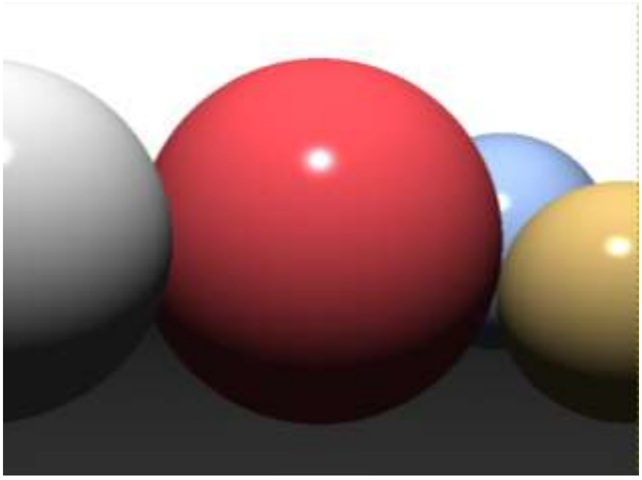
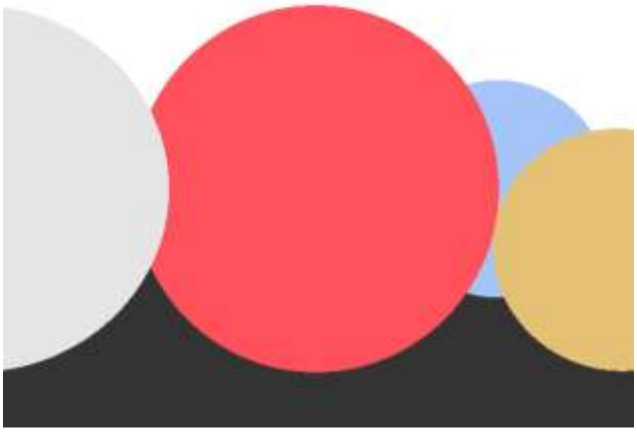
Advanced Graphics Programming

Workshop Two (Part A) – Ray Tracing Spheres (Phong)

This workshop follows on from the Ray Casting algorithm so you need to have completed at least workshop one (part A and B). This workshop involves modelling light to create diffuse and specular lighting as illustrated below:



Before you start this workshop you should already have something similar to:



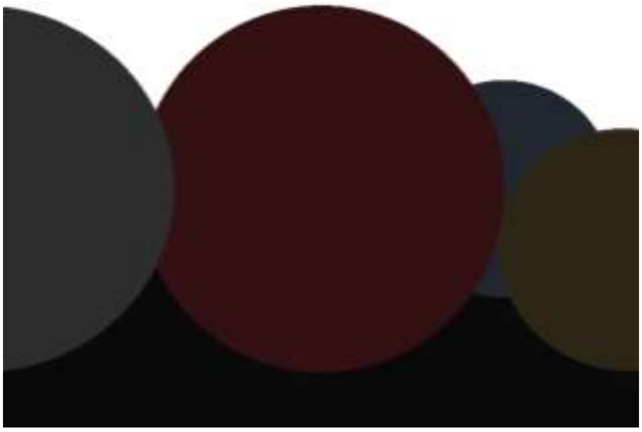
In a previous session the colour of each pixel was set to the surface colour of the nearest intersected sphere. In this session you will light your scene using the Phong shading model. It works on the simple principle that objects have three material properties: ambient, diffuse and specular reflectivity. These properties are assigned colour values, with brighter colours representing a higher amount of reflectivity. Light sources have the same three properties and the final colour is the sum of the lighting and material interactions.

Step 1: Ambient light is the simplest form of light. The equation for the ambient colour of the current pixel $\mathbf{c_a}$ is given below:

$$\mathbf{c_a} = \mathbf{k_a}\,\mathbf{i_a}$$

Where $\mathbf{k_a}$ is the surface's ambient coefficient, or “ambient color,” when in doubt set this to be the same as the surface colour and $\mathbf{i_a}$ is the ambient light intensity e.g. (0.1, 0.1, 0.1).

Experiment with different values of $\mathbf{i_a}$ to create scenes with different ambient light. When you have achieved this step temporally comment out the ambient light so that you can test each of the lighting types individually. This makes each step easier to debug and you can combine all the lights later to get the final result.



To add more interesting material effects such as dull or shiny surfaces you will need to a light source in your scene.

Step 2: Add a point light source to your scene. Your light source just needs a position e.g. (0, 20, 0) and an intensity, $\mathbf{i}$. Light intensity should be neutral in colour with equal red, green and blue intensities e.g. (1.0, 1.0, 1.0). Notice that the example light has been positioned above the scene and is out of the bounds of the image so will not be visible and will have no effect until you complete the next step.

Step 3: Add diffuse lighting. Diffuse lighting makes points on the sphere facing the light source brightly illuminated, whilst points at an angle are darker and points facing away are black. The diffuse equation for the colour of the current pixel $\mathbf{c_d}$ is given below:

$$\mathbf{c_d} = \mathbf{k_d}\,\mathbf{i}\,\max(0, \hat{\mathbf{l}} \cdot \hat{\mathbf{n}})$$

Where $\mathbf{k_d}$ is the diffuse coefficient or the surface colour of the intersected sphere, $\mathbf{i}$ is the intensity of the light source, $\mathbf{l}$ is the light ray from the intersection point $\mathbf{p}$ to your light source with position $\mathbf{s}$.

$$\mathbf{l} = \mathbf{s} - \mathbf{p}$$

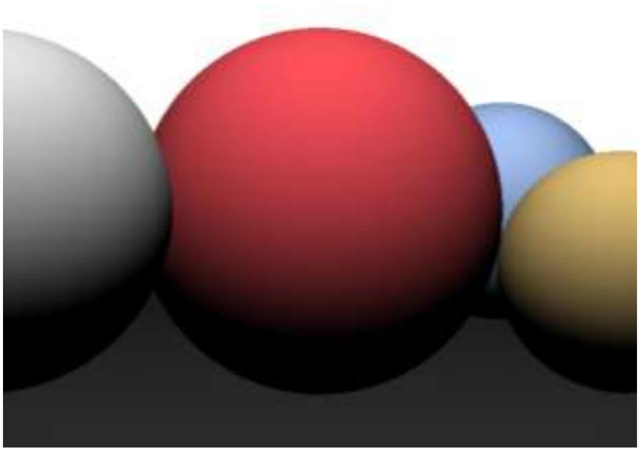
And $\mathbf{n}$ is the normal at the intersection point $\mathbf{p}$ of your sphere with centre $\mathbf{c}$:

$$\mathbf{n} = \mathbf{p} - \mathbf{c}$$

Hints: Don't forget to normalise $\mathbf{l}$ and $\mathbf{n}$ before calculating the diffuse light. Make light sources neutral in colour, with equal red, green and blue intensities.

Note: The dot product will be negative when the surface is pointing away from the light, which is handled by the "max" function.

The results of the diffuse shading are shown below:



As before when you have achieved this step temporally comment out the diffuse light so that you can test each of the lighting types individually.

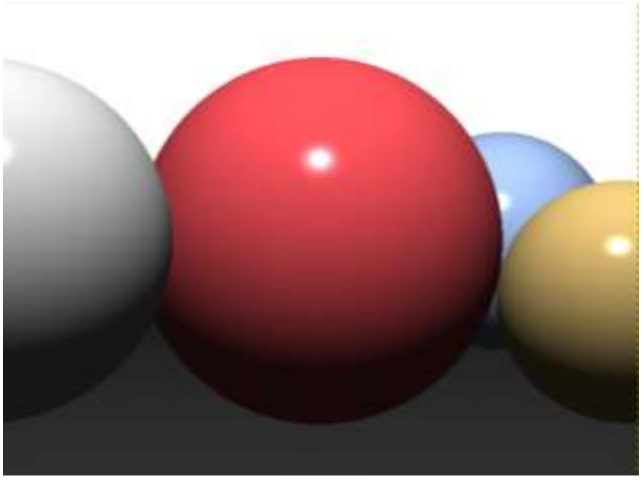
Step 4: Add specular highlights. Specular light depends on the viewer's position and causes a bright spot on the surface it shines on. The specular equation for the colour of the current pixel $\mathbf{c_s}$ is given below:

$$\mathbf{c_s} = \mathbf{k_s}\,\mathbf{i}\max(0,\ \hat{\mathbf{r}}\cdot\hat{\mathbf{v}})^p$$

Where $\mathbf{v}$ is the primary ray direction and $\mathbf{r}$ is the reflection of $\mathbf{l}$ and p is a shininess constant for the sphere. $\mathbf{k_s}$ is the specular coefficient, or the specular colour, of the surface, when in doubt, make the specular colour grey, with equal red, green, and blue values e.g. (0.7,0.7,0.7). The shininess constant can be set for each sphere and should be larger for smoother, mirror like surfaces. In the example below $p = 128$ for all spheres except the ground sphere where $p = 0$.

$$\mathbf{r} = 2\,(\mathbf{l}\cdot\mathbf{n})\,\mathbf{n} - \mathbf{l}$$

Step 5: To put it all together you can simply add the ambient, diffuse and specular intensities together but only add the specular intensity if the diffuse intensity is positive.



Depending on the surface colour of the spheres your colours may become washed out, especially the lighter colours.