CSI 4341, Computer Graphics

Lecture 3: Shaders

Date: 2012-08-28
Author(s): Christian Marcantel, Caleb Sumpter
Lecturer: Fredrik Niemelä

This lecture showed us the various types of shaders and how they are used.

1 GLSL

Definition 1.1 *Open GL Shader lanugage.*

Definition 1.2 *It is a high-level shading language based on the syntax of the C programming language.*

- It was created for a more direct control of the graphic pipeline.

1.1 Types of Shaders

- There are two shades that we have discussed in class: Vertex Shader and Fragment Shader
  - The Vertex Shader's purpose is to transform each vertex's 3D position in virtual space to a 2D screen.
  - The Fragment Shader or Pixel Shader's purpose is to compute the attributes such as color for each pixel.

2 Vertex Shader

in vec4 vPosition;
void main(void)
$gl\_Position = vPosition$

- gl_Position is already a variable and it is the return
- gl_Position has attributes
  - so you can reference other variables gl_Position= $vec4(vPosition.x, vPosition.y, vPosition.z, vPosition.w$

- if you want to color something, below in vec4 create a out vec4 vColor then assign vColor to a new vec4
- void main (void)

  gl_Position...
  vColor = vec4(1, 0, 0, 1)

3 Fragment Shader

The above code that was given would be overwritten by the gl_FragColor because the Fragment Shader gives the attributes for color

void main (void)
    gl_FragColor = $vec4(1, 0, 0, 1)$

So vColor in the previous example is useless
If you want to change color but not transparency could fill in some of the data
vColor = $vec4(0, 1, 0, 1)$
vColor.rgb = $vec3(1, 0, 0)$