Subject: cgWindow Update

Posted by David Fanning on Thu, 01 Sep 2011 20:15:05 GMT

View Forum Message <> Reply to Message

Folks,

Because of the way the resizeable graphics window,
cgWindow, stores graphical "commands" in the window to
play over again when the window is resized, it is difficult
to get keyword and parameter values to have one value on
the display and another value when the "commands" are
rendered in the PostScript device. In other words,
sometimes these values need to be evaluated at run-time.
(Any keyword or parameter than uses TexToIDL or Greek,
for example, runs into this problem.)

I came up with a solution and wrote about it last month.
There were a couple of problems with that solution. One,
it used an EXECUTE to evaluate the keyword, which made
it impossible to use this program on a Virtual Machine.
And, two, the article I wrote was incomprehensible. (I
read it a dozen times yesterday, and I *still* couldn't
figure out what it meant!)

In truth, the writing wasn't so bad, but the "solution"
was WAY too complicated for anyone (including me, I guess)
to figure out. It any case, it wasn't working for me!

So, I spent 2-3 hours yesterday afternoon thinking about it,
and went to bed with no good ideas, but suddenly woke up
about 2AM with a different (and MUCH simpler!) solution.
So, I have thrown out the old solution and implemented
this new one. I have to say, it works MUCH better than
the old one, and in particular, it works with all these
gnarly test cases I have been accumulating. I'm really
quite happy with it.

Basically, it allows you to specify keywords and
positional parameters that can have different values
from the values you use with the command itself when
you add it to cgWindow. These "alternative" keyword
and parameter values are used instead of the display
values, when the commands are "executed" in the PostScript
device.

I've re-written the article that describes all this,
and you can find it here:

http://www.idlcoyote.com/cg_tips/kwexpressions.php

In addition to these changes, I have found it necessary
to be able to set cgWindow up to that it can specify decomposed
color in the PostScript device. (All Coyote Graphics
routines work in decomposed color mode, although they can
be dragged kicking and screaming into indexed color mode
if the user insists. But, things are just so EASY in
decomposed color mode, that I really prefer to work there.)

So, you can either allow all cgWindows to set the PostScript
device to decomposed color (you will need IDL 7.1 and higher
to do this), by typing a command like this:

    cgWindow_Set_Defs, Decomposed=1

Or, you can do it for a single window on the display with cgControl:

    cgWindow
    cgControl, Decomposed=1

All this to say that probably 7-8 programs changed overnight, so it
might be a good idea to upgrade your Coyote Library, if you need
this kind of functionality. :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")