Subject: Re: Shadows (Yet Another Object Graphics Question)
Posted by davidf on Tue, 30 Jan 2001 23:44:39 GMT
View Forum Message <> Reply to Message

Jason P. Meyers (jpm7934@cis.rit.edu) writes:

> I learned my lesson with the Orb/sphere thing. Before, I try to do
> things the hard way, I'll pose the question and see what surfaces.
>
> 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:
>
> 1) I created two spheres (using the orb object)
> 2) I place them in 3-D space according to some user specified params
> 3) I "tether" them to the origin with a polyline
> 4) I display some axes (centered on [0,0,0])
> 5) I create a white ground plane just under the x & y axes (i.e. z=-0.1)
> 6) I place it all in an IDLexObjView object (I love being lazy!)
> 7) I display it in a draw widget
>
> This all works nicely. However, I would like to see a projection (i.e.
> shadows) of the spheres (and tethers too) in the ground plane.
>
> If I can't do this, my current idea is to draw two more lines and a pair
> of
> circles on the ground plane. I have all the data necessary to do this.
> But
> if something line MyView->TurnOnShadows exists, I would be all for using
> that!
>
> As always, I am open to any suggestions and thank you in advance.

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. :-(

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

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

Subject: Re: Shadows (Yet Another Object Graphics Question)
Posted by Pavel A. Romashkin on Wed, 31 Jan 2001 17:37:23 GMT
View Forum Message <> Reply to Message

My personal opinion is that when Karl posts a reply, the text on the
screen deserves to be in gold (or at least bold red). Thanks Karl!

Pavel

Karl Schultz wrote:
>
> 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

Subject: Re: Shadows (Yet Another Object Graphics Question)
Posted by Jason P. Meyers on Thu, 01 Feb 2001 00:49:05 GMT
View Forum Message <> Reply to Message

Karl Schultz wrote:
>
>
> 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

Wow! This works great! I had already suspected the Z-buffer fight problem and had pre-positioned the ground plane so it wouldn't be an issue. It does help that I have a simple geometry case.

Once again thanks!
--
Jason Meyers
Ph.D. Student, Center for Imaging Science
Rochester Institute of Technology
jpm7934@rit.edu

Subject: Re: Shadows (Yet Another Object Graphics Question)
Posted by davidf on Thu, 01 Feb 2001 00:56:00 GMT
View Forum Message <> Reply to Message

Jason P. Meyers (jpm7934@cis.rit.edu) writes:

> Wow! This works great! I had already suspected the Z-buffer fight
> problem and had pre-positioned the ground plane so it wouldn't be an
> issue. It does help that I have a simple geometry case.

Oh, yeah!? You wouldn't want to post a little code,
would you, so the rest of us could have a look-see. :-)

Cheers,

David

P.S. Let's just say I spent about $200 bucks on Amazon
today, now that I have a couple of pointers to the books
I ought to own. I hope I can understand them better than
I can understand the IDL documentation concerning
these topics. :-(

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Shadows (Yet Another Object Graphics Question)
Posted by Jason P. Meyers on Thu, 01 Feb 2001 19:33:17 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> Jason P. Meyers (jpm7934@cis.rit.edu) writes:
>
>> Wow! This works great! I had already suspected the Z-buffer fight
>> problem and had pre-positioned the ground plane so it wouldn't be an
>> issue. It does help that I have a simple geometry case.
>
> Oh, yeah!? You wouldn't want to post a little code,
> would you, so the rest of us could have a look-see. :-)
>
> Cheers,
>
> David
>
> P.S. Let's just say I spent about $200 bucks on Amazon
> today, now that I have a couple of pointers to the books

Dave: can I have a copy of the pointers so I can de-reference them!
:-)
(I know, bad joke but I couldn't resist!)

> I ought to own. I hope I can understand them better than
> I can understand the IDL documentation concerning
> these topics. :-(
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting
> Phone: 970-221-0438 E-Mail: davidf@dfanning.com
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Toll-Free IDL Book Orders: 1-888-461-0155

I would be glad to post the code. However, I want to fix a recent "problem" with the re-sizable window trick I learned from Dave's book and website. I discovered the problem yesterday in class when I tried to demo my little program to a classmate on our school's sun workstations. I was surprised to see that the draw window got corrupted when I resized the widget because it was working fine on my WindowsNT machine. After some investigations, I discovered the problem was isolated to version 5.3 on the suns (version 5.2 worked fine and I don't know about 5.4 since our department hasn't installed it yet) I was told by our computer folks that they do not consider 5.3 a "supported" version. As it turns out, I suspect the problem is related to the viewport size not being updated automatically when the window size

changes. I even tried explicitly setting the viewport size to [0,0] but it didn't work. My latest idea is to also update the viewport at the same time as the window to see if that works. If it does, I will then make my little demo available on my webpage.

--
Jason Meyers
Ph.D. Student, Center for Imaging Science
Rochester Institute of Technology
jpm7934@rit.edu
