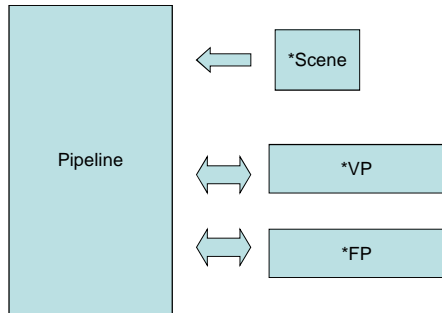


Overview



VertexProcessor

- IN: world position, color, normal, texture coordinates.
- OUT: "Vertex" class: screen position, attributes
- Ignore "triangle" method

FragmentProcessor

- IN: "Fragment" class: screen position (x,y), attributes.
- OUT: Modify "FrameBuffer"

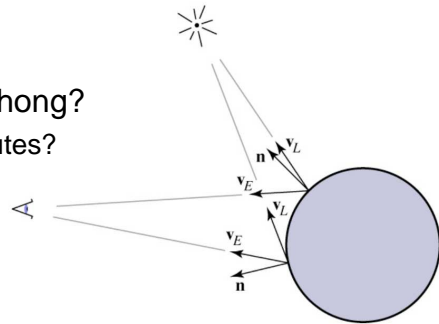
Attributes

- VP: output.attrs[i]
- FP: f.attrs[i]
 - OFF BY ONE!
 - f.attr[0] is depth-value
- VP/FP.nAttr()
- Linearly interpolated
 - Normals?



Local, Eye-space Point Lights

- What assumptions?
 - Eye?
 - Lights?
- How to do Phong?
 - What attributes?



© 2008 Steve Marschner

Local, Eye-space Point Lights

- Pipeline.lights
- PointLight.getIntensity() $\rightarrow i_d$
- i_s = assume white
- k_s = Pipeline.specularColor
- k_d = Vertex color or texture.sample

$$I_p = k_a i_a + \sum_{\text{lights}} (k_d (L \cdot N) i_d + k_s (R \cdot V)^\alpha i_s)$$

Software vs. OpenGL

- Will not match for *FragmentShadedVP
- All else should match
- Demo

* Reflection Mapping



* Reflection Mapping



* Reflection Mapping

- Calculate reflected direction
- Look up pixel in spherical environment map



* Reflection Mapping

- Demo