Subject: More Keyboard events
Posted by John-David T. Smith on Tue, 23 May 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Many of you may be using my keyboard event hack David put on his sight few years ago (http://www.dfanning.com/tips/keyboard_events.html). Recently, I really wanted to be able to use the arrow keys, in addition to just normal "enterable" ascii keys. I've been living with compromises like "ijkm" for up left right down, but I thought there *must* be some way to accomplish it. There is. But beware, it's even more hackery than the previous hack.

The example code is attached. As before, it uses a text widget hidden in a bulletin-board base (nor /ROW or /COLUMN), but this time, in addition to processing text insertion events, it also examines selection events. A carefully chosen amount of data is set into the widget, and the cursor is positioned in a special position. Hitting a cursor key generates a selection event which advertises the change in the insertion point. Immediately returning the insertion point to it's original location will allow a repeatable offset to be mapped to the correct arrow key.

Unfortunately, IDL's implementation of WIDGET_TEXT_SEL events is spotty at best on some platforms. Under UNIX, it works just fine, missing no events and never getting confused. Under Windows, a bug exists in which consecutively pushing the same arrow key does not generate an event every other time. Also, the offsets into the text are different from unix to windows, likely due to the cr/lf vs lf end of line markers. The latter can easily be dealt with by changing the offset mappings for different platforms as I've done, but the former poses something of a problem. In any case, the example code is attached. If you're just going to use it just for unix, you can dispense with the !VERSION.OS_FAMILY nonsense. Windows will ignore the 3rd event after two consecutive presses in a single direction, since it will be some unknown offset. That makes repeatedly pressing a single arrow key frustrating.

Remember all the caveats on the page above... notice how I've made certain to set INPUT_FOCUS when there is a chance it was taken away. Also beware that for Windows you must SET_TEXT_SELECT after the widget is REALIZE'd (see below).

To guard against the code run when an arrow is hit taking a long time to compute, and thus permitting events with unwanted and unknown selection offsets, I desensitize the text widget during calculation. This will cause it to miss events when a calculation is blocking (no input possible). E.g. if pressing the left arrow rescales some large image. If you want to get more events through during these conditions and can tolerate occassional skips due to funky offsets, remove the SENSITIVE pair.

Forward working offset mapping for other platforms, and I'll include them too.

Good luck,

JD

--
J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
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': $
     xyouts,.5,.5,/NORMAL,string(ev.ch),CHARSIZE=2.,ALIGNMENT=.5
  endcase

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 83 times