
Subject: Re: Delayed plotting in object graphics
Posted by [Mark Hadfield](#) on Wed, 13 Dec 2000 01:52:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Ben Tupper" <pemaquidriver@tidewater.net> wrote in message
news:3A36BBD9.1A4F589C@tidewater.net...

> Hello,

>

> I am converting some 2d direct graphics to object graphics.
> Past users of the program producing the graphics have wanted
> to plot the points in 'slow motion'. That was easy to brute
> force a lag using PLOTS and WAIT in a loop. Is there a way
> of doing this with object graphics?

Hmmm. Interesting. Some ideas:

1. Naive and ugly: Add the points to the scene one at a time; each time a point is added, redraw the scene and then wait. But once the scene has been constructed, every redraw will be instantaneous (well, as instantaneous as OG ever is).

2. Sophisticated and ugly: Add the points to the scene all at once, but with each point in a separate model. Make these models instances of a purpose-built subclass of IDLgrModel, say SlowModel. In SlowModel's class structure add a tag called delay, to be set to a sensible value in the Init method. Define SlowModel draw as:

```
pro SlowModel::Draw, oSrcDest, oView
  self->IDLgrModel::Draw, oSrcDest, oView
  wait, self.delay
end
```

Then every redraw will be slowed down. (But you might be surprised how often OG scenes get redrawn, especially when printing them!)

3. Best I can think of: Achieve a similar effect by manipulating colour. Put all the points in the same atom (much more efficient!) and set the atom's COLOR property so that one point is in a conspicuous colour (red, black) and the others in a less conspicuous one (grey). Each time you want to highlight a different point, get, rotate and reset the COLOR property vector. Calling WAIT in a loop is a pretty ugly way of driving a process like this--with a bit of extra effort you can get a better result using widget timer events.

In general I find animation of one sort or another a very effective tool for helping visualise data. I have been experimenting with different ways of animating direct & object graphics (mainly the latter). You might want to look at some of my attempts at <http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/>. In particular the

MGHgrAnimatorBase class implements basic animator functionality with user controls and the MGHgrAnimator class is a general-purpose object graphics animator application. But I never thought of trying the sort of animation you describe and I don't have anything that would implement it directly.

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

Subject: Re: Delayed plotting in object graphics
Posted by [Ben Tupper](#) on Thu, 14 Dec 2000 02:24:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mark Hadfield wrote:

```
> "Ben Tupper" <pemaquidriver@tidewater.net> wrote in message
> news:3A36BBD9.1A4F589C@tidewater.net...
>> Hello,
>>
>> I am converting some 2d direct graphics to object graphics.
>> Past users of the program producing the graphics have wanted
>> to plot the points in 'slow motion'. That was easy to brute
>> force a lag using PLOTS and WAIT in a loop. Is there a way
>> of doing this with object graphics?
>
> Hmmm. Interesting. Some ideas:
>
> 1. Naive and ugly: Add the points to the scene one at a time; each time a
> point is added, redraw the scene and then wait. But once the scene has been
> constructed, every redraw will be instantaneous (well, as instantaneous as
> OG ever is).
>
> 2. Sophisticated and ugly: Add the points to the scene all at once, but with
> each point in a separate model. Make these models instances of a
> purpose-built subclass of IDLgrModel, say SlowModel. In SlowModel's class
> structure add a tag called delay, to be set to a sensible value in the Init
> method. Define SlowModel draw as:
>
> pro SlowModel::Draw, oSrcDest, oView
>   self->IDLgrModel::Draw, oSrcDest, oView
>   wait, self.delay
> end
>
> Then every redraw will be slowed down. (But you might be surprised how often
> OG scenes get redrawn, especially when printing them!)
```

>
> 3. Best I can think of: Achieve a similar effect by manipulating colour. Put
> all the points in the same atom (much more efficient!) and set the atom's
> COLOR property so that one point is in a conspicuous colour (red, black) and
> the others in a less conspicuous one (grey). Each time you want to highlight
> a different point, get, rotate and reset the COLOR property vector. Calling
> WAIT in a loop is a pretty ugly way of driving a process like this--with a
> bit of extra effort you can get a better result using widget timer events.
>
> In general I find animation of one sort or another a very effective tool for
> helping visualise data. I have been experimenting with different ways of
> animating direct & object graphics (mainly the latter). You might want to
> look at some of my attempts at
> <http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/>. In particular the
> MGHgrAnimatorBase class implements basic animator functionality with user
> controls and the MGHgrAnimator class is a general-purpose object graphics
> animator application. But I never thought of trying the sort of animation
> you describe and I don't have anything that would implement it directly.
>

Thanks Mark,

I think I shall try idea #3, making the points that are 'yet-to-be-plotted' the background color while the 'already-plotted' points a color distinct from the background. I suppose I could link the rate of the plot to a slide widget, which, sliding from left to right, indicates the relative position in the sequence of points. Then the user could move back and forth through the sequence at will (maybe.)

I will also peek at your source page.

Ben

--

Ben Tupper
248 Lower Round Pond Road
POB 106
Bristol, ME 04539

Tel: (207) 563-1048
Email: PemaquidRiver@tidewater.net

Subject: Re: Delayed plotting in object graphics
Posted by [Mark Hadfield](#) on Thu, 14 Dec 2000 03:15:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Ben Tupper" <pemaquidriver@tidewater.net> wrote in message
news:3A382F64.878031CD@tidewater.net...

>

> I think I shall try idea #3, making the points that are
'yet-to-be-plotted' the
> background color while the 'already-plotted' points
> a color distinct from the background. I suppose I could link the rate of
the
> plot to a slide widget, which, sliding from left to right, indicates the
> relative position in the sequence of points. Then the user could move
back and
> forth through the sequence at will (maybe.)

I think allowing user control is a very good thing. Watching an animation
unfold is one thing, but being able to interact with it is much better.

> I will also peek at your source page.

I think it would take me a couple of hours to hack my MGHgrAnimator class
into something that does what you want. (I might just have a go, because I
think it would be pretty cool. On the other hand, I might do some work...)
As I said before, the animation functionality is in the superclass,
MGHgrAnimatorBase. To apply it to a different situation one just has to
override MGHgrAnimatorBase's methods (CountFrames, Display, Play, PlayNext,
PlayNext, etc) for the particular situation.

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

Subject: Re: Delayed plotting in object graphics
Posted by [Mark Hadfield](#) on Thu, 14 Dec 2000 22:18:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Mark Hadfield" <m.hadfield@niwa.cri.nz> wrote in message
news:976672336.180471@clam-ext...

> "Ben Tupper" <pemaquidriver@tidewater.net> wrote in message
> news:3A36BBD9.1A4F589C@tidewater.net...

>> Hello,

>>

>> I am converting some 2d direct graphics to object graphics.
>> Past users of the program producing the graphics have wanted
>> to plot the points in 'slow motion'. That was easy to brute
>> force a lag using PLOTS and WAIT in a loop. Is there a way
>> of doing this with object graphics?

>

> Hmmm. Interesting. Some ideas:

One more idea: I have added a CUMULATIVE property to my MGHgrAnimator class. Setting this property to an integer n greater than 1 causes n-1 frames to be displayed in addition to the current one. Setting it to a negative value causes *all* frames up to the current one to be displayed. Thus this allows interactively controlled, progressive display of a cloud of points. The value of the CUMULATIVE property can be changed at any time.

See <http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/>

Performance is limited by the speed with which the scene can be drawn. With each data point in a separate atom like this, drawing speed becomes very slow when the number of points displayed at each time exceeds a few hundred.

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
