Subject: widget_programming
Posted by R.Bauer on Wed, 02 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Hi,

position keywords are fine if I use always the same platform for my
application.
But unfortunately we have some computers and some platforms where
positional widgets
are shown very bad.

The main reason is the different system font and the different screen
resolutions.

Which fonts are on unix idl and nt idl will give the same results? Is
there a list avaliable?

Did the fontsize append on the screen resolution and how will I get
independent to the screen resolution?

I do understand well thats if I defined 1280 * 1024 and I have on an
other one only 1024 x 768 I got in trouble.
But if I define 200 * 160 both should look simliar. What happens is
thats on the best system (it's always the one where the widget was
builded) all is ok and on an other one all is shifted on top of each
other. Input fields could not be reached.

R.Bauer

Subject: Re: widget_programming
Posted by mgs on Fri, 04 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

In article <3757BF62.5494763D@astro.estec.esa.nl>, Michael Werger
<mwerger@astro.estec.esa.nl> wrote:

> Dear all
>
> David and Richard wrote about cross-platform development of GUIs
> and the problems with this
>
> [... discussion simply cut ...]
>
> I simply gave up (was trying on Solaris and Windows95/98) - I
> got them by default pretty similar but never equal...
>

> I was doing some cross-platform development for Tcl/Tk too and
> had similar problems. There I developed simply a routine which
> reads a resource file where you can specify font sizes etc.
> I think this way might be a good way here too...? Even different
> users on the same platform have different preferences for
> colors and font sizes

I've been working sporadically on a way to handle this. I hard-coded something up a couple years ago called OsInfo that is not very useful. I've been working on a version that creates sliders, text widgets, etc. in the background, then stores the sizes in a structure. You can use the sizes as part of a ButtonSize calculation. Here's an idea, where mMisc.mOsInfo.mBuffer.button contains the number of pixels that a button takes up without text in it:

```
   ;   the widget button group definition
   wBGAAC = CW_BGroup2(wBase, asResText, /Frame, /Row, $
      ButtonSize=mGeo.XSize / N_Elements(asResText) - $
      mMisc.mOsInfo.mBuffer.button, Font=mMisc.labelFont)
```

The idea was to use the XSize of a pre-existing base in the widget hierarchy to determine the ButtonSize of a BGroup. At the time, I was developing on a Mac with an SGI target. It worked, but it's not flexible enough. The OsInfo code on <http://www.ivsoftware.com/IV_Code.html> has the old incarnation of OsInfo if someone wants to use it for a starting point.

--
Mike Schienle   Interactive Visuals, Inc.
mgs@ivsoftware.com   Remote Sensing and Image Processing
http://www.ivsoftware.com/   Analysis and Application Development

---

Subject: Re: widget_programming
Posted by rivers on Fri, 04 Jun 1999 07:00:00 GMT

View Forum Message <> Reply to Message

In article <MPG.11beddbd77f6ed409897cb@news.frii.com>, davidf@dfanning.com
> (David Fanning) writes:
>
> Perhaps this article will generate some other ideas
> we can use. I'll collect them and post them on my
> web page. My main problem is that I don't have all these
> machines, so when a problem develops I solve it and
> forget to write the solution down. :-(
>
> (I'm also vaguely remembering a Mark Rivers function
> that helped get nice fonts in a machine-independent

> fashion. Mark, are you reading. Haven't heard from
> you in a while.)

I'm still here!

Here is the function I use to get useable and reasonably similar fonts on Motif, (e.g. Unix and VMS) and Windows NT machines. You use the function as in the following examples:
font = get_font_name(/helvetica, /large, /bold)
font = get_font_name(/courier, size=3)

Sorry I haven't even written a standard documentation header!

Using this function I am able to write widget applications which look very similar on Windows and Motif. I have used it a lot on Sun, Digital Unix, VMS, and Windows NT, less so on SGI, HP and Linux.

```
 **********************************************************
function get_font_name, $
        helvetica=helvetica, times=times, courier=courier, $
        tiny=tiny, small=small, medium=medium, large=large, huge=huge, $
        size=size, $
        bold=bold, italic=italic, $
        dpi75=dpi75, dpi100=dpi100

; Returns the name of the font with the specified characteristics

; This routine should know more about the Mac.  Right now it only works well on
; Windows and Motif

if (!version.os_family eq 'Windows') then begin
    font = ''
    if keyword_set(helvetica) then font = font + 'Helvetica' else $
    if keyword_set(times)     then font = font + 'Times' else $
    if keyword_set(courier)   then font = font + 'Courier New' else $
    font = font + 'MS San Serif'

    if keyword_set(bold) then font = font + '*Bold'
    if keyword_set(italic) then font = font + '*Italic'
    if keyword_set(tiny)   then size=0
    if keyword_set(small)  then size=1
    if keyword_set(medium) then size=2
    if keyword_set(large)  then size=3
    if keyword_set(huge)   then size=4
    if (n_elements(size) eq 0) then size=2
    font_size_strings = ['12', '14', '16', '18', '20']
    size = (size > 0) < (n_elements(font_size_strings)-1)
    font = font + '*' + font_size_strings(size)
```

```
   return, font
endif else if (!version.os_family eq 'Mac') then begin
  font='Helvetica'
  return, font
endif else begin  ; Assume Motif
   font = '-adobe-'
   if keyword_set(helvetica) then font = font + 'helvetica-' else $
   if keyword_set(times)     then font = font + 'times-' else $
   if keyword_set(courier)   then font = font + 'courier-' else $
                              font = font + 'helvetica-'

   if keyword_set(bold) then font = $
          font + 'bold-' else font = font + 'medium-'
   if keyword_set(italic) then font = font + 'o-' else font = font + 'r-'
   font = font + 'normal--*-'

   if keyword_set(tiny)   then size=0
   if keyword_set(small)  then size=1
   if keyword_set(medium) then size=2
   if keyword_set(large)  then size=3
   if keyword_set(huge)   then size=4
   if (n_elements(size) eq 0) then size=2
   font_size_strings = ['80-', '100-', '120-', '140-', '180-']
   size = (size > 0) < (n_elements(font_size_strings)-1)
   font = font + font_size_strings(size)

   if keyword_set(dpi100) then font = font + '100-100-' else $
   if keyword_set(dpi75) then font = font + '75-75-' else $
   font = font + '*-*-'
   font = font + '*-*-iso8859-1'
return, font
endelse

end
```

Mark Rivers   (773) 702-2279 (office)
CARS   (773) 702-9951 (secretary)
Univ. of Chicago   (773) 702-5454 (FAX)
5640 S. Ellis Ave.   (708) 922-0499 (home)
Chicago, IL 60637   rivers@cars.uchicago.edu (e-mail)

or:
Argonne National Laboratory   (630) 252-0422 (office)
Building 434A   (630) 252-0405 (lab)
9700 South Cass Avenue   (630) 252-1713 (beamline)
Argonne, IL 60439   (630) 252-0443 (FAX)

Subject: Re: widget_programming
Posted by Michael Werger on Fri, 04 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Dear all

David and Richard wrote about cross-platform development of GUIs
and the problems with this

[... discussion simply cut ...]

I simply gave up (was trying on Solaris and Windows95/98) - I
got them by default pretty similar but never equal...

I was doing some cross-platform development for Tcl/Tk too and
had similar problems. There I developed simply a routine which
reads a resource file where you can specify font sizes etc.
I think this way might be a good way here too...? Even different
users on the same platform have different preferences for
colors and font sizes

--
Michael Werger ------------o
ESA ESTEC & Praesepe B.V. |
Astrophysics Division mwerger@astro.estec.esa.nl|
| Postbus 299 http://astro.estec.esa.nl |
| 2200 AG Noordwijk +31 71 565 3783 (Voice)
o------------------ The Netherlands +31 71 565 4690 (FAX)

Subject: Re: widget_programming
Posted by David Foster on Tue, 08 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

R.Bauer wrote:
>
> Hi,
>
> position keywords are fine if I use always the same platform for my
> application.
> But unfortunately we have some computers and some platforms where
> positional widgets
> are shown very bad.
>
> The main reason is the different system font and the different screen
> resolutions.
>
> Which fonts are on unix idl and nt idl will give the same results? Is

> there a list avaliable?
>
> Did the fontsize append on the screen resolution and how will I get
> independent to the screen resolution?
>
> I do understand well thats if I defined 1280 * 1024 and I have on an
> other one only 1024 x 768 I got in trouble.
> But if I define 200 * 160 both should look simliar. What happens is
> thats on the best system (it's always the one where the widget was
> builded) all is ok and on an other one all is shifted on top of each
> other. Input fields could not be reached.
>
> R.Bauer

When I ran into this problem trying to port code from Sun/Solaris to
SGI I included the following commands in the "idl_startup" file:

```
   ; Choose pseudo-color 8-bit visual
   device, pseudo_color=8

   ; Select default backing-store method to be provided by IDL, as SGI
   ;  X server does not seem to provide it
   device, retain=2

   ; Change size of font so programs fit on-screen (IRIX 4.0 or later)
   ; Reference: sgi.doc document in $IDL_DIR/notes
   WIDGET_CONTROL, $
      DEFAULT_FONT="-adobe-helvetica-bold-r-normal-*-14-100-*-*-*-*-* "

   ; Set default plotting font to same hardware font above
   ;  (Create a pixmap window to avoid window creation upon
   ;   calling DEVICE, FONT= ; then delete window)
   window, xsize=5,ysize=5,/free,/pixmap
   !p.font = 0      ; Use hardware font
   device, FONT="-adobe-helvetica-bold-r-normal-*-10-100-*-*-*-*-*"
   wdelete          ; Delete window created
```

I used something very similar, without the RETAIN=2, for Solaris 2.6.

From my experience, the two major issues are (1) default font sizes,
and (2) screen resolution. I really like the idea of developing
routines that can choose reasonably sized fonts under multiple
platforms.

As far as getting things to look similar goes, screen resolution
can be an important issue as well. If widgets are too big, chances
are your default font(s) are too big, but if widgets are too small,

chances are your screen resolution may be higher than you're used to.

In our lab, we have actually forced the resolution of our newer Sun
Ultra machines to a lower value than their default, because the higher
resolution was making it too hard for our brain-anatomy people to see
what they were doing. Once you figure out which damn command controls
your particular frame-buffer, it's quite easy to do this. As an example,
on our Sun Ultra1 Creator we use:

```
 /usr/sbin/ffbconfig -res 1152x900x76 now
```

in the user's startup file, before starting the window manager,
and then restore the default resolution later using:

```
 /usr/sbin/ffbconfig -res 1280x1024x76 now
```

Dave Foster
--

```
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~~
    David S. Foster         Univ. of California, San Diego
     Programmer/Analyst     Brain Image Analysis Laboratory
     foster@bial1.ucsd.edu  Department of Psychiatry
     (619) 622-5892         8950 Via La Jolla Drive, Suite 2240
                            La Jolla, CA  92037
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~~
```

Subject: Re: widget_programming
Posted by R.Bauer on Thu, 10 Jun 1999 07:00:00 GMT

View Forum Message <> Reply to Message

David Foster wrote:

> R.Bauer wrote:
>>
>> Hi,
>>
>> position keywords are fine if I use always the same platform for my
>> application.
>> But unfortunately we have some computers and some platforms where
>> positional widgets
>> are shown very bad.
>>
>> The main reason is the different system font and the different screen
>> resolutions.
>>
>> Which fonts are on unix idl and nt idl will give the same results? Is

```
>>  there a list avaliable?
>>
>>  Did the fontsize append on the screen resolution and how will I get
>>  independent to the screen resolution?
>>
>>  I do understand well thats if I defined 1280 * 1024 and I have on an
>>  other one  only 1024 x 768 I got in trouble.
>>  But if I define 200 * 160 both should look simliar. What happens is
>>  thats on the best system (it's always the one where the widget was
>>  builded) all is ok and on an other one all is shifted on top of each
>>  other. Input fields could not be reached.
>>
>>  R.Bauer
>
>  When I ran into this problem trying to port code from Sun/Solaris to
>  SGI I included the following commands in the "idl_startup" file:
>
>      ; Choose pseudo-color 8-bit visual
>      device, pseudo_color=8
>
>      ; Select default backing-store method to be provided by IDL, as SGI
>      ;  X server does not seem to provide it
>      device, retain=2
>
>      ; Change size of font so programs fit on-screen (IRIX 4.0 or later)
>      ; Reference: sgi.doc document in $IDL_DIR/notes
>      WIDGET_CONTROL, $
>          DEFAULT_FONT="-adobe-helvetica-bold-r-normal-*-14-100-*-*-*-*-* "
>
```

That's a good idea!!!!

Thanks
David

```
>
>      ; Set default plotting font to same hardware font above
>      ;  (Create a pixmap window to avoid window creation upon
>      ;   calling DEVICE, FONT= ; then delete window)
>      window, xsize=5,ysize=5,/free,/pixmap
>      !p.font = 0      ; Use hardware font
>      device, FONT="-adobe-helvetica-bold-r-normal-*-10-100-*-*-*-*-*"
>      wdelete          ; Delete window created
>
>  I used something very similar, without the RETAIN=2, for Solaris 2.6.
```

>
> From my experience, the two major issues are (1) default font sizes,
> and (2) screen resolution. I really like the idea of developing
> routines that can choose reasonably sized fonts under multiple
> platforms.
>
> As far as getting things to look similar goes, screen resolution
> can be an important issue as well. If widgets are too big, chances
> are your default font(s) are too big, but if widgets are too small,
> chances are your screen resolution may be higher than you're used to.
>
> In our lab, we have actually forced the resolution of our newer Sun
> Ultra machines to a lower value than their default, because the higher
> resolution was making it too hard for our brain-anatomy people to see
> what they were doing. Once you figure out which damn command controls
> your particular frame-buffer, it's quite easy to do this. As an example,
> on our Sun Ultra1 Creator we use:
>
>   /usr/sbin/ffbconfig -res 1152x900x76 now
>
> in the user's startup file, before starting the window manager,
> and then restore the default resolution later using:
>
>   /usr/sbin/ffbconfig -res 1280x1024x76 now
>
> Dave Foster
> --
>
> ~~~~~~ ~~~~~~~
> David S. Foster   Univ. of California, San Diego
> Programmer/Analyst   Brain Image Analysis Laboratory
> foster@bial1.ucsd.edu Department of Psychiatry
> (619) 622-5892   8950 Via La Jolla Drive, Suite 2240
>   La Jolla, CA 92037
> ~~~~~~ ~~~~~~~