
Subject: Keyboard events

Posted by [dsreyn](#) on Fri, 28 Jul 2000 07:00:00 GMT

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

I'm in the process of writing a new application, and several of the likely users have requested that I allow various keyboard "shortcuts", as an alternative to clicking on various controls with the mouse. For example, we will need to define a region in an image that will be zoomed in a separate draw window, and it would be nice to be able to use arrow keys to position and resize the zoom region.

I've never tried to do this before in IDL (previously I've been content to handle mouse events), so I went looking through the manuals to figure out how to do it. Unfortunately, the event handling routines seem to be limited to mouse and focus events. The closest thing I could find was the `get_kbrd` function; someone here suggested putting a call to `get_kbrd` in the event handling function. However, it would seem that this would cause keyboard events to queue up, unprocessed, until the user wiggled the mouse. In other words, it doesn't seem to be quite what I need.

Is there something in the documentation that I have overlooked? Any ideas would be most appreciated.

Doug

Subject: Re: Keyboard events

Posted by [John-David T. Smith](#) on Thu, 03 Aug 2000 07:00:00 GMT

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

By the way, are people aware that you can trap control keys with the hack (either version). At least for me, Control+a-z map to keys 1B-26B. C-p for print anyone? Haven't tried this one on anything but Unix. Just in case people couldn't find the post I've reattached the latest hack with this modification. Just run `tt` and try various keys, including control+letter and arrows. If people want to experiment along these lines on other platforms/os's, I'd be willing to collect the results and bundle the routine into something generically useable and distributeable. Maybe then RSI would take notice and work to include a less hackish alternative.

JD

--

J.D. Smith /*\ WORK: (607) 255-6263
Cornell University Dept. of Astronomy */ (607) 255-5842
304 Space Sciences Bldg. /*\ FAX: (607) 255-5875
Ithaca, NY 14853 */
pro catch_text,ev

```

type=tag_names(ev,/STRUCTURE_NAME)
erase
case type of
  'WIDGET_TEXT_SEL': begin
    widget_control, ev.id, get_uvalue=ulrdo
    widget_control, ev.id,SET_TEXT_SELECT=ulrdo[4],SENSITIVE=0
    case ev.offset of
      ulrdo[0]: xyouts,.5,.5,/NORMAL,'Up', CHARSIZE=2.,ALIGNMENT=.5
      ulrdo[1]: xyouts,.5,.5,/NORMAL,'Left', CHARSIZE=2.,ALIGNMENT=.5
      ulrdo[2]: xyouts,.5,.5,/NORMAL,'Right',CHARSIZE=2.,ALIGNMENT=.5
      ulrdo[3]: xyouts,.5,.5,/NORMAL,'Down', CHARSIZE=2.,ALIGNMENT=.5
    else:
    endcase
    widget_control, ev.id,/SENSITIVE,/INPUT_FOCUS

  end
  'WIDGET_TEXT_CH': begin
    if ev.ch le 26 then $
      xyouts,.5,.5,/NORMAL,'C-'+strtrim(ev.ch+96b,2),CHARSIZE=2., $
      ALIGNMENT=.5 $
    else $
      xyouts,.5,.5,/NORMAL,string(ev.ch),CHARSIZE=2.,ALIGNMENT=.5

    end
  endcase

end

end

pro tt_event,ev
  type=tag_names(ev,/STRUCTURE_NAME)
  widget_control, ev.top,GET_UVALUE=info
  if type eq 'WIDGET_TRACKING' then begin
    widget_control, info.key,/INPUT_FOCUS
    return
  endif

  if ev.type le 1 then widget_control, info.key,/INPUT_FOCUS
  str=string(format='(2(I2.2,;,":"))',ev.X,ev.Y)
  case ev.type of
    0:str='P'+str
    1:str='R'+str
    2:str='M'+str
    else:str='*'
  endcase
  erase
  xyouts,.5,.5,/NORMAL,str, CHARSIZE=1.,ALIGNMENT=.5
end

```

```
pro tt
  base=widget_base()
  draw=widget_draw(base,XSIZE=60,YSIZE=60,/TRACKING_EVENTS,/MO TION_EVENTS, $
    /BUTTON_EVENTS)
  case !VERSION.OS_FAMILY of
    'unix': ulrdo=[1,3,5,7,4]
    'Windows': ulrdo=[1,4,6,9,5]
    else: message,'Not supported.'
  endcase
  key=widget_text(base,/ALL_EVENTS,FRAME=0,XSIZE=2,YSIZE=3, $
    VALUE=['**', '**', '**'], EVENT_PRO='catch_text',UVALUE=ulrdo)
  widget_control, base,SET_UVALUE={draw:draw,key:key},/REALIZE
  widget_control, key,SET_TEXT_SELECT=ulrdo[4],/INPUT_FOCUS
  XManager,'tt',base,/NO_BLOCK
end
```

File Attachments

1) [tt.pro](#), downloaded 80 times

Subject: Re: Keyboard events

Posted by [John-David T. Smith](#) on Thu, 03 Aug 2000 07:00:00 GMT

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

Doug Reynolds wrote:

>
> I'm in the process of writing a new application, and several of the likely
> users have requested that I allow various keyboard "shortcuts", as an
> alternative to clicking on various controls with the mouse. For example,
> we will need to define a region in an image that will be zoomed in a
> separate draw window, and it would be nice to be able to use arrow keys
> to position and resize the zoom region.
>
> I've never tried to do this before in IDL (previously I've been content
> to handle mouse events), so I went looking through the manuals to figure
> out how to do it. Unfortunately, the event handling routines seem to be
> limited to mouse and focus events. The closest thing I could find was the
> get_kbrd function; someone here suggested putting a call to get_kbrd in
> the event handling function. However, it would seem that this would cause
> keyboard events to queue up, unprocessed, until the user wiggled the mouse.
> In other words, it doesn't seem to be quite what I need.
>
> Is there something in the documentation that I have overlooked? Any ideas
> would be most appreciated.
>
> Doug

I think I recently posted a follow-on entitled "More keyboard events" to my hack

of many years ago which allows cursor key widget events. It increases the hackery to some degree and has cross-platform issues, but sometimes, you just need them.

An example program is included in the post. My original hack is widely used in a variety of programs, including popular ones such as atv, so it's stability is probably pretty good. The new hack has undergone limited testing, so beware. I've been using it with success under Unix, but your mileage may vary.

Good luck,

JD

--

J.D. Smith /*\ WORK: (607) 255-6263
Cornell University Dept. of Astronomy */ (607) 255-5842
304 Space Sciences Bldg. /*\ FAX: (607) 255-5875
Ithaca, NY 14853 */
