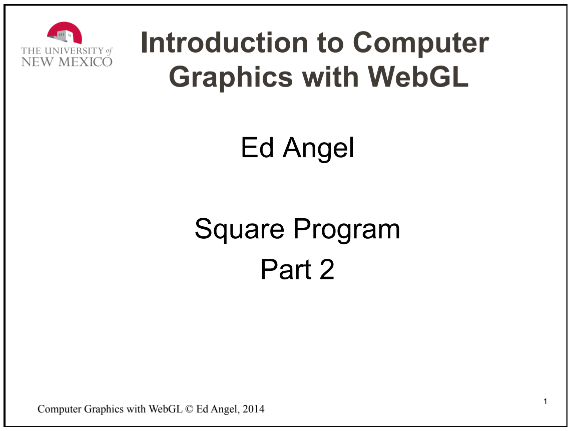
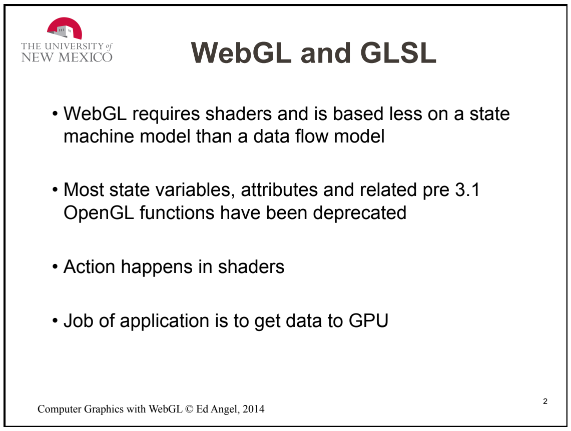


Introduction to Computer Graphics with WebGL

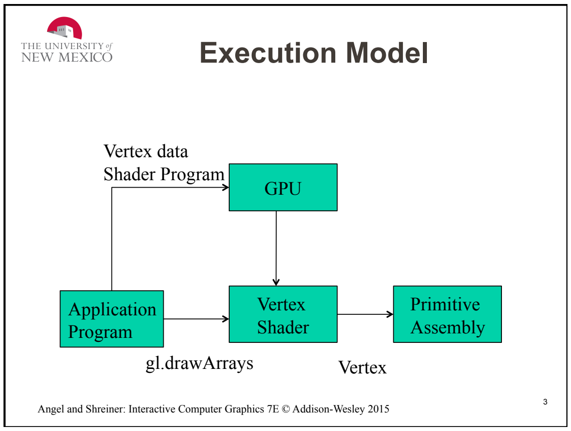
Ed Angel

Square Program
Part 2

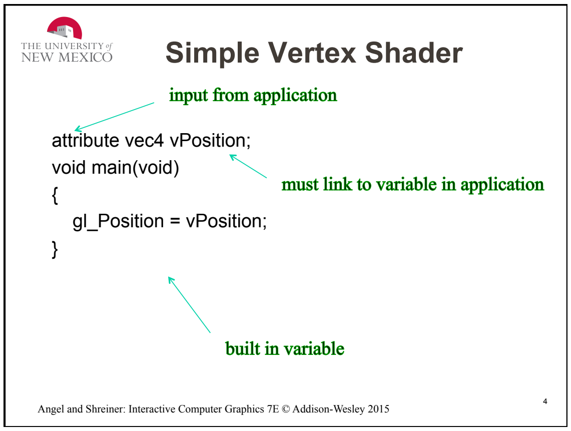


WebGL and GLSL

- WebGL requires shaders and is based less on a state machine model than a data flow model
- Most state variables, attributes and related pre 3.1 OpenGL functions have been deprecated
- Action happens in shaders
- Job of application is to get data to GPU



Execution Model



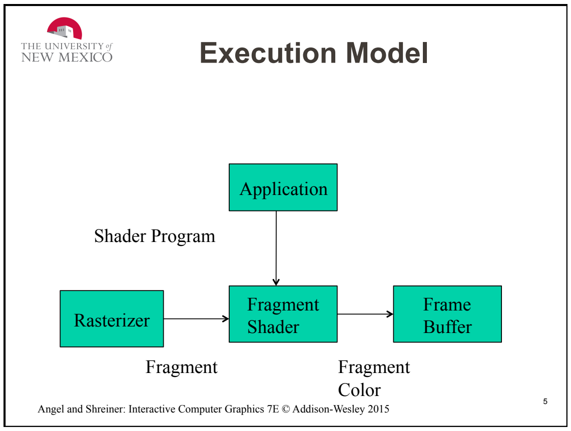
Simple Vertex Shader

input from application

```
attribute vec4 vPosition;
void main(void)
{
   gl_Position = vPosition;
}
```

must link to variable in application

built in variable



Execution Model

Application

Shader Program

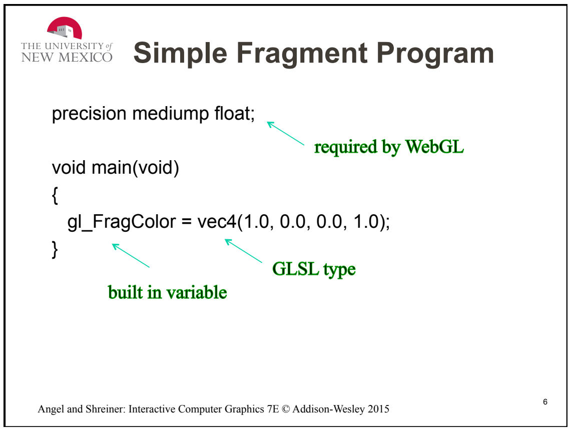
Rasterizer

Fragment Shader

Frame Buffer

Fragment

Fragment Color



Simple Fragment Program

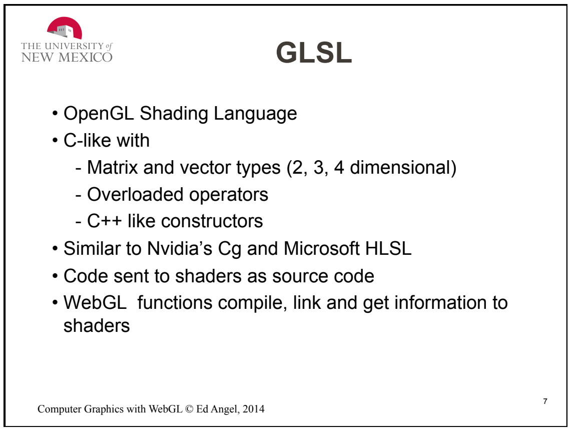
```
precision mediump float;

void main(void)
{
  gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);
}
```

required by WebGL

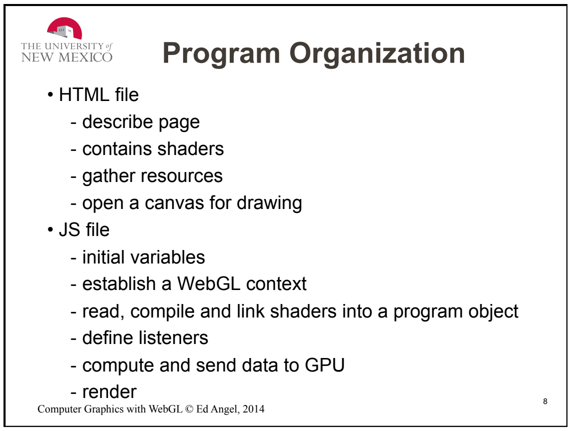
GLSL type

built in variable



GLSL

- OpenGL Shading Language
- C-like with
  - Matrix and vector types (2, 3, 4 dimensional)
  - Overloaded operators
  - C++ like constructors
- Similar to Nvidia's Cg and Microsoft HLSL
- Code sent to shaders as source code
- WebGL functions compile, link and get information to shaders



Program Organization

- HTML file
  - describe page
  - contains shaders
  - gather resources
  - open a canvas for drawing
- JS file
  - initial variables
  - establish a WebGL context
  - read, compile and link shaders into a program object
  - define listeners
  - compute and send data to GPU
  - render