Subject: Re: Shadows (Yet Another Object Graphics Question)
Posted by Karl Schultz on Wed, 31 Jan 2001 16:44:19 GMT
View Forum Message <> Reply to Message

"David Fanning" <davidf@dfanning.com> wrote in message
news:MPG.14e0ffd2bbc1eb0b989d4e@news.frii.com...
> Jason P. Meyers (jpm7934@cis.rit.edu) writes:
>
>> Does IDL support casting shadows caused by one object onto another, for
>> example a ball over a ground plane? Here is what I am doing:
>
> Oh, oh. He just got WAY beyond my meager knowledge
> of objects.
>
> Cheers,
>
> David
>
> P.S. Let's just say I wish I were young again. For
> a *lot* of reasons. :-(

The short answer is that there is no magic shadow switch in object graphics.

But here is a reference that might be of help. There is a section on shadows in the OpenGL ("red book") Programming Guide about generating shadows. It involves computing a projection transform that projects your objects onto a plane along the light direction, which is exactly what you want. The book tells how to come up with the matrix given the plane equation of your ground plane and the direction vector of your light source.

You can create an object graphics model structure where you have your objects and their "normal" transform in one model (IDLgrModel). You would also create an additional model whose parent is the first model, and add the same graphic objects (sans ground plane) to it with the /ALIAS keyword, to avoid complete duplication. Finally you adjust the transform in the second model using the transform derived from the red book. The intent is to draw the same objects twice, using a different transform each time.

I haven't actually tried it, but I think it would work and would make an interesting project.

There is one additional problem that you may run into. The shadow polygons will be drawn onto the same plane as the ground plane (we are trying to do this!), so you may get Z-buffer "fights" or stitching effects, because the rasterizer may not generate the same Z-coords for each pixel since the plane equations for the polygons may not be exactly the same. If you encounter this, I would try moving the shadow plane slightly away from the light source so that the shadow polygons sort of float over the ground plane. You would use the original ground plane plane equation when computing the shadow transform matrix, but actually draw the ground plane with a slightly different plane equation. If the ground plane is perpendicular to the light source direction, then it would be a simple matter of adjusting D in the ground plane plane equation.

Hope this helps,
Karl