Subject: Parent continues before child is painted
Posted by slashupp on Sun, 01 Nov 2015 08:42:58 GMT
View Forum Message <> Reply to Message

(debian 64bit)

How can I synchronize updates to a child control so that function calls in my app are guaranteed to happen after the paint-event of the child has completed?
I've tried Sync(), both in the parent and the child but to no avail.
The child's paint-event calculates and sets values in the items that are painted (their x,y-coord's) which I want to use elsewhere. This happens during start-up in the ctor of the parent, but the parent ctor always finishes before the paint-event of the child is called - meaning the x,y's are zeros. I want to ensure that the child's paint-event happens in sequence to the function-calls in the parent-ctor. How can I force this?

Subject: Re: Parent continues before child is painted
Posted by slashupp on Sun, 01 Nov 2015 10:00:01 GMT
View Forum Message <> Reply to Message

A bit of hack for a workaround:
I added a Callback WhenPaintDone to the child which I then call at the end of the Paint()-routine.
but this should not be necessary - there should be a way to force painting to happen sequentially.

Subject: Re: Parent continues before child is painted
Posted by dolik.rce on Sun, 01 Nov 2015 11:15:15 GMT
View Forum Message <> Reply to Message

Hi slashupp,

I don't know the answer to your question, but I think there is something wrong with your design, if you need to paint the controls in given order...

Each control should only display some state of your application. If there are data that are visualized by more then one control, then it should not be "owned" by any of them, but by some higher level entity, which takes care about it's state and calls Refresh() on all of the controls when necessary (e.g. when the data change).

Best regards,
Honza

Subject: Re: Parent continues before child is painted

Posted by mirek on Sun, 01 Nov 2015 13:06:02 GMT
View Forum Message <> Reply to Message

Agreed, this is really a bad design. Paint should basically be 'const' (although in special cases, this rule can be broken). Order is not guaranteed.

Now I can suggest two solution, based on what values you expect to 'compute' in Paint.

Model situation 1: You need some coordinates computed during Paint e.g. to use in LeftDown. If that is the case, separate painting into some routine other than ::Paint. In Paint, call this other routine. When you need coordinates computed, call this routine with "NilDraw". If there are some really heavy computation, you can add bool flag to avoid them.

Model situation 2: Some values displayed are slow to compute, you do not want to repeat this. In this case, use lazy evaluation/cache to compute them...

Mirek

Subject: Re: Parent continues before child is painted
Posted by slashupp on Sun, 01 Nov 2015 13:47:55 GMT
View Forum Message <> Reply to Message

Hi mirek

The "work-area" is bigger than the actual "view-area" of the window.
a simplified description of my "bad?" design computes positions for items in the work-area and updates the item's x,y variables. I want to select one item at random and scroll that portion of the work-area into view (if it is outside) but cannot do so because the x,y values are still zero's.

I'll try to do as you suggest and see if I can pre-compute the x,y's in another function
(if it doesn't work I'll still have my ugly hack above)

Another question: Why doesn't Sync() work as advertised: "Forces immediate repainting of areas marked using Refresh, RefreshFrame or ScrollView methods." - this would have done the trick nicely.

Subject: Re: Parent continues before child is painted
Posted by slashupp on Sun, 01 Nov 2015 14:25:55 GMT
View Forum Message <> Reply to Message

Okay! I did not know about NilDraw, and it went way easier than I expected
Implemented it something like this:

...
NilDraw nd;

myctrl.paint_helper(nd, vmydata); //paint_helper() extracted from Paint() for computations
//and the only thing Paint(Draw &w) does is to call paint_helper(w, vmydata);
...
myctrl.ensure_visible(vmydata[idx]);
...

and it works!

This NilDraw-call is once only at start-up so the double calculation is OK, don't know if it would be efficient if it's to be done for every paint

Thx mirek

Subject: Re: Parent continues before child is painted
Posted by mirek on Sun, 01 Nov 2015 19:50:18 GMT
View Forum Message <> Reply to Message

slashupp wrote on Sun, 01 November 2015 14:47Hi mirek

The "work-area" is bigger than the actual "view-area" of the window.
a simplified description of my "bad?" design computes positions for items in the work-area and updates the item's x,y variables. I want to select one item at random and scroll that portion of the work-area into view (if it is outside) but cannot do so because the x,y values are still zero's.

I'll try to do as you suggest and see if I can pre-compute the x,y's in another function
(if it doesn't work I'll still have my ugly hack above)

Another question: Why doesn't Sync() work as advertised: "Forces immediate repainting of areas marked using Refresh, RefreshFrame or ScrollView methods." - this would have done the trick nicely.

Let us say it is meant 'visually'...

Hard to say what is going on in your case...
