JET PROPULSION LABORATORY

INTEROFFICE MEMORANDUM
SFB: 366-90-02
August 13, 1990

To: Time Warp Fans
From: Steve Bellenot
Re: Porting mplot to XView

The Time Warp tool known as "mplot" was updated to support X-windows. This port was sucessful in the sense that no features were lost. It was a true port in the sense that the same code supports both X and its old window environment, SunView. Also the event handling in mplot was moved from a low level to the "more approved" high level using only callbacks. Support for extended virtual time and for state migration messages was added. However there are problems with the color. (See the Appendices.)

The "toolkit" XView was chosen to implement mplot on X-windows. XView is a Sun product designed for migrating SunView applications to X. XView comes with conversion scripts. Had the event handling in mplot been at a high level, these conversion scripts would have done almost all of the port to X. Except for the color problems below, XView was a good choice for a fast conversion to X. The conversion scripts did not change any of the "pixwin" calls from SunView, they are support in the XView code. Rumor has it that converting these pixwin calls to native Xlib calls speeds up the drawing by between 10 and 30 percent. Mplot still has the original pixwin calls. The version of XView used was 1.0.1, which is the must current version on our machines. XView version 2.0 is now available as a product (as part of Open Windows 2.0). There is a chance that some of the color problems below are "fixed in version 2", but it is my belief that others are likely to appear.

Currently mplot will not display correct colors with all window managers for X-windows. In particular, "twm" decides mplot is illegally changing color maps, and immediately restores the old color map. The window manager "olwm" allows these illegal color map switches, and hence is the window manager to use when running mplot. (Even the sunview version may have a minor color problem. On ice9 (now quartz) mplot has correct colors, but on onyx the background color switches from black to white.)

In some sense the problem with colors is that X-windows and SunView have a different color model. For the moment mplot works fine with SunView and with X using olwm. Most likely, it will become advisable to replace the pixwin calls in mplot with native Xlib calls. However, it is reasonable to wait for XView 2.0 before making such changes.

Eventually I reported my experience to "xviewbugs" an email address which is read by the people at Sun working on XView. The appendices contain copies of the email which bounced back and forth.

This copy of mplot has been added to the TW2.4 source tree.

Appendix A: My email to xviewbugs

Not quite a full bug report but a tale.

XVIEW: a case study (perhaps with bugs?)

The job was to port a color sunview program to XView without trashing the old sunview code. (The same source with a few '#define's will produce the old sunview program when compiled with -DSUNVIEW, and an X windows version would be produced when the source was compiled with -DXVIEW.)

The sunview program used a 'writemask' technique for overlaying information which could latter be pealed off. This was done by making all the colors 8-15 yellow (the underlying info was drawn in colors 0-7). When it was time to overlay, a call to pw_putattributes (pixwin, &planes) where planes = 8 was made. The text was drawn using pw_text (... PIX_SRC | PIXCOLOR(8).. Eventually the text was cleared via pw_writebackground (... PIX_SRC). This cleared the '8 or 2-to-the=third bit' and color i + 8 returned to color i when i < 8. Another call to pw_putattributes would return the program to normal coloring.

The xview version of the program used these same calls to pw_putattributes, pw_text and pw_writebackground and it worked, but the colors were not right. There few three problems with the color given below together with my work arounds.

1. The first version of the program asked for so few colors that these colors were allocated from the default colormap. Instead of getting colormap locations 0-15, the program got j+0 through j+15, where j was the number of colormaps entries which had already been allocated. xview would change my color i to the windows color i+j tranparently(?). Now when pw_writebackground was called the color 6 now 6+j would be changed to color 6+j-8=j-2 a color the program had no control over, instead of remaining color 6 (or 6+j) as the program did in sunview.

2. The second version of the program solved this by asking for 256 colors so that xview was forced to allocate a new colormap for the program. This caused false colors to appear as X switched colormaps to reflect which window the cursor was in. However, most folks here were using 'twm' which decided that this program was making an illegal change of colormaps and changed the colormap back to the default. The window manager 'olwm' would allow the colormap changes.

3. The third problem was with this call to pw_text. In sunview only the text, and not the background was drawn. The behavior in xview depended on the length of the string to be drawn. If the string was too long to fit in the window, all was well. But it the string was short enough to fit in the window the background would be drawn too, overwriting the graphic below permanently! The ugly but obvious work around, pad the string with ' 's.

The port of the code from sunview to xview had other problems as well. But these problems were more the fault of the program than either sunview or xview. (For example the old program depended on memu's blocking and it called 'sleep'.) The convert_to_xview programs helped, but they missed the change that WIN_EVENT_PROC now is an attribute of the canvas_paint_window instead of the canvas itself.

If I wasn't attempting to retain the sunview version, there will be other ways using X alone to do overlays. (For example, see overlay.c in the xvstuff archive. Do a mail xvstuff@kimba.eng.sun.com and put 'send Answers Mar5-Mar11.90' as the message.)

comments?

steve@topaz.jpl.nasa.gov

Appendix B: Their reply to my first email

From hvr@Eng.Sun.COM Tue Jul 24 14:15:29 1990
To: steve@sapphire.Jpl.Nasa.Gov
Subject: pixwin and plane groups
Cc: xviewbugs@kimba.Eng.Sun.COM

Steve,

pw_putattributes has not been ported to X, so the behavior is not one for one.

To get the plane grouping behavior in X that you were using in SunView, you'll want to use some X functionality. i.e. to get contiguous colormap entries, you need to use the call XAllocColorPlanes (from the PseudoColor visual). This will cause your application to flash against others on the desktop, but not much you can do about it. (If big issue, read in key values from the default colormap and put them in the same cells...or don't use all planes and use reserve colors util in open windows to push the common entries to the bottom of the default colormap--but this is something the end user has to do, so may not be an option.)

Once you have allocated the number of planes to use, then fill up the colormap however you like--just like in SunView.

You can still do colormap buffing in X, it's just different and the model Pixwin uses does not necessarily map well to the model X uses. Dave Lemke did a simple example of colormap buffering called xfade which went out on comp.windows.x.srcs a while ago.

Once the colormap is set up correctly, the pw routines should draw just fine.

Regards,

Heather

----- Begin Included Message -----

Date: Fri, 20 Jul 90 12:35:40 PDT
From: steve@sapphire (Steve)

... text deleted see Appendix A

comments?
    steve@topaz.jpl.nasa.gov

----- End Included Message -----

From smarks@Eng.Sun.COM Tue Jul 24 16:13:39 1990
To: steve@topaz.Jpl.Nasa.Gov (Steve)
Cc: xviewbugs@Sun.COM

I can't comment on all of your message, but only part of it. I'll leave the rest to others.

| 2. The second version of the program solved this by asking for 256 colors
| so that xview was forced to allocate a new colormap for the program. This
| caused false colors to appear as X switched colormaps to reflect which
| window the cursor was in. However, most folks here were using 'twm' which
| decided that this program was making an illegal change of colormaps
| and changed the colormap back to the default. The window manager 'olwm'

| would allow the colormap changes.

You say that "X switched colormaps to reflect which window the cursor was in." X itself (i.e. the server) doesn't do this, and if twm was complaining, then twm wasn't doing it either. I don't think there's any code in XView that explicitly installs colormaps. Was your program calling XInstallColormap() whenever it got an Enter event? If so, then it shouldn't. According to the ICCCM, clients shouldn't install colormaps themselves. That is the responsibility of the window manager. Twm outputs error messages because it is trying to enforce this rule. Olwm is somewhat more forgiving, but that doesn't mean your program is correct.

What you should do instead is to use either the Xlib or XView calls to set up the colormap information for the window manager. Under Xlib, the calls would be XSetWindowColormap() and XSetWMColormapWindows(). Under XView, the attributes to use are WIN_CMS_NAME and WIN_CMS_DATA. Once it has this information, the window manager will install your colormaps for you.

s'marks

Stuart W. Marks
Window Systems Group
Sun Microsystems, Inc.

ARPA: smarks@eng.sun.com
UUCP: sun!smarks

Appendix D: Their reply to my second email

From hvr@Eng.Sun.COM Thu Jul 26 10:48:36 1990
To: smarks@Eng.Sun.COM, steve@topaz.Jpl.Nasa.Gov
Subject: Re: illegal colormap switches
Cc: xviewbugs@Sun.COM

In XView 2.0, the bug is fixed. The bottom of window_cms.c looks like:

```
    /*
     * Tell window manager to install colormap if we aren't using the default.
     * We don't want to set it on the frame, because they are handled by the windo
     * manager automatically.
     */
    if ((cmap != default_cmap) &&
        !(int)xv_get(win_public, XV_IS_SUBTYPE_OF, FRAME_CLASS, 0)) {
        frame_public = (Frame)xv_get(win_public, WIN_FRAME);
        DRAWABLE_INFO_MACRO(frame_public, frame_info);

        atom = XInternAtom(xv_display(info), "WM_COLORMAP_WINDOWS", False);
        XChangeProperty(xv_display(info), xv_xid(frame_info), atom, XA_WINDOW,
                        32, PropModeAppend, (char *)&xv_xid(info), 1);
    }
}
```

Regards,

Heather

From dmaustin@Eng.Sun.COM Thu Jul 26 11:44:38 1990
To: steve@topaz.Jpl.Nasa.Gov
Subject: Re: illegal colormap switches
Cc: xviewbugs@Sun.COM

, smarks@Eng

> Indeed the program was making calls to XInstallColormap. However
> XVIEW was making this call. My program installs the colormap with a
> call to pw_putcolormap (in XView/lib/libxvin/pw/pw_cms.c) which calls
> window_set_cms_data and eventually window_set_cms (both in
> XView/lib/libxvin/window/window_cms.c). The bottom of window_set_cms
> has a bug warning and turns on event processing of LOC_WINENTER and
> LOC_WINEXIT events. Apparently window_default_event_func (which is in
> the file XView/lib/libxvin/window/windowutils.c) now gets called
> for such events, and it calls XInstallColormap. From the comment in
> window_set_cms I guess you already knew about this behavior(bug?).

I just checked the latest version of the XView sources, and there is no
call anywhere to XInstallColormap, as this bug has been removed.

Thanks for pointing this out,
--Darren

From sriram@Eng.Sun.COM Thu Jul 26 12:28:35 1990
To: smarks@Eng.Sun.COM, steve@topaz.Jpl.Nasa.Gov
Subject: Re: illegal colormap switches
Cc: xviewbugs@Sun.COM

From steve@topaz Thu Jul 26 08:11:19 1990
Stuart

Thanks for your help and your fast reply.

Indeed the program was making calls to XInstallColormap. However
XVIEW was making this call. My program installs the colormap with a call to pw_putcolormap (in XView/lib/libxvin/pw/pw_cms.c) which calls window_set_cms_data and eventually window_set_cms (both in XView/lib/libxvin/window/window_cms.c). The bottom of window_set_cms has a bug warning and turns on event processing of LOC_WINENTER and LOC_WINEXIT events. Apparently window_default_event_func (which is in in/window/windowutils.c) now gets called
for such events, and it calls XInstallColormap. From the comment in window_set_cms I guess you already knew about this behavior(bug?).

I suspect you are looking at an outdated version of the XView source.
XView used to install/uninstall colormaps explicitly since there were
no ICCCM compliant window managers that understood the WM_COLORMAP_WINDOWS
property. This deficiency has been fixed for a while, both in the XView
implementation and in the window managers.

sriram/.

From hvr@Eng.Sun.COM Thu Jul 26 17:47:14 1990
To: dmaustin@Eng.Sun.COM, smarks@Eng.Sun.COM, sriram@Eng.Sun.COM,
steve@topaz.Jpl.Nasa.Gov
Subject: Re: illegal colormap switches
Cc: xviewbugs@Sun.COM

Well, ahem...I think Steve now knows that this problem has been fixed ;-)

Heather