
Subject: Ctrl: Paint only affected area?

Posted by [Lance](#) on Sun, 20 Mar 2011 12:44:34 GMT

[View Forum Message](#) <> [Reply to Message](#)

Does there exist a mechanism to allow a Ctrl derivative to get informed the actual Rect that needs to be redrawn? virtual Paint(Draw& w); doesn't seem to allow that.

Or the design of Upp has determined the whole area of a Ctrl needs to be redrawn however minor the change might be.

Let's say, we have a really large ArrayCtrl/GridCtrl/etc that spans multiple monitors displaying instant stock market information. Do we need to redraw the whole array when only one cell is changed?

Thanks,
Lance

Subject: Re: Ctrl: Paint only affected area?

Posted by [koldo](#) on Sun, 20 Mar 2011 13:50:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Lance

As fas as I know only by default only the changed subrect is changed so you do not need to redraw it all.

Subject: Re: Ctrl: Paint only affected area?

Posted by [Lance](#) on Sun, 20 Mar 2011 13:54:24 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thank you very much. That's good to know! U++ is terrific.

As a way of learning U++, I was considering to write a Array/Grid control. Not to oversmart the existing ones, just to learn the U++ stuff and approach the problem in a different way. There are too much to learn before I can do that plus I don't really have a lot of time to invest into it right now.

But when I do become less busy, will you be able to mentor me on this topic?

Thank you,
Lance

Subject: Re: Ctrl: Paint only affected area?
Posted by [dolik.rce](#) on Sun, 20 Mar 2011 14:03:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Lance

If I am not mistaken, U++ always repaints only the parts that are necessary. The Paint() method always repaints the entire area of the Ctrl, but only parts that need to be updated are actually transferred to the screen (which is the slowest part of the process usually). The Ctrl itself should usually repaint its whole area. In some cases, you can cache the image inside the Ctrl and change only parts that you know that they need it (e.g. GridCtrl uses this approach I think), but it requires you keep track of what needs to be repainted yourself.

The mechanism works like this: When something needs to be updated, you call its Refresh() method, if only part of Ctrl needs to be updated then use Refresh(Rect r). U++ then calls Paint() on all Ctrls with changed areas and then transfers all the marked areas to the screen. To see which parts of your GUI are repainted and when, call Ctrl::ShowRepaint(50) inside your code, it will help you understand and it is sometimes also quite useful for debugging

There is also a set of methods intended to track the changes inside Ctrl, which can be helpful sometimes: SetModify(), IsModified() and ClearModify(). These are not directly related to the repainting (AFAIK), but they are worth mentioning since they provide common interface that can be often used to optimize the performance (I (mis)use it for caching the Ctrls visuals) while keeping the code readable.

Hope I didn't simplify it too much, but this is actually all I ever needed to know about this

Best regards,
Honza

Subject: Re: Ctrl: Paint only affected area?
Posted by [Lance](#) on Sun, 20 Mar 2011 14:31:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Shall I call you Honza or dolik? I use the Ubuntu ppa you maintain regularly, and thank you for the time you you invested for the community.

Your reply is closer to what I was actually seeking. A little question, as Upp will always mask and transfer the changed Rect or possibly combined area to the screen, what's the point of cache the Ctrl image? I should be able to draw on a blank image representing the Ctrl Rect, and draw only the part that's changed, Upp will mask off other parts before it transfer to the screen anyways.

Am I right or do I misinterpret your answer?

Thank you again,
Lance

Subject: Re: Ctrl: Paint only affected area?
Posted by [dolik.rce](#) on Sun, 20 Mar 2011 14:51:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Lance wrote on Sun, 20 March 2011 15:31 Shall I call you Honza or dolik? I use the Ubuntu ppa you maintain regularly, and thank you for the time you you invested for the community. Honza is my real name, dolik.rce is a nick. You can use whichever you prefer, I respond on both

Lance wrote on Sun, 20 March 2011 15:31 Your reply is closer to what I was actually seeking. A little question, as Upp will alway mask and transfer the changed Rect or possibly combined area to the screen, what's the point of cache the Ctrl image? I should be able to draw on a blank image representing the Ctrl Rect, and draw only the part that's changed, Upp will mask off other parts before it transfer to the screen anyways.

Am I right or do I misinterpret your answer?

You are quite possibly right. I just never tried this, so I am not sure how it works. There might be some hidden obstacles with this approach, e.g. when the whole area needs to be repainted even though nothing or only part has changed (e.g. when minimalized window is restored). This could be worked around if there was a way to obtain description of area which is scheduled for repainting, but AFAIK that is not possible at the moment (the screen area is invalidated right away and the rect is not stored anywhere). But as I say, I never tried so the problems I imagine might not exist or might have a solution

Honza

Subject: Re: Ctrl: Paint only affected area?
Posted by [mirek](#) on Sun, 20 Mar 2011 20:46:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Also check `Draw::IsPainting`.

Generally, you do not use this as drawing is masked out and operations to masked out areas not performed.

However, sometimes there is some processing power associated outside draw operations, then it can be handy to have a way to check whether some area is to be painted...

Mirek

Subject: Re: Ctrl: Paint only affected area?
Posted by [koldo](#) on Sun, 20 Mar 2011 20:55:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Lance

As far as I know, (Honza post in summary), you draw all Ctrl Rect but U++ finally only paints on

the screen updated area (that is the slower side of painting).

This way code is simple but effective.

Put here the slowest part of your code, in a package better, and I am sure it will finally work fast.

Subject: Re: Ctrl: Paint only affected area?

Posted by [Lance](#) on Mon, 21 Mar 2011 11:22:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

Great information. Thanks koldo and Mirek.

I will paint it the lazy way, and look into IsPaint only when there is actually need for further improvement.
