Subject: Re: Ctrl-C keyboard event missing
Posted by JD Smith on Thu, 20 Mar 2003 18:46:18 GMT

View Forum Message <> Reply to Message

On Thu, 20 Mar 2003 05:47:05 -0700, Mark Servetus wrote:

> It looks like I'll stick with the WIDGET_DRAW keyboard events, since I
> would like the speed and platform independence. I'm using object
> graphics with just one draw widget. The niceness of a Ctrl-c to which I
> respond by placing the scene into the clipboard is hard to give up.
>
>> Key statement from the manual: "The method by which a widget receives
>> the keyboard focus is dependent on the window manager in use."
>
> I do find it hard to believe that of all the keyboard combinations, the
> window manager is withholding the Ctrl-c (if it is the window manager
> and not some other IDL glitch that is blocking it).
>
> Of course, it could be a bug on my platform, Win XP. Hmm... And maybe
> it's just when running from IDLDE. Any more information out there?
>
> Mark
>
> JD Smith <jdsmith@as.arizona.edu> wrote in message
> news:<pan.2003.03.18.17.14.18.437273.31090@as.arizona.edu>...
>> Had KEYBOARD_EVENTS been written as a WIDGET_BASE functionality, such
>> that you could catch events no matter where in the application they
>> occur (except in input forms), it would have made a big difference. As
>> it exists on WIDGET_DRAW, you still need to set focus explicitly to the
>> draw widget, if you hope to catch key events while "outside" your draw
>> widget. Key statement from the manual: "The method by which a widget
>> receives the keyboard focus is dependent on the window manager in use."
>>
>> The nice thing about the WIDGET_DRAW key-event support is it's
>> presumably platform independent, and it can dish out keyboard events at
>> a much higher rate (e.g. for holding down the arrow key). Also, in
>> principle, you could hide a very small WIDGET_DRAW behind other
>> elements of a drawless widget-app, in the same way you hide a
>> widget_text with the original hack.
>>

I've discovered UNIX versions have a keyboard event bug of their own,
which I've reported to RSI -- you can't give a widget_draw text focus
using:

  widget_control,draw_id,/INPUT_FOCUS

This makes it all but useless on Unix systems (tested with Linux,

MacOSX, and Solaris), since you have to click or use tab before you can catch keyboard events. This also means the hidden WIDGET_DRAW method I mentioned won't work on these systems, since you can't give it focus.

Another anomaly: Control plus a character generates a non-printable-ascii character sequence. E.g., Control-c gives: 3b (== ^C), rather than 99b (== c), with modifiers set to 2. So at least we can get Control-c, but we have to add 96b to it find out which character it was, which may or may not be platform independent. Alt plus a character behaves as I'd expect, setting the appropriate modifier and leaving the character as is, which seems more reasonable. I suppose Shift + character gives some justification to changing both ev.ch and ev.modifiers, but it stays within the printable ASCII set.

JD