
Subject: Re: objects in loop
Posted by [shearerm](#) on Wed, 31 Jan 2001 13:15:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Firstly apologies for the multiple posting of same question.

Secondly I have tried to solve my problem using container objects so I now have.

```
Instruments = OBJ_NEW('IDL_Container')
for p=0,a do begin
  ip = OBJ_NEW('orb',pos=[x(p), y(p), z(p)])
  model -> ADD, ip
  Instruments -> Add, ip
endfor
```

When I have finished with the objects I destroy the "instruments" container object. Is this a correct way of doing things? It does seem to work.

Sent via Deja.com
<http://www.deja.com/>

Subject: Re: objects in loop
Posted by [davidf](#) on Wed, 31 Jan 2001 13:51:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Murray Shearer (shearerm@bp.com) writes:

> The problem is this bit of code is part of a program for creating
> animations of data. When I want to change the instrument location I
> destroy the model and recreate from scratch, which also works (for the
> first couple of animations at least). However after a few loops IDL
> turns into a memory gobbling monster, windows has a whine about running
> out of memory and promptly collapses into a gibbering heap (this may be
> because I am running 5.3 in Win2000?).

This sounds suspiciously like a a problem I ran into last week, which turned out to be a hardware rendering issue. By switching to software rendering, the problem completely disappeared. In any case, turning software rendering on is the FIRST thing I check when I have ANY kind of problem with object graphics. It solves about 80% of my problems. :-)

I think you solved your other problem correctly by

using a container object.

Cheers,

David

--

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: objects in loop

Posted by [Richard Adams](#) on Wed, 31 Jan 2001 17:10:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Murray, I have done similar things to plot hundreds of cells moving around in a 3D field. I use the following - convoluted - method. I derived this some time ago, so it is perhaps not the most elegant solution..but it works on my Macs. I define just once a set of orbs for my models and put these into column 1 of an N by 2 object array. In column 2 I put symbols with each object as their shape. That way each object could have its own colour and shape. (If I change colour over time I change the orb colour.) For each frame I make a polyline plot using these symbols, take an image and paste it into a movie frame. That way you are not moving lots of objects about and you create them just once. When all is done, the orbs get deleted.

Hope this helps.

Richard.

--

Richard J Adams }<}}}}i¿½>

e: r.j.adams@bath.ac.uk

MRC Senior Research Fellow

t: +44 1225 826436

Developmental Biology Programme

f: +44 1225 826779

Department of Biology and Biochemistry

University of Bath

Bath, BA2 7AY

UK

Subject: Re: objects in loop

Posted by [Pavel A. Romashkin](#) on Wed, 31 Jan 2001 17:55:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

I have not quite figured out why do you have to *destroy* the objects. Why not place them into an OBJARR, save it and use SetProperty, pos=*** to simply move the objects to new positions, then re-draw the view? Destroying works, too, but by moving them instead you'd save destruction-creation overhead. Besides, ORB is a subclass of IDLgrMODEL, so it has a ton of useful parameters that you can alter instead of re-creating.

Cheers,
Pavel

Murray Shearer wrote:

```
>
> I want to insert a bunch of spheres to represent instrument locations
> into my model. At the moment I have something like:
>
> for p=0,a do begin
>   ip = OBJ_NEW('orb',pos=[x(p), y(p), z(p)])
>   model -> ADD, ip
> endfor
>
> Where x, y and z are instrument coords.
>
> This did work fine (although I guess it is not technically correct) I
> could see all my instruments.
>
> The problem is this bit of code is part of a program for creating
> animations of data. When I want to change the instrument location I
> destroy the model and recreate from scratch, which also works (for the
> first couple of animations at least). However after a few loops IDL
> turns into a memory gobbling monster, windows has a whine about running
> out of memory and promptly collapses into a gibbering heap (this may be
> because I am running 5.3 in Win2000?).
>
> I think the solution might be to reuse the model after destroying only
> the spheres that represent instrument location (since that is all I
> really want to change). However because of the way I created the
> spheres I can't destroy them.
>
> My question is therefore how do I create my spheres so that they are
> all one object or automatically generate them with individual names?
>
> Ps. I used to use mesh_obj to create the spheres but thanks to a recent
> posting I now use orb, which is much neater!
>
> Sent via Deja.com
> http://www.deja.com/
```

Subject: Re: objects in loop

Posted by [Mark Hadfield](#) on Wed, 31 Jan 2001 20:33:52 GMT

[View Forum Message](#) <> [Reply to Message](#)

"Richard Adams" <r.j.adams@bath.ac.uk> wrote in message
news:B69DF7F8.173EE%r.j.adams@bath.ac.uk...

> Hi Murray, I have done similar things to plot hundreds of cells moving
> around in a 3D field. I use the following - convoluted - method. I derived
> this some time ago, so it is perhaps not the most elegant solution..but it
> works on my Macs. I define just once a set of orbs for my models and put
> these into column 1 of an N by 2 object array. In column 2 I put symbols
> with each object as their shape. That way each object could have its own
> colour and shape. (If I change colour over time I change the orb colour.)
> For each frame I make a polyline plot using these symbols, take an image
and
> paste it into a movie frame. That way you are not moving lots of objects
> about and you create them just once. When all is done, the orbs get
deleted.

That's not quite how I would do it, though maybe I don't understand the
motivation for your approach.

I would describe the cloud of cells (or whatever) as a single IDLgrPolyline
object attached to one or more IDLgrSymbol objects. The polyline stores the
position information for the N cells in its DATA property as a [2,N] or
[3,N] array. It probably has LINESTYLE=6 to suppress the lines between the
vertices, and it stores the shapes in its SYMBOL property. You can use as
many symbols as you want--they are cyclically repeated as necessary. Here is
the relevant IDL documentation

SYMBOL (Get, Set)

Set this keyword to a vector containing one or more instances of the
IDLgrSymbol object class to indicate the plotting symbols to be used at each
vertex of the polyline. If there are more vertices than elements in SYMBOL,
the elements of the SYMBOL vector are cyclically repeated. By default, no
symbols are drawn. To remove symbols from a polyline, set SYMBOL to a
scalar.

Bundling an arbitrary shape into an IDLgrSymbol is straightforward.

You can animate by changing the polyline's vertex data, or by creating a
series of polyline (one per animation frame) all referring to the same
symbol object(s).

Mark Hadfield
m.hadfield@niwa.cri.nz <http://katipo.niwa.cri.nz/~hadfield/>
National Institute for Water and Atmospheric Research

PO Box 14-901, Wellington, New Zealand
