Subject: FullScreen: why does not work?
Posted by 281264 on Tue, 14 Sep 2010 16:02:34 GMT
View Forum Message <> Reply to Message

Why?

Cheers,

Javier

File Attachments
1) Fullscreen_test.7z, downloaded 484 times

Subject: Re: FullScreen: why does not work?
Posted by cbpporter on Tue, 14 Sep 2010 16:57:40 GMT
View Forum Message <> Reply to Message

Because FullScreen only effects the next "creation by the underlying windowing API" of the window, while Maximize is an imperative method that changes the status of the window.

Implementation wise one either needs to call some API or force recreate the HWND as is often the case for Windows API when changing a few select parameters.

I'll put it on my never ending list of things to investigate. For now you can call FullScreen in the constructor.

Subject: Re: FullScreen: why does not work?
Posted by alendar on Sun, 05 Dec 2010 03:56:29 GMT
View Forum Message <> Reply to Message

I solved this with Win32 APIs. First I tried to use the U++ wrappers but I could not get them to work.

Here's the code:

```
bool isFullScreenModeActive;

virtual bool Key(dword key, int count) {
 static dword style;
 static Rect normalwindowrect;
```

```
 switch (key) {
  case K_F11:
   if (!isFullScreenModeActive) {
    // Get the current Top window's style bits
    style = GetStyle();
    // Strips the caption bit, which is the title bar
    style = (style & ~WS_CAPTION);
    // Save the "overlapped" or normal window shape
    normalwindowrect = GetRect();
    SetWindowLong(GetHWND(), GWL_STYLE, style);

    // Get the full size of the screen
    long cx = GetSystemMetrics(SM_CXSCREEN);
       long cy = GetSystemMetrics(SM_CYSCREEN);
       // Expand the window to full size
       SetWindowPos(GetHWND(),HWND_TOP,0,0,cx,cy,SWP_SHOWWINDOW);
    //or use Maximize(false) if you like a
    //delayed resizing, even with zoom effects turned off
   } else {
    // Set the caption bit back so we can get our title bar back
    style |= WS_CAPTION;
    // Pass to windows (Style() method gets confused on SyncCaption0)
    SetWindowLong(GetHWND(), GWL_STYLE, style);
    // Return size to normal; key flag is SWP_FRAMECHANGED
    SetWindowPos(GetHWND(),HWND_TOP,
     normalwindowrect.left,
     normalwindowrect.top,
     normalwindowrect.Width(),
     normalwindowrect.Height(), SWP_SHOWWINDOW|SWP_FRAMECHANGED);
    }
    // Track logically in our application
    isFullScreenModeActive = !isFullScreenModeActive;
    break;
 }
 return false;
}
```

This was on MS Windows XP Professional V 2002 SP 3
U++ 2791
This is a win32 only fix. This works on a TopWindow while your in it, so you don't have to use the constructor method.

I might try getting the virtual window size so I can zoom across multiple screens.
The trick to getting the title bar back after your zoom is the SWP_FRAMECHANGED flag.
Without this Windoze doesn't know to rebuild the window frame after you added the WS_CAPTION bit back in.
You can use the TopWindow.Maximize() function, but for some reason it still tries to do effects

when you pass a false bit, so I used the SetWindowPos instead for the zoom.
I haven't tried this with any windows besides a TopWindow.
This covers the taskbar on the bottom on my computer. Sometimes there is a delay in minimizing the taskbar.

Edit from 5 min later:
To go across all screens just change:

```
cx = GetSystemMetrics(SM_CXVIRTUALSCREEN);
  long cy = GetSystemMetrics(SM_CYSCREEN);
  cy = GetSystemMetrics(SM_CYVIRTUALSCREEN);
```

Subject: Re: FullScreen: why does not work?
Posted by koldo on Sun, 05 Dec 2010 10:06:50 GMT
View Forum Message <> Reply to Message

Very interesting .

X11 experts, could it be done in Linux?

Subject: Re: FullScreen: why does not work?
Posted by kohait00 on Tue, 21 Dec 2010 09:47:38 GMT
View Forum Message <> Reply to Message

could this somehow be included in Ctrl::FullScreen() in a platform dependant manner? it'd be really nice to have this feature switchable at runtime..

Subject: Re: FullScreen: why does not work?
Posted by 281264 on Tue, 21 Dec 2010 10:20:35 GMT
View Forum Message <> Reply to Message

Quote:could this somehow be included in Ctrl::FullScreen() in a platform dependant manner? it'd be really nice to have this feature switchable at runtime..

I utterly encourage the implementation of this.

Thanks,

Javier.

Subject: Toggle FullScreen on X11
Posted by oan1971 on Wed, 22 Dec 2010 11:54:07 GMT
View Forum Message <> Reply to Message

Hello Koldo,

I am no X11 expert at all, but the following seems to work on Linux. I guess that the window manager needs to support "Extended Window Manager Hints" (see http://standards.freedesktop.org/wm-spec/wm-spec-latest.html #id2551694).

```
#include <CtrlLib/CtrlLib.h>

using namespace Upp;

class FullScreenTest : public TopWindow
{
public:
 typedef FullScreenTest CLASSNAME;
 FullScreenTest();
 virtual bool Key(dword key, int count);
 virtual void Paint(Draw& w);
 void SetFullScreen(bool b);
};

FullScreenTest::FullScreenTest()
{
 Title("FullScreenTest").Sizeable();
}

void FullScreenTest::Paint(Draw & w)
{
 Rect r = GetView();
 w.DrawRect(r, Blue);
}

bool FullScreenTest::Key(dword key, int count)
{
 if (key == K_F11)
  if (IsFullScreen())
   SetFullScreen(false);
  else
   SetFullScreen(true);
}

void FullScreenTest::SetFullScreen(bool b)
{
 Sizeable(!b);
 FullScreen(b);
```

```
 XEvent event;
 event.xclient.type = ClientMessage;
 event.xclient.serial = 0;
 event.xclient.send_event = true;
 event.xclient.message_type = XAtom("_NET_WM_STATE");
 event.xclient.window = GetWindow();
 event.xclient.format = 32;
 event.xclient.data.l[0] = b;
 event.xclient.data.l[1] = XAtom("_NET_WM_STATE_FULLSCREEN");
 event.xclient.data.l[2] = 0;
 event.xclient.data.l[3] = 0;
 event.xclient.data.l[4] = 0;

 XSendEvent(Xdisplay, Xroot, false, SubstructureRedirectMask | SubstructureNotifyMask,
&event);

 if (b)
  SetRect(0, 0, Xwidth, Xheight);
}

GUI_APP_MAIN
{
 FullScreenTest().Run();
}
```

Best regards,
Oliver

Subject: Re: Toggle FullScreen on X11
Posted by koldo on Wed, 22 Dec 2010 13:12:49 GMT
View Forum Message <> Reply to Message

Great. I have to try it .

"Extended Window Manager Hints" seems to be supported at least by Gnome.

Subject: Re: Toggle FullScreen on X11
Posted by koldo on Sun, 26 Dec 2010 16:10:25 GMT
View Forum Message <> Reply to Message

Windows code runs perfect, but X11 no. And I do not understand: oan1971 code seems to be right.

I have included both in last SDLCtrl_demo, demo1 (demo1main.cpp).

Subject: Re: Toggle FullScreen on X11
Posted by 281264 on Mon, 27 Dec 2010 10:22:20 GMT
View Forum Message <> Reply to Message

Windows version works but partially in my computer (Windows XP SP3), for it is happening the following:

1.- the lower Windows Tool Bar is not occupied;

2.- when pressing F11 successively, the size of the minimized windows is smaller and smaller; why?

3.- I reckon that the correct initialization of isFullScreenModeActive=false; Is it correct?

4.- I do not understand this:

Quote:Edit from 5 min later:
To go across all screens just change:

```
cx = GetSystemMetrics(SM_CXVIRTUALSCREEN);
long cy = GetSystemMetrics(SM_CYSCREEN);
cy = GetSystemMetrics(SM_CYVIRTUALSCREEN);
```

So in my case it partially works. Is there any special consideration that I am not aware?

Thank you.

Best wishes,

Javier

Subject: Re: Toggle FullScreen on X11
Posted by mirek on Mon, 27 Dec 2010 17:52:55 GMT
View Forum Message <> Reply to Message

TopWindow FullScreen fixed in X11...

Mirek

Subject: Re: Toggle FullScreen on X11
Posted by 281264 on Mon, 27 Dec 2010 18:04:01 GMT
View Forum Message <> Reply to Message

Mirek, please don't forget Windows users...

Subject: Re: Toggle FullScreen on X11
Posted by mirek on Mon, 27 Dec 2010 18:32:34 GMT
View Forum Message <> Reply to Message

281264 wrote on Mon, 27 December 2010 13:04Mirek, please don't forget Windows users...

I do not. Unfortunately, I have spent last hour trying to do that, to no avail. That said, original contract was that it would work only before opening the window...

Mirek