Subject: Object Images
Posted by J.D. Smith on Fri, 25 Jul 1997 07:00:00 GMT
View Forum Message <> Reply to Message

I have several question for those object-graphics experts among you. I am trying to create an object version of a image drawing widget I made long ago. I am running into several difficulties.

1. The object image is of unacceptable quality. In order to allow free changes in dimensions, some sort of dithering interpolation is done, which I have yet to figure out how to control. The manual quotes: "The image will be resampled as necessary to fit withing this rectangle." Not much help. Is it possible to fix the aspect ratio of pixels in an image, and do a simple kind of rebinning (Sample=1) in which pixels are just scaled up, rather than new interpolated pixels created? I notice the same problem, for instance, in David Fanning's ximage test procedure.

2. DEVICE,SET_GRAPHICS=sg does not work in the usual way, since everything is drawn at once when the Draw method is invoked. I suppose a way around this would be to put whatver I wanted drawn with a different graphics mode into it's own view, set it before invoking the window's draw method on this view, and unsetting it after. Is this correct, or will Graphics Modes not function at all now?

3. Am I interpreting the LOCATION and DIMENSION keywords correctly by assuming that they control these features in a heirarchical sense in this order: image -> view -> window? That is, does changing the *view* LOCATION and DIMENSION change these properties of the image, or merely translate the image withing the view accordingly?

Anyway, I may revert to direct graphics to get the job done, since I've already gotten it working that way before.

Also, perhaps of interest to some of you trying to integrate widgets and objects is a prototype class I have been working on, ObjWidget. This superclass defines a standard interface by which objects belonging to this class or it's subclasses can communicate with eachother, sending what I've termed as 'messages'. These messages may be little more than the widget events generated within ObjWidget objects, but they need not be limited to that, and can be general 'object messages'. This allows the object communication network to be of arbitrary complexity, not just 'up-the-widget-tree' event network of pure widgets.

One ultimate idea utilizing this superclass is of an ObjWidget derived widget_draw object with the capacity to have other ObjWidget modules 'plugged-in' to it very simply. These modules would be easy to write and activate, deactivate, etc. Examples might be a module which performs statistics on some subset of the displayed image (as defined by the mouse) or a color stretching tool (with logarithmic, etc. color stretching capabilities). They should be easy to develop and maintain, since they don't have to worry about the details of message exchange (they simply sign up for the messages they'd like to get, from the list available). I should have a working prototype soon, (more soon if I abandon object graphics). Anyway, just avoiding real work.

JD