Subject: Re: Displaying three images simultaneously (using Object Graphics)
Posted by Rick Towler on Tue, 08 Nov 2005 18:38:08 GMT
View Forum Message <> Reply to Message

Victor wrote:
> I have 3 images (im1,im2,im3) which I need to display on XObjView (or
> any alternate object graphics viewer). When I tried to add more than 1
> image to the same graphics object as follows:
>
>> myModel = Obj_New('IDLgrModel')
>> myModel->Add, im1
>> myModel->Add, im2
>
>
> IDL complains:
> "IDLGRMODEL::ADD: Objects can only have one parent at a time:
> <ObjHeapVar9869(IDLGRCOLORBAR)>"

You have something else going on here. You *can* add multiple images to an instance of IDLgrModel. The error is complaining about a colorbar object. In your code below you add 3 images to a single model.

> These obviously plot the three images into 3 separate XObjView windows.
> I want them to be displayed in the same XObjView window. So I try to
> add them to a container and display the it in XObjView as follows:
> myContainer = OBJ_NEW('IDL_Container')
> myContainer->Add,myModel_1, POSITION=0.1
> myContainer->Add,myModel_2, POSITION=0.5
> myContainer->Add,myModel_3, POSITION=0.9
> XOBJVIEW,myContainer,TITLE='All 3 images'
>
> IDL complains (This time a nastier one):
> "XOBJVIEW: IDLGRMODEL::ADD: The ALIAS keyword is only allowed for
> IDLgrComponent class objects."
>
> Looks like XOBJVIEW doesn't even work with IDL_Container or IDLgrView
> objects either (I tried that)
>
> Also,
>
>> myModel = Obj_New('IDLgrModel')
>> myModel->Add, [myModel_1, myModel_2, myModel_3], POSITION=[10,20,30]*B
>> XOBJVIEW,myModel,TITLE='All 3 images'
>
> Works, but doesn't give the expected results (just overlays the three
> images, Position keyword seems to have no effect whatso ever)

No, XOBJVIEW only works with IDLgrModel objects and subclasses. You are also misunderstanding the POSITION keyword of IDL_Container and IDLgrModel. First, it is a long index value that specifies the "place in line" of the object you are adding. Think of a stack. The position is the location in the stack, starting at 0, from the top down. POSITION has nothing to do with locating the object in world space.

> I know the following might be useful, but I am not sure how to
> incorporate them in the above logic (which I'm not even sure is the
> best approach)
> 1) "!P.MULTI"
> 2) "Position"

!P.MULTI is a direct graphics system variable and has no meaning in object graphics.

As I mentioned above, the POSITION keyword isn't what you are looking for. As some Antonio suggested, check out the LOCATION keyword of IDLgrImage instead.

Another approach would be to transform your image data such that your images are lined up as you desire. For example, say I have 2 images that are 100x100 pixels. I first add each of them to a model:

```
mod1=OBJ_NEW('IDLgrModel')
mod1->add, img1

mod2=OBJ_NEW('IDLgrModel')
mod2->add, img2
```

Then I move the second image to the right (positive X direction) by using the TRANSLATE method:

```
mod2->translate, 100,0,0
```

Now viewing the images in XOBJVIEW I have 2 images side by side:

```
XOBJVIEW, [mod1,mod2]
```

Don't get too crazy with this though. For reasons we don't need to get into here, rotating IDLgrImage objects that have been translated doesn't always yield the desired result. Remember view->full reset in XOBJVIEW.

-Rick