Deck” shaders

- Yes, they work
- But, simple example

```
in    float Texcoord;  
isampler2D idMap;  
//...  
int i    = texture( idMap, Texcoord ).x;
```

# WIN/FAIL?

# WIN/FAIL?

- goto?

# WIN/FAIL

- goto – WIN!

# WIN/FAIL

- goto – WIN!
- noise?



# WIN/FAIL

- goto – WIN!
- noise – FAIL!

# WIN/FAIL

- goto – WIN!
- noise – FAIL!
- Shading language function man pages?

# WIN/FAIL

- **goto – WIN!**
- **noise – FAIL!**
- **Shading language function man pages – WIN!**
  - [www.opengl.org/sdk/docs/manglsl/](http://www.opengl.org/sdk/docs/manglsl/)
  - (Thank you to Graham Sellers)

# Questions?