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
>
>