Subject: Problem with ScrollBar and multiple controls
Posted by ultimatecoding on Tue, 02 Jul 2019 23:00:00 GMT
View Forum Message <> Reply to Message

Hello,

I'm pretty new to Ultimate++, and I've been fiddling with it on and off for a couple of days now. Everything was going well until I decided to test my concept.

The general idea is that I would like to have something like a panel containing sub-controls in a form of a vertical list. Now, I want to be able to scroll through those controls. Here's a simplified version of what I've come up with so far:

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

using namespace Upp;

struct ButtonPanel : Ctrl
{
    Vector<Button*> buttons;
    const int btnHeight = 100;

    ButtonPanel(int n)
    {
        for (int i = 0; i < n; ++i) {
            Button *btn = new Button();
            btn->SetLabel("Button " + AsString(i + 1));
            buttons << btn;
            *this << btn->HSizePos(0, 0).TopPos(i * btnHeight, btnHeight);
        }
    }

    ~ButtonPanel()
    {
        for (Button *btn : buttons) {
            delete btn;
        }
    }

    int getHeight()
    {
        return buttons.size() * btnHeight;
    }

};

struct MainWindow : TopWindow
{
```

```
    ButtonPanel btnPanel;
    ScrollBar scrlBar;
    bool isLoaded;

    MainWindow() :
        btnPanel(163), // <-- here
        isLoaded(false)
    {
        CenterScreen().Sizeable().Zoomable();
        Title("ScrollTest");
        AddFrame(scrlBar);
        scrlBar.SetLine(btnPanel.btnHeight);
        scrlBar.SetTotal(btnPanel.buttons.size() * btnPanel.btnHeight);
        scrlBar.WhenScroll = [=] { Refresh(); };
    }

    virtual void Layout()
    {
        scrlBar.SetPage(GetSize().cy);
    }

    void MainWindow::MouseWheel(Point, int zdelta, dword)
    {
        scrlBar.Wheel(zdelta);
    }

    virtual void Paint(Draw& w)
    {
        w.DrawRect(GetSize(), SGray);
        if (!isLoaded) {
            *this << btnPanel.HSizePos(0, 0).TopPos(0, btnPanel.getHeight());
            isLoaded = !isLoaded;
        }
        // moving entire panel based on scroll position
        btnPanel.TopPos(-scrlBar, btnPanel.getHeight());
    }
};

GUI_APP_MAIN
{
    MainWindow mainWin;
    mainWin.Run();
}
```

It seems that for heights roughly above 16,300 pixels, things start getting out of control. For example, if I change the number of buttons from 163 to 164 in the code above, I'm not able to see any buttons.

Unless there's some internal bug I'm unaware of, I honestly can't see what I'm doing wrong (at least conceptually).

Any help would be greatly appreciated. If there's a better way to accomplish what I'm trying to do, please also let me know.

Thank you.

Subject: Re: Problem with ScrollBar and multiple controls
Posted by mirek on Thu, 04 Jul 2019 08:58:45 GMT
View Forum Message <> Reply to Message

ultimatecoding wrote on Wed, 03 July 2019 01:00Unless there's some internal bug I'm unaware of, I honestly can't see what I'm doing wrong (at least conceptually).

There is internal limitation, positions are limited to +/-16383. The general reason is that sometimes we are using large numbers of widgets, e.g. in ArrayCtrl, so optimizing sizeof(Ctrl) makes sense.

Quote:
If there's a better way to accomplish what I'm trying to do, please also let me know.

You can definitely achieve what you want, you just need to place widgets relatively to view and eventually just hide / show those that are out of range. Or alternatively place them to some "out of view" position (liky y pos at 32000). (This is done e.g. in ArrayCtrl::SyncLineCtrls(int i, Ctrl *p) near the end of the method).

Also, doing this in Paint is ugly. You should rather have some "Sync positions" methods and call it from WhenScroll...

Mirek

Subject: Re: Problem with ScrollBar and multiple controls
Posted by ultimatecoding on Sat, 06 Jul 2019 21:07:47 GMT
View Forum Message <> Reply to Message

Thank you for your reply and the explanation.

mirek wrote on Thu, 04 July 2019 10:58You can definitely achieve what you want, you just need to place widgets relatively to view and eventually just hide / show those that are out of range.
That's the workaround I've found. It seems to work correctly for regular controls, but when I try to add an OpenGL control, such as the one used here

https://www.ultimatepp.org/reference$OpenGL$en-us.html, the part of the control that should be hidden as I scroll is actually drawn on top of any other controls, such as menu bars or tool bars. It looks like it's constrained only by the window itself. Is it possible to have multiple OpenGL controls that would be scrollable without that side-effect?

mirek wrote on Thu, 04 July 2019 10:58Also, doing this in Paint is ugly. You should rather have some "Sync positions" methods and call it from WhenScroll...
Yeah, now I see that's not the best idea. I was basically just using this https://www.ultimatepp.org/reference$ScrollBar$en-us.html as my template without giving it too much thought.

Thanks again.

Subject: Re: Problem with ScrollBar and multiple controls
Posted by mirek on Tue, 09 Jul 2019 06:20:12 GMT
View Forum Message <> Reply to Message

ultimatecoding wrote on Sat, 06 July 2019 23:07Thank you for your reply and the explanation.

mirek wrote on Thu, 04 July 2019 10:58You can definitely achieve what you want, you just need to place widgets relatively to view and eventually just hide / show those that are out of range.
That's the workaround I've found. It seems to work correctly for regular controls, but when I try to add an OpenGL control, such as the one used here https://www.ultimatepp.org/reference$OpenGL$en-us.html, the part of the control that should be hidden as I scroll is actually drawn on top of any other controls, such as menu bars or tool bars. It looks like it's constrained only by the window itself. Is it possible to have multiple OpenGL controls that would be scrollable without that side-effect?

Well, OpenGL is sometimes tricky. What is your platform?

Mirek

Subject: Re: Problem with ScrollBar and multiple controls
Posted by ultimatecoding on Wed, 10 Jul 2019 15:20:41 GMT
View Forum Message <> Reply to Message

I was testing that on Windows 10.

I can think of two solutions:
 Decrease the height of the OpenGL control as it goes off-screen while trying to prevent its content from moving/resizing. Use just one OpenGL control with several vertical sections and try to synchronize it with the scroll bar.

Another option would be to drop OpenGL altogether and try to use Ultimate++'s drawing system, since the shapes I'd like to draw are not very complex, and they're not necessarily 3D. However, using OpenGL would make that part of the code more portable, so I kind of want to stick with OpenGL. I'm trying to avoid relying too much on a specific GUI library/framework.

By the way, ArrayCtrl seems to have the same problem.