Subject: Re: 8-bit vs. 24-bit color on Windows
Posted by davidf on Wed, 27 Jan 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Oh, heck. I'm a sucker for a little newsgroup controversy
every now and again....

William Thompson (thompson@orpheus.nascom.nasa.gov ) writes:

> This seems to be a strictly widget-oriented solution. Not everything is
> widgets!!!! Nor should it be--a lot of specialized data analysis is not done
> in a widget environment. Quite a bit of it, in fact, is done directly from the
> command line. Most scientific users who are writing programs their own use
> don't bother to go to the trouble of writing widget programs.

While I'm dubious about "most scientific users" working at the command
line, I'll let that pass. But color table manipulation is NOT
strictly a widget-oriented solution. It is simply easier to
implement with widgets. (As are most user-interactive type of
programs. After all, that is the *point* of a widget program.)

XCOLORS, for example, works equally well with any object
that has a "DRAW" method. Give XCOLORS a structure that has
an object reference, the name of the "DRAW" method to call,
and the window you want to draw into (even a regular old
graphics window) and without doing much else you have a window
that can update itself when the colors in the color table change.

But if Bill's right, most scientific programmers aren't
using XCOLORS or XLOADCT anyway. They are building their
own color table vectors and loading them with TVLCT the
way God intended. What help can we offer to them?

I don't honestly know. Perhaps RSI *could* hack a PSUEDOCOLOR
like thing together for the Windows platform. But it sure as
hell would be non-standard Windows programming and a gigantic
pain in the neck to maintain, I suspect.

More to the point, would I want them spending their resources
doing this, so that we can continue to work with the old
tired 8-bit strategy far into the future, or do I want them
adding new features to the language that will allow us to
write better programs for better and faster computers now?
Personally, I vote for better programs going forward. And
which strategy do you think is more likely to result in
RSI even being around in the future?

> One thing I don't understand is why the graphic needs to be regenerated.

> Couldn't one just read in the byte values from all active windows and write
> them back out again?

Yes. If the byte value represented the color of the pixel,
which it simply does not in a 24-bit true-color system
(*any* 24-bit true-color system).

> One thing that seems to be happening with IDL lately is that people are being
> expected to use more complicated programming techniques to solve problems that
> used to be much simpler.

I don't think this problem is specific to IDL. Have you
looked at any new word processing software that has come to
market in the last five years? What about just typing one
character after the other until you get to the end of the
line and hit a carriage return? Microsoft Word is so damn
complicated I haven't even worked up the courage to figure
it out. Squiggly red lines everywhere!

> The problem of displaying traditional pseudo-color
> images on the more advanced graphics cards is one such example (although only
> on Windows platforms). Another is the changes within modal widgets, where one
> is asked to use complicated things like timer events for things that just used
> to work without thinking about it.

I hate to tell you this, Bill. But the problem here is
that RSI is hacking operating system code to make it
*look* like another operating system, all in the name
of cross-platform compatibility. If I recall correctly,
that is the solution you are proposing here for yet one
more operating system difference. I, for one, could do without
any more of these kinds of "solutions". :-)

Cheers,

David

---