Shader Overview

CPU

slow; synchronization issues

GPU

VBO: Vertex Buffer Object

uniforms

attributes

position: x, y, z
normal: nx, ny, nz
texture coords: s, t

vertex shader

clipping, rasterization

fragment shader

textures

uniforms

samplers

attributes: associated with vertices

uniforms: global variables that can be changed by user for each rendering call, e.g., each rendered object.

varying: variables to be interpolated across rendered triangles.

Example Vertex Shader

```c
void main() {
    vec4 view_pos = u_ModelViewMatrix * a_Position;
    vec4 proj_pos = u_ProjectionMatrix * view_pos;
    gl_Position = proj_pos;  // final assigned vertex position (in CCS)
    // variable attributes, interpolated across triangle, used by fragment shader
    v_ViewPosition = vec4(view_pos.xyz, 1);  // want interpolated VCS coords
    v_TexCoord = a_TexCoord;  // want interpolated tex coords
    v_ViewNormal = a_Normal;  // want interpolated normal
}
```

Note: should really also be transforming vertex normals before interpolation.
Example Fragment Shader

```glsl
uniform vec4 u_FragColor;
uniform sampler2D u_AlbedoTex;

varying vec4 v_ViewPosition;
varying vec4 v_ViewNormal;
varying vec2 v_TexCoord;

void main() {

  vec4 texColour = texture2D(u_AlbedoTex, v_TexCoord);
  gl_FragColor = texColour;
}
```

Programmable Pipeline

vertex shader

fragment shader