When looking at our pipleline, we see that

$$\text{Projection} \cdot \text{View} \cdot \text{Model} \cdot \text{vPosition}$$

where the first two are camera, and the the last is object specific. A fragment is all the data we may store related to a pixel. The rastorizer converts the normalized device coordinates into actual pixels. The fragment shader runs on every generated rasterized pixel. Another word for rasterization is "scan conversion." Feel free to look up scan conversion algorithms. We will put lighting into the fshader. Next week on the exam will have to write out the matricies for projection, view, model etcetra.

We now turn to the point light source. It serves as a decent model that we may use in our computer generated graphics. We learn diffuse, ambient, and specular reflection. We cover Phong and Blinn-Phong today! For ambient light, we say that

$$I_a = L_a * K_a, 1 \leq K_a \leq 1$$

Where $8*$ is elementwise multiplication, with l_a determined by light and K_a determined by surface. We cover lambertian surfaces and Lambert's law. From this we get the following equation

$$\max(\hat{l} \cdot \hat{n}, 0)/d^2$$

for the ambient light.

For specular light, we consider

$$L_s K_s \max(r \cdot v, 0)^{\alpha}$$

For the Blinn-Phong model, we compute

$$H = \frac{L + V}{||L + V||}$$

And replace $R \cdot V$ with $N \cdot H$.

We will use Phong shading, which uses per-vertex normals. Note that we may use a more accurate shading model to disguise approdimations from triangulation. For example, for the cylinder we worked on last time, we could use the plane normals or the calculated normals from the center of the cylinder. An althernative to Phong shading is Gouraud shading, which performs an interplated average from flat shading.

Next class, we will do review, and get on the same page. Rebuild project by then?