
Subject: Re: Interfacing the functions

Posted by [Mark Hadfield](#) on Mon, 08 Apr 2002 02:09:08 GMT

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

"Akhila" <idlfreak@yahoo.com> wrote in message
news:b1ad7b05.0204071626.691b0575@posting.google.com...

> ...I have individual program that performs the following:
> a. Zoom in and Out
> b. Pan
> c. Contrast adjustments
> d. Brings the next slice
>
> Mouse controls all these....Please tell me how I can link all these
> functions so that I can perform these on an image.

You need to modify your event-handling code so that the action it takes depends on the mouse button that was pressed (the "press" tag in the event structure for mouse events with type 0) or released (the "release" tag in the event structure for mouse events with type 1).

I have puzzled over how to make mouse-event-handling code reasonably simple & flexible. I settled on the idea of "mouse-handler" objects. Each graphics-window object has (up to) 3 such handlers; code in the window's event-handler sends mouse events to the appropriate handler; each mouse handler is responsible for remembering its state and calling the window's methods as necessary. The window object normally displays droplist widgets by which the user can select the handler for left and middle buttons; the right button is normally associated with a context-menu handler. This approach is implemented in the following files

http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/mgh_mouse_handler_library.pro

http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/mgh_window__define.pro

You're welcome to look at this code but I warn you that it might take some time & effort to understand what it's doing and why I wrote it that way.

--

Mark Hadfield

m.hadfield@niwa.co.nz

Ka puwaha et tai nei

<http://katipo.niwa.co.nz/~hadfield> Hoea tatou

National Institute for Water and Atmospheric Research (NIWA)
