Subject: Re: Object Graphics: multiple Views of same model
Posted by David Fanning on Tue, 18 Dec 2001 14:34:43 GMT
View Forum Message <> Reply to Message

Martin Downing (martin.downing@ntlworld.com) writes:

> Ok, todays object graphics question:
> Say you have a 3D object model and you want to view it in 2 or 3 orthogonal
> directions as you rotate/manipulate it.
> Can this be done with a single object instance? Which we would represent :
>
>   grObject
> |
>   IDLgrModel_Group
> | |
> IDLgrModel_Transform1   IDLgrModel_Transform2
> | |
>   View1   View2
>
> I guess not as we now have the reverse of IDLs graphics Hierarchy, and our
> model_group is not allowed two parents (poor thing!). However it seems to me
> a very reasonable thing to want to do, as a graphics model should be
> viewable from multiple positions. I cant see how Scenes or Viewgroups can
> help, so is the only way to use one view, switching all its settings and the
> top level model transform to look like other views before drawing to the
> other windows?

I didn't mean to insult you the other day, Martin.
I know perfectly well you know what you are doing with
3D graphics, but sometimes I like to overemphasize the
point for our readers. :-)

If I wanted to see two or more views of the same
polygon object, I think I would start by create
two or more polygon objects that all shared the
same data (with the SHARE_DATA keyword). Each
object could go into its own model, each model
into its own view, and the views could be
collected into a scene, that I would display
in my window.

This scheme allows you to manipulate the models
independently to get two or more views of the
same polygon dataset.

I have to admit, I've never had occasion to
use the shared data trick, but it seems to me
the application you describe is exactly why

it is there.

Cheers,

David

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

Subject: Re: Object Graphics: multiple Views of same model
Posted by Martin Downing on Tue, 18 Dec 2001 16:11:02 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> I didn't mean to insult you the other day, Martin.
> I know perfectly well you know what you are doing with
> 3D graphics,
> but sometimes I like to overemphasize the
> point for our readers. :-)

Hi David,
Thats alright, no insult taken!
>
> If I wanted to see two or more views of the same
> polygon object, I think I would start by create
> two or more polygon objects that all shared the
> same data (with the SHARE_DATA keyword). Each
> object could go into its own model, each model
> into its own view, and the views could be
> collected into a scene, that I would display
> in my window.

The key for this application is that we want to set up fixed views in world coordinates and then move the object model around relative to that, ie linked transforms. Your method and explicitly ensuring linked model transforms is the only way to do it that I can think of though. Just tried it for 3 views and it certainly works. I am disappointed though that the way RSI has implemented IDL graphics objects does not allow this type of graphics tree, but there you go!

So currently I have 2 views, both as graphics roots. I cant for the life of me work out what the purpose of Scenes or Viewgroups are!

>
> This scheme allows you to manipulate the models
> independently to get two or more views of the
> same polygon dataset.
>
> I have to admit, I've never had occasion to
> use the shared data trick, but it seems to me
> the application you describe is exactly why
> it is there.
>

I tried

obj2 = OBJ_NEW('IDLgrPolygon', SHARE_DATA = obj1, col=[255,0,0])

but the connectivity and normals were junk, so I have just repeated
generation of the object.

thanks,

Martin

Subject: Re: Object Graphics: multiple Views of same model
Posted by David Fanning on Tue, 18 Dec 2001 16:23:38 GMT
View Forum Message <> Reply to Message

Martin Downing (martin.downing@ntlworld.com) writes:

> I tried
>
> obj2 = OBJ_NEW('IDLgrPolygon', SHARE_DATA = obj1, col=[255,0,0])
>
>
> but the connectivity and normals were junk, so I have just repeated
> generation of the object.

Really!? Do you have an example of this you
could send me? I'm working on some, uh, "notes"
on this topic and this would be interesting.

Cheers,

David

--

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

Subject: Re: Object Graphics: multiple Views of same model
Posted by Dick Jackson on Tue, 18 Dec 2001 16:59:45 GMT
View Forum Message <> Reply to Message

"David Fanning" <david@dfanning.com> wrote in message
news:MPG.168901ed44c97ffe9897b0@news.frii.com...
> Martin Downing (martin.downing@ntlworld.com) writes:
>
>> Ok, todays object graphics question:
>> Say you have a 3D object model and you want to view it in 2 or 3
orthogonal
>> directions as you rotate/manipulate it.
>> Can this be done with a single object instance?
>
> If I wanted to see two or more views of the same
> polygon object, I think I would start by create
> two or more polygon objects that all shared the
> same data (with the SHARE_DATA keyword). Each
> object could go into its own model, each model
> into its own view, and the views could be
> collected into a scene, that I would display
> in my window.
>
> This scheme allows you to manipulate the models
> independently to get two or more views of the
> same polygon dataset.

SHARE_DATA is handy, but as a helpful colleague just reminded me, I think what you may be looking for is to use IDLgrModel::Add, /Alias as follows...

Add graphic objects in as usual for the first view

View 0:
  Model 0:
    obj0
    obj1
    ...

Then, add the *same* objects to the second model using Add, /Alias

View 1:

Model 1:
*obj0*alias*
*obj1*alias*
...

The advantage over SHARE_DATA is that all the object attributes are maintained (color, shading settings, etc.), not just the data (vertices/connectivity). Of course, for the times when you want it, SHARE_DATA is just the ticket.

It might be handy if we could put all our objects in one model and add it to the two views (one real, one as alias), but this is not possible.

Watch out when destroying objects, you can safely destroy the Model 1 (with aliases), which *won't* destroy the contained objects, then destroy Model 0, which *will*.

Hope this helps.

Cheers,
--
-Dick

Dick Jackson / dick@d-jackson.com
D-Jackson Software Consulting / http://www.d-jackson.com
Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392

Subject: Re: Object Graphics: multiple Views of same model
Posted by Rick Towler on Tue, 18 Dec 2001 18:27:49 GMT
View Forum Message <> Reply to Message

I think Dick wins the prize. The alias keyword to IDLgrModel::Add will allow you to do exactly what you want to do.

On a related note, you should try my camera object instead of the IDLgrView object for setting up your view volume and manipulating your model transforms. It will simplify your life.

Create your object (we'll say it is centered at 0,0,0)

Add your object to 3 models, using the alias keyword for 2 of them.

Create 3 instances of the camera object, one with pitch and yaw of 0, one with a pitch of -90, yaw of 0, and one with a yaw of -90 and a pitch of 0 to get your 3 orthogonal views. Set the third_person keyword to something appropriate (say I have an orb with a radius of 1, and I want my "eye" to be

1 unit from the orb I would set third_person=2). Place the cameras at the origin.

Add one model to each camera.

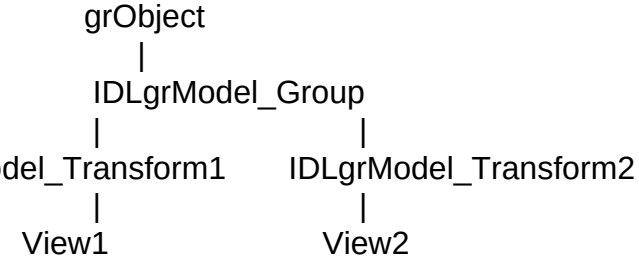
Draw the cameras to your 3 windows.

To manipulate, use the camera::pan method. Since you specify the pan in change in degrees, you simply pan all three cameras the same way and you will keep your orthogonal views.

You can get my camera object here:
http://www.acoustics.washington.edu/~towler

As for viewgroups, the best use I have come up with is the heads up display. Two of the examples on the above web page demonstrate this technique.

-Rick

"Martin Downing" <martin.downing@ntlworld.com> wrote in message
news:CWGT7.26238$4e3.3402029@news6-win.server.ntlworld.com.. .
> Ok, todays object graphics question:
> Say you have a 3D object model and you want to view it in 2 or 3 orthogonal
> directions as you rotate/manipulate it.
> Can this be done with a single object instance? Which we would represent :
>
>


>
> I guess not as we now have the reverse of IDLs graphics Hierarchy, and our
> model_group is not allowed two parents (poor thing!). However it seems to me
> a very reasonable thing to want to do, as a graphics model should be
> viewable from multiple positions. I cant see how Scenes or Viewgroups can
> help, so is the only way to use one view, switching all its settings and the
> top level model transform to look like other views before drawing to the

> other windows?
>
> Martin
>
>
>
>

Subject: Re: Object Graphics: multiple Views of same model
Posted by karl_schultz on Tue, 18 Dec 2001 19:11:41 GMT
View Forum Message <> Reply to Message

David Fanning <david@dfanning.com> wrote in message
news:<MPG.168901ed44c97ffe9897b0@news.frii.com>...
> Martin Downing (martin.downing@ntlworld.com) writes:
>
>> Ok, todays object graphics question:
>> Say you have a 3D object model and you want to view it in 2 or 3 orthogonal
>> directions as you rotate/manipulate it.
>> Can this be done with a single object instance? Which we would represent :
>>
>>   grObject
>> |
>>   IDLgrModel_Group
>> | |
>> IDLgrModel_Transform1   IDLgrModel_Transform2
>> | |
>>   View1   View2
>>
>> I guess not as we now have the reverse of IDLs graphics Hierarchy, and our
>> model_group is not allowed two parents (poor thing!). However it seems to me
>> a very reasonable thing to want to do, as a graphics model should be
>> viewable from multiple positions. I cant see how Scenes or Viewgroups can
>> help, so is the only way to use one view, switching all its settings and the
>> top level model transform to look like other views before drawing to the
>> other windows?
>
> I didn't mean to insult you the other day, Martin.
> I know perfectly well you know what you are doing with
> 3D graphics, but sometimes I like to overemphasize the
> point for our readers. :-)
>
> If I wanted to see two or more views of the same
> polygon object, I think I would start by create
> two or more polygon objects that all shared the
> same data (with the SHARE_DATA keyword). Each
> object could go into its own model, each model

> into its own view, and the views could be
> collected into a scene, that I would display
> in my window.

I think that a better way is to create a single polygon object and then add it to multiple models using the ALIAS keyword on IDLgrModel::Add.

You can also add a *model* to multiple models. So, in this application:

IDLgrModel_Transform1->Add, IDLgrModel_Group
IDLgrModel_Transform2->Add, IDLgrModel_Group, /ALIAS

> This scheme allows you to manipulate the models
> independently to get two or more views of the
> same polygon dataset.

As is the case when using ALIAS.

> I have to admit, I've never had occasion to
> use the shared data trick, but it seems to me
> the application you describe is exactly why
> it is there.

I think that it is not used that often.

SHARE_DATA is intended for when you have a huge set of data that you don't want to duplicate if you use it in multiple graphics objects. The Object Graphics objects all store a copy of the data (e.g., vertex lists) and SHARE_DATA is a way of easing the costs of storing this data.

There are other uses, like "linking" data across multiple objects, so that if you change one, you change them all. Also, it is useful if you want to use the same vertex list for an IDLgrPolygon and IDLgrPolyline.

SHARE_DATA is a way to solve the problem brought up here, but it is just a way to save some storage while making copies of entire objects.

Also, making copies of objects (with or without the storage-saving benefits of SHARE_DATA) would allow you to change attributes like color so that you get a different color in each view, if that is desirable. Using ALIAS is much more efficient, but won't let you change the attributes used in each view, which I think is the desired effect in this situation.

Hope this helps,
Karl

Subject: Re: Object Graphics: multiple Views of same model
Posted by Martin Downing on Tue, 18 Dec 2001 23:26:34 GMT
View Forum Message <> Reply to Message

"David Fanning" <david@dfanning.com> wrote in message
news:MPG.16891b716012fe429897b1@news.frii.com...
> Martin Downing (martin.downing@ntlworld.com) writes:
>
>> I tried
>>
>> obj2 = OBJ_NEW('IDLgrPolygon', SHARE_DATA = obj1, col=[255,0,0])
>>
>>
>> but the connectivity and normals were junk, so I have just repeated
>> generation of the object.
>
> Really!? Do you have an example of this you
> could send me? I'm working on some, uh, "notes"
> on this topic and this would be interesting.
>
> Cheers,
Hi David,

Here is some code you may find useful. Im sure Im just not instantiating the new object correctly when I use
share_data - still I'm sure that will give useful tutorial data !

Martin
```
================================
pro test_object_share_data
; Tests the fact that I do not know how to correctly use the
; object Share_Data creation method
; MRD 18/12/2001
 obj = OG_Extrude( poly = circle3d(rad = .2,n=4), axis = [0,0,1], col =
[255,0,0])
 obj2 = OBJ_NEW('IDLgrPolygon', SHARE_DATA = obj, col=[255,0,0])
 xobjview, obj2
 xobjview, obj
end

function Circle3d, rad=r, n=n
; Creates a disk in the z=0 plane, centre (0,0,0) as a 3d point array
; MRD 18/12/2001
 pa = fltarr(3,n)
```

```
 theta = indgen(n)*!dpi*2/n
 pa[0,*] = r*cos(theta)
 pa[1,*] = r*sin(theta)
 pa[2,*] = 0
 return, pa
end

function OG_Extrude, polygon=poly, axis=axis, color=color, OPEN=OPEN
; Extrudes the given planar polygon along axis to form a closed solid
; (if OPEN keyword set, then the ends are not closed)
; returned as a IDLgrPolygon object
; MRD 18/12/2001
 pa1 = poly
 pa1[0,*] = pa1[0,*] + axis[0]
 pa1[1,*] = pa1[1,*] + axis[1]
 pa1[2,*] = pa1[2,*] + axis[2]
 pa1 = [[poly],[ pa1]]
 n = n_elements(poly[0,*])
 a = indgen(n)
 con = lonarr(2*(n+1)+n*5+1)
 con[0:n] = [n,a]
 aa =  [n,2*n-a-1]
 con[n+1:2*n+1] = aa
 i0 =2*(n+1)
 for i = 0,n-1 do begin
  con[ i0 + i*5:i0+i*5+4]= [4,i, (i+1) mod n,((i+1) mod n)+n, n+i]
 endfor
 con[2*(n+1)+n*5] = -1
 if keyword_set(OPEN) then con = con[2*n+2:*]

 oPoly = OBJ_NEW("IDLgrPolygon", data = pa1, poly = con, color = color)
 return, oPoly
end

 ============================================================ ====
```

Subject: Re: Object Graphics: multiple Views of same model
Posted by Martin Downing on Tue, 18 Dec 2001 23:57:32 GMT
View Forum Message <> Reply to Message

> SHARE_DATA is handy, but as a helpful colleague just reminded me, I think
> what you may be looking for is to use IDLgrModel::Add, /Alias as
follows...
>
.....
>
> The advantage over SHARE_DATA is that all the object attributes are

> maintained (color, shading settings, etc.), not just the data
> (vertices/connectivity). Of course, for the times when you want it,
> SHARE_DATA is just the ticket.
>
> It might be handy if we could put all our objects in one model and add it
to
> the two views (one real, one as alias), but this is not possible.
>
Thanks for the help, Alias is the answer. In fact it seems to me that you
can put all objects into one model, including lights,
 then bind that to as many views as you wish through a final model each.

For those intesested the following code should run showing two viewpoints of
the same rotating object model. (which rotates inside another which holds
the lights for the system. It also includes use of the viewGroup.

cheers for all the help guys

Martin

paste the code below and type:
> demo_object_multiview_alias, Obj, oWindow=oWindow
 == =

```
function Circle3d, rad=r, n=n
; Creates a disk in the z=0 plane, centre (0,0,0) as a 3d point array
; MRD 18/12/2001
 pa = fltarr(3,n)
 theta = indgen(n)*!dpi*2/n
 pa[0,*] = r*cos(theta)
 pa[1,*] = r*sin(theta)
 pa[2,*] = 0
 return, pa
end

function OG_Extrude, polygon=poly, axis=axis, color=color, OPEN=OPEN
; Extrudes the given planar polygon along axis to form a closed solid
; (if OPEN keyword set, then the ends are not closed)
; returned as a IDLgrPolygon object
; MRD 18/12/2001
 pa1 = poly
 pa1[0,*] = pa1[0,*] + axis[0]
 pa1[1,*] = pa1[1,*] + axis[1]
 pa1[2,*] = pa1[2,*] + axis[2]
 pa1 = [[poly],[ pa1]]
 n = n_elements(poly[0,*])
 a = indgen(n)
 con = lonarr(2*(n+1)+n*5+1)
```

```
 con[0:n] = [n,a]
 aa =  [n,2*n-a-1]
 con[n+1:2*n+1] = aa
 i0 =2*(n+1)
 for i = 0,n-1 do begin
  con[ i0 + i*5:i0+i*5+4]= [4,i, (i+1) mod n,((i+1) mod n)+n, n+i]
 endfor
 con[2*(n+1)+n*5] = -1
 if keyword_set(OPEN) then con = con[2*n+2:*]

 oPoly = OBJ_NEW("IDLgrPolygon", data = pa1, poly = con, color = color)
 return, oPoly
end

pro demo_object_multiview_alias, Obj, oWindow=oWindow, scale = scale,
offset=offset
; Demonstrates use of Alias keyword to allow two views
; (here used within a Viewgroup) to share an object  model tree

 if obj_valid(obj) eq 0 then begin
  obj = OG_Extrude( poly = circle3d(rad = .1,n=11), axis = [0,0,0.5], col =
[255,0,0] )
 endif

 pos1 = [0,0]
 vdim =  [512,512]
 pos2 = [vdim[0],0]
 windim = vdim*[2,1]
 if n_elements(offset) eq 0 then  offset = 30   ; camera offset degrees

 ; build model
 oModel = OBJ_NEW('IDLgrModel', NAME = "Model")
 oModel->add, obj
 oGroup = OBJ_NEW('IDLgrModel', NAME = "Group")
 oGroup->add, oModel

 ; orient data
 oModel->Rotate, [1,0,0], -90
 if n_elements(scale) gt 0 then oModel->Scale, scale,scale,scale

 ; lighting
 oLight = OBJ_NEW('IDLgrLight', TYPE=0, INTENSITY=0.3)
 oGroup->Add, oLight
 oLight = OBJ_NEW('IDLgrLight', LOCATION=[2,2,4], TYPE=1, intensity = 0.5)
 oGroup->Add, oLight

 ; View1
 oView1 = OBJ_NEW('IDLgrView', PROJECTION=2, COLOR=[0,0,0], dim = vdim, loc
```

```
= pos1)
 oTop1 = OBJ_NEW('IDLgrModel', NAME = "TOP")
 oTop1->add, oGroup
 oView1->Add, oTop1
 ; ROTATE View 1
 oTop1->Rotate, [0,1,0], offset

 ; View2 : NOTE USE OF ALIAS
 oView2 = OBJ_NEW('IDLgrView', PROJECTION=2,  COLOR=[0,0,0],dim = vdim, loc
= pos2)
 oTop2 = OBJ_NEW('IDLgrModel', NAME = "TOP")
 oTop2->add, oGroup, /Alias
 oView2->Add, oTop2
 ; ROTATE View 2
 oTop2->Rotate, [0,1,0], -offset

 ; Create View Group
 oViewGroup = OBJ_NEW('IDLgrViewGroup')
 oViewGroup->add, oView1
 oViewGroup->add, oView2

 oWindow = OBJ_NEW('IDLgrWindow', quality = 1, dim = windim, location=pos0,
graphics = oViewGroup);,color_model = 1 )
 ; or for display free use:
 ;oWindow = OBJ_NEW('IDLgrBuffer', quality = 1, dim = windim, graphics =
oViewGroup);,

 oWindow->Draw
; t = systime(1)
 for i = 1,100 do begin
   oModel->Rotate, [1,1,0], 2     ; axis and angle to rotate by
   oWindow->Draw
 ; for reading rendering back to an image uncomment these lines
 ;  oImage = oWindow->Read()
 ;  oImage->GetProperty, data=imtc
 ; obj_destroy, oImage
 ;    tv, imtc[0,*,*]
 endfor
 ;print, "time elapsed = ", systime(1)-t

end

==========================================================

--
---------------------------------------
Martin Downing,
Clinical Research Physicist,
```

Grampian Orthopaedic RSA Research Centre,
Woodend Hospital, Aberdeen, AB15 6LS.
Tel. 01224 556055 / 07903901612
Fax. 01224 556662

m.downing@abdn.ac.uk

"Dick Jackson" <dick@d-jackson.com> wrote in message
news:5CKT7.446$Lx1.2599@shaw-ty1...
> "David Fanning" <david@dfanning.com> wrote in message
> news:MPG.168901ed44c97ffe9897b0@news.frii.com...
>> Martin Downing (martin.downing@ntlworld.com) writes:
>>
>>> Ok, todays object graphics question:
>>> Say you have a 3D object model and you want to view it in 2 or 3
> orthogonal
>>> directions as you rotate/manipulate it.
>>> Can this be done with a single object instance?
>>
>> If I wanted to see two or more views of the same
>> polygon object, I think I would start by create
>> two or more polygon objects that all shared the
>> same data (with the SHARE_DATA keyword). Each
>> object could go into its own model, each model
>> into its own view, and the views could be
>> collected into a scene, that I would display
>> in my window.
>>
>> This scheme allows you to manipulate the models
>> independently to get two or more views of the
>> same polygon dataset.
>
> SHARE_DATA is handy, but as a helpful colleague just reminded me, I think
> what you may be looking for is to use IDLgrModel::Add, /Alias as
follows...
>
> Add graphic objects in as usual for the first view
>
> View 0:
> Model 0:
> obj0
> obj1
> ...
>
> Then, add the *same* objects to the second model using Add, /Alias
>
> View 1:
> Model 1:

> *obj0*alias*
> *obj1*alias*
> ...
>
> The advantage over SHARE_DATA is that all the object attributes are
> maintained (color, shading settings, etc.), not just the data
> (vertices/connectivity). Of course, for the times when you want it,
> SHARE_DATA is just the ticket.
>
> It might be handy if we could put all our objects in one model and add it
to
> the two views (one real, one as alias), but this is not possible.
>
> Watch out when destroying objects, you can safely destroy the Model 1
(with
> aliases), which *won't* destroy the contained objects, then destroy Model
0,
> which *will*.
>
> Hope this helps.
>
> Cheers,
> --
> -Dick
>
> Dick Jackson / dick@d-jackson.com
> D-Jackson Software Consulting / http://www.d-jackson.com
> Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392
>
>

Subject: Re: Object Graphics: multiple Views of same model
Posted by Martin Downing on Wed, 19 Dec 2001 09:09:41 GMT
View Forum Message <> Reply to Message

"Rick Towler" <rtowler@u.washington.edu> wrote in message
news:9vo1tb$1b8i$1@nntp6.u.washington.edu...
> I think Dick wins the prize. The alias keyword to IDLgrModel::Add will
> allow you to do exactly what you want to do.
>
> On a related note, you should try my camera object instead of the
IDLgrView
> object for setting up your view volume and manipulating your model
> transforms. It will simplify your life.
>
>
Thanks Rick,

I've posted code (in reply to Dick) which shows the first stage at least of what I am trying to do, and yes it is related to a stereo head up display. Key thing about it is that you can set up a model frame, then a world frame in which you might add lights, then alias this world frame to the views, when you rotate the single model frame all the views see the same change from their viewpoint.

I will look at your stuff later - cant get through now.

Martin

Subject: Re: Object Graphics: multiple Views of same model
Posted by Rick Towler on Wed, 19 Dec 2001 18:03:06 GMT
View Forum Message <> Reply to Message

I'm sorry, but I go the link wrong.

try this: http://www.acoustics.washington.edu/~towler/

I forgot the trailing slash....

-Rick

"Martin Downing" <martin.downing@ntlworld.com> wrote in message news:cQYT7.37564$0A4.2785570@news11-gui.server.ntli.net...
> "Rick Towler" <rtowler@u.washington.edu> wrote in message
> news:9vo1tb$1b8i$1@nntp6.u.washington.edu...
>> I think Dick wins the prize. The alias keyword to IDLgrModel::Add will
>> allow you to do exactly what you want to do.
>>
>> On a related note, you should try my camera object instead of the
> IDLgrView
>> object for setting up your view volume and manipulating your model
>> transforms. It will simplify your life.
>>
>>
> Thanks Rick,
>
> I've posted code (in reply to Dick) which shows the first stage at least
of
> what I am trying to do, and yes it is related to a stereo head up display.
> Key thing about it is that you can set up a model frame, then a world
frame
> in which you might add lights, then alias this world frame to the views,

> when you rotate the single model frame all the views see the same change
> from their viewpoint.
>
> I will look at your stuff later - cant get through now.
>
> Martin
>
>
>
>

Subject: Re: Object Graphics: multiple Views of same model
Posted by Martin Downing on Wed, 19 Dec 2001 22:47:23 GMT
View Forum Message <> Reply to Message

"Karl Schultz" <karl_schultz@yahoo.com> wrote in message
news:e415b359.0112181111.50e94e0b@posting.google.com...
> David Fanning <david@dfanning.com> wrote in message
news:<MPG.168901ed44c97ffe9897b0@news.frii.com>...
>> Martin Downing (martin.downing@ntlworld.com) writes:
>>
> I think that a better way is to create a single polygon object and
> then add it to multiple models using the ALIAS keyword on
> IDLgrModel::Add.
>
> You can also add a *model* to multiple models. So, in this
> application:
>
> IDLgrModel_Transform1->Add, IDLgrModel_Group
> IDLgrModel_Transform2->Add, IDLgrModel_Group, /ALIAS
>

I agree, this is what I was saying and demonstrating in the code I posted
in my replies yesterday

thanks

Martin
