Phong Shading
-The Journey - The Experience-
By: Johnathan O'Malia

Phong shading is fun. It does stuff. Fancy stuff. To start, let's look at the Vertex Shader.
The vertex shader is pretty much the prep phase for getting all the numbers necessary for calculating everything.

1. Get the WorldView * Projection matrix
2. Transform the position by the WorldViewProjection matrix
3. Get the position in world space by multiplying the position by the world matrix
4. Get the lighting vector from the unit vectors of the lighting position - world space
5. Set the view vector to world space
6. Set the normal vector to world space

All this goes to make sure I get the variables needed for pixel shading, and so that it's in the right position.

The Pixel Shader

In the beginning there was Ambient Lighting.

I had defined an ambient color externally so I take that.

Then there was Diffuse Lighting.

So I take that location I calculated for Lighting. Dot it with the normal. Then Clamp it between 1 and 0 it. The Dotting is a way to NOT take the cosine of the light vector to normal. Trigonometry does lead you into horror movies.

The viewer looks upon the object and saw specular reflections.

I needed to find the reflections to make the specular. The reflection vector represents the direction of the reflection from the light vector. I get that by taking the Normal * the Diffuse * 2 – the lighting direction and then normalizing it all. So I have the reflections, but now I need that speculat. So I take it, dot it with the NEGATIVE view Directions, clamp it. The take it to the power of 8.

The almighty Phong equation:

I = Ambient + Shadow * (DiffuseAmount * DiffuseColor * Specular)

The names are fairly plain text. I take the ambient lighting and add the Shadow color which is multiplied by Diffuse Component (Amount and Color) and the Specular.

But Wait! There's Textures!

I = Ambient * Texture + Shadow * (Diffuse * Texture * Specular)

Take that ambient and multiply it by the texture. This lets an ambient color lighting be applied to the texture; Great for palette swapping! The shadow also has to be multiplied by the texture or else it's not going to light with the texture in mind.