Subject: Palettes: suggestions on how to do them
Posted by Anonymous on Thu, 26 Sep 1985 14:43:01 GMT
View Forum Message <> Reply to Message

Originally posted by: schley&#64mmm.UUCP (Steve Schley)

Article-I.D.: mmm.194
Posted: Thu Sep 26 10:43:01 1985
Date-Received: Mon, 30-Sep-85 01:06:55 EDT
Distribution: net
Organization: 3M Company, St. Paul, Minn.
Lines: 49

A few months back, I solicited suggestions on how to put palettes (like MacPaint has) in my application. I stated that I would summarize the responses to the net.

Well, better late than never. Here they are, briefly and without attribution:

--> Use the Dialog Mgr only to draw the dialog. Put everything in controls, and use PtInRect to locate where MouseDown occurred. Highlight the controls where appropriate.

--> Use a normal window instead of a dialog. Otherwise, same as first suggestion.

--> Instead of using SelectWindow on the dialog to get events, use BringToFront. This won't deselect the main window. (HiLiteWindow should work, too.) Might have to do a BringToFront on the main window afterwards, though.

--> Dialog and Control Mgrs are overkill. Draw palette with QuickDraw, calculate MouseDown relative to palette buttons, and do highlighting using QuickDraw. In other words, do all the work yourself.

--> Put the palette inside the main window, as Helix and MacDraw do.

I solved the problem as the last response did suggested. This has other plusses, in that multiple windows will each have their own control palette. You must, however, ensure that your user cannot draw on the palette.

I did not try any of the other suggestions. The third one, in particular, doesn't seem like it would do the trick. I am passing these suggestions along without any verification of correctness. If anyone finds out that these do or do not work, post to the net or at

least send me a note. Furthermore, if anyone has more suggestions, please come forward. I don't think this issue has been entirely cleared up.

Finally, I've seen that PageMaker from Aldus uses a very interesting mini-palette. It takes the form of a small, movable window (they call it the "toolbox") that is always active and always in front of all other windows! It has eight control areas in it, and it's very handy to be able to "float" it around the page -- it's always handy, and never in the way of the document. How do they do it?

--
Steve Schley

ihnp4!mmm!schley

Subject: Re: Palettes: suggestions on how to do them
Posted by Anonymous on Tue, 08 Oct 1985 20:10:52 GMT
View Forum Message <> Reply to Message

Originally posted by: robdye&#64ut-sally.UUCP (Rob Dye)

Article-I.D.: ut-sally.3128
Posted: Tue Oct 8 16:10:52 1985
Date-Received: Fri, 11-Oct-85 07:27:32 EDT
References:
Reply-To: robdye@sally.UUCP (Rob Dye)
Distribution: net
Organization: U. Texas CS Dept., Austin, Texas
Lines: 61

In article schley@mmm.UUCP (Steve Schley) writes:
> A few months back, I solicited suggestions on how to put palettes (like
> MacPaint has) in my application. I stated that I would summarize the
> responses to the net.
...
> --> Dialog and Control Mgrs are overkill. Draw palette with QuickDraw,
> calculate MouseDown relative to palette buttons, and do highlighting
> using QuickDraw. In other words, do all the work yourself.
...
> Finally, I've seen that PageMaker from Aldus uses a very interesting
> mini-palette. It takes the form of a small, movable window (they call
> it the "toolbox") that is always active and always in front of all
> other windows! It has eight control areas in it, and it's very handy

> to be able to "float" it around the page -- it's always handy, and
> never in the way of the document. How do they do it?
> Steve Schley
> ihnp4!mmm!schley

Well, I haven't seen PageMaker, but I just recently hacked together something which sounds similar. My palette is a small (four tools the size of MacPaint's tools, arranged vertically), drag-able (it has a cute little drag bar at the top) pseudo-window which is always active and always in front (this turns out to be a little disconcerting under some circumstances, as I will describe later).

I went about it by "doing all the work myself", i.e., I used QD and calculated mouseDowns to see if they were in the palette. The trick was in using a relatively undocumented toolbox global called grayRgn (look in the WindowMgr chapter of IM under InitWindows, I think). This is the desktop region with rounded corners minus the menu bar. The WindowMgr will draw windows clipped to this region (as well as all other appropriate regions), which is what you would expect since it always avoids drawing on top of the menu bar.

When showing the palette (showing/hiding are controlled from a menu item), I made use of this fact by subtracting the region containing my palette from the grayRgn, drawing the palette in the WMgrPort, then calling CalcVisBehind(FrontWindow(), paletteRgn) (I think that's the calling convention) to tell the WindowMgr to recalculate the visRgns of all windows intersecting with paletteRgn. These visRgns will have my paletteRgn subtracted from them, preventing the window from drawing on my palette even if the palette is on top of the FrontWindow.

Note that since the palette is not a window in the WindowMgr sense, I never *really* have more than one active window.

The disconcerting aspect of this approach is that the palette literally shows through everything! When it shows up on top of the OK button of a dialog box, it definitely looks like a bug, not a feature! DAs are no exception either. And since the code for dragging the palette is encountered in the main loop of the program (and NOT in the event loop for the modal dialog), you can't drag it out of the way! Fortunately, clicks on the palette at this point "pass through" the palette and are received by the dialog box, but this is obviously not the way things should work. I'm still trying to find the right way to handle this problem. Suggestions are welcome.

Right now I'm considering using another relatively undocumented toolbox global called ghostWindow, and actually implementing the palette as a WindowMgr window. So far this has been nothing but a pain in the ass.

So it goes...

Subject: Re: Palettes: suggestions on how to do them
Posted by lsr on Thu, 17 Oct 1985 01:02:50 GMT
View Forum Message <> Reply to Message

Article-I.D.: apple.2106
Posted: Wed Oct 16 21:02:50 1985
Date-Received: Fri, 18-Oct-85 01:15:42 EDT
References:
Reply-To: lsr@apple.UUCP (Larry Rosenstein)
Distribution: net
Organization: Advanced Development Group, Apple Computer
Lines: 52

There are a number of ways to do palettes according to what you like best; there is no "right" way. There are 2 issues to think about. How to implement the palette and where to implement it.

I agree that using the Dialog Manager is overkill. Palettes are generally very regular in structure, so it is only a matter of divising coordinates to get the palette item a user clicks in. In addition, you would have to go to a fair amount of work to get the Dialog Manager to display all the little symbols you wanted.

As far as showing your palettes, there are several possibilities:

in the window (as in MacDraw)
globally (as in MacPaint)
in graphical menus (used in MacDraw)

Each has advantages and disadvantages. One advantage of menus is that they are available regardless of the position of the document windows on the screen. (If MacPaint had resizable windows, then you might want to make the window the full screen size sometimes, which would cover the palettes.)

In PageMaker, the Toolbox window is a regular window with title bar, that is in front of the document window. The document window's title bar is also highlighted. It appears that the program makes the Toolbox window the "ghost window", because clicks in the document window are processed as it that window was frontmost.

If you store a windowPtr in the ghost window low memory location, then the FrontWindow call will never report that window as the

frontmost, even if it is.

The idea of modifying the grayRgn variable to affect each window's visRgn, as described by robdye@sally.UUCP (Rob Dye), is very clever. There is not good answer to the problem he described (his palette shows up on top of DAs and Dialogs!).

One way would be to change the visRgn of just the windows belonging to his application. This would have to be done when the palette or a document window is opened/closed/moved. Making the palette a real window and setting it up as the ghost window might help in this respect.

--
Larry Rosenstein
Apple Computer

UUCP: {voder, nsc, ios, mtxinu, dual}!apple!lsr
CSNET: lsr@Apple.CSNET

Subject: Re: Re: Palettes: suggestions on how to do them
Posted by bart on Fri, 18 Oct 1985 16:36:32 GMT
View Forum Message <> Reply to Message

Article-I.D.: reed.2019
Posted: Fri Oct 18 12:36:32 1985
Date-Received: Sun, 20-Oct-85 05:11:14 EDT
References:
Distribution: net
Organization: Reed College, Portland, Oregon
Lines: 44

> There are a number of ways to do palettes according to what you like
> best; there is no "right" way. There are 2 issues to think about.
> How to implement the palette and where to implement it.
>
> I agree that using the Dialog Manager is overkill. Palettes are
> generally very regular in structure, so it is only a matter of
> divising coordinates to get the palette item a user clicks in. In
> addition, you would have to go to a fair amount of work to get the
> Dialog Manager to display all the little symbols you wanted.

>
> As far as showing your palettes, there are several possibilities:
>
> in the window (as in MacDraw)
> globally (as in MacPaint)
> in graphical menus (used in MacDraw)
>
> Each has advantages and disadvantages. One advantage of menus is that
> they are available regardless of the position of the document windows
> on the screen. (If MacPaint had resizable windows, then you might
> want to make the window the full screen size sometimes, which would
> cover the palettes.)

So how is it any more work to write a CDEF and put up a bunch of USERITEM buttons in a modeless dialog that it is to write an MDEF to put up the whole palette? In other words, write a CDEF for something that acts like a button with a picture on it. Then in the DITL for your modeless dialog, just pack them side by side to make your palette. Either neither or both are overkill, surely. If you use a modeless dialog, you get something that acts like a "real artist's" palette -- it's always around, can be moved to where you need it, covered by things, etc. Probably a "Show Palette" menu item and a close box would also be needed. There are certainly times when I would prefer this to MacDraw's "palette menu" approach...

Note that you could almost get by without a CDEF for a palette in a modeless dialog, through some kind of gross hack involving superposing PICTITEMS or ICONITEMS and BUTTONITEMS with no text legend... You could even simply put the pictures next to the buttons, if you were in a hurry -- this would take up a little more space, but would be perfectly legible, and certainly within the bounds of the user interface guidelines as I read them...

Bart Massey
..tektronix!reed!bart
