

Creating a Raytraced Scene With PovRay

I will be using PovRay in my examples, for it is the most popular and widely used Raytracer. Now we come to using the raytracer. We will be writing code to create scenes that will then be compiled by the PovRay into a TGA image file. One of the most important things you must understand about Raytracing is the 3-D coordinate System. Its Syntax is $\langle X Y Z \rangle$. The X axis moves Horizontally Left and right, the Y axis moves vertically up and down, and the Z axis moves forward to backward 3 dimensionally.

Every RayTraced scene has an origin, which is usually at $\langle 0 0 0 \rangle$, meaning 0 units to the left or right, 0 units up or down, and 0 units forward or backward. So a location of $\langle 2 3 1 \rangle$ would mean 2 units to the right of the origin, 3 units above the origin and 1 unit in front of the origin. And $\langle -2 -3 -1 \rangle$ would mean 2 units to the left of the origin, 3 units below, and 1 unit in behind it.

Ok we have a way to measure our scenes, and a way to find locations. Now we need an Object to be in our scene. The basic geometric shapes (such as the sphere, box, cone, disk, and triangle) and a few not-so-basic shapes (like the torus or donut, polynomial, height field, Bezier patch, and triangular patch) are called primitives. These shapes can be used as building blocks for other more complicated shapes.

Another thing we need in our scene is a view point, or a "Camera". A camera is treated as a special object that both has a location and a place to look at or be pointed at, but can NEVER be seen, because it will be taking the picture. The following example declares a camera in your scene:

```
camera {  
  location <0 1 -2>  
  look_at <0 1 2>  
}
```

This will produce a camera that is located 1 unit above the origin and 2 units behind it. It would be looking at a location 1 unit above the origin and 2 units in front of the origin.

All this is great, but there is one thing missing, light. Without light you cannot see anything, so now we introduce what is known as the light source. The light source is treated as an object, meaning you give it a location. Look at the following example of stating a light source

```
object {  
  light_source { <3 3 -3> color white }  
}
```

This would create a light source that produces White Light, 3 units to the right of the origin, 3 units above the origin, and 3 units in front of the origin of our scene.

Next we will actually be creating scenes. You will need a standard ascii based text editor if you want to actually follow along and create the scenes, EDIT is an editor that comes with dos, and will work fine. Otherwise if you don't want to participate you may just simply continue reading.