Subject: Paint outside Ctrl Rect
Posted by koldo on Sun, 22 Nov 2009 10:11:20 GMT

View Forum Message <> Reply to Message

Hello all

I have derived a class from Ctrl and I would like to paint outside its Rect, but Ctrl painting clips the painting area to the Ctrl Rect.

Is it a way to disable the Rect clipping in Rect ?

Best regards
Koldo

Subject: Re: Paint outside Ctrl Rect
Posted by mirek on Sun, 22 Nov 2009 22:09:31 GMT

View Forum Message <> Reply to Message

Sort of.

There is an "overpaint" parameter that allows Frames to paint outside of widget (OverPaint is number of widgets "outside").

Mirek

Subject: Re: Paint outside Ctrl Rect
Posted by koldo on Mon, 23 Nov 2009 10:05:17 GMT

View Forum Message <> Reply to Message

luzr wrote on Sun, 22 November 2009 23:09Sort of.

There is an "overpaint" parameter that allows Frames to paint outside of widget (OverPaint is number of widgets "outside").

Mirek

Sorry Mirek

I have added this method to the class:

int OverPaint() const {return 1;};

But nothing happens. I have seen too that it is called in the window constructor but not in the Ctrl Paint function.

Best regards
Koldo

Subject: Re: Paint outside Ctrl Rect
Posted by mrjt on Mon, 23 Nov 2009 10:09:31 GMT
View Forum Message <> Reply to Message

shouldn't that be

virtual int OverPaint() const {return 1;};

?

Subject: Re: Paint outside Ctrl Rect
Posted by koldo on Mon, 23 Nov 2009 16:36:18 GMT
View Forum Message <> Reply to Message

Hello mrjt

Unfortunately the control is clipped too with OverPaint().

Best regards
Koldo

Subject: Re: Paint outside Ctrl Rect
Posted by mirek on Tue, 24 Nov 2009 14:01:17 GMT
View Forum Message <> Reply to Message

koldo wrote on Mon, 23 November 2009 11:36Hello mrjt

Unfortunately the control is clipped too with OverPaint().

Best regards
Koldo

Ctrl view is. Its Frames are not.

Frankly, your request is quite specific, it makes the whole paradigm upside-down. Anyway, we had similar problem in the past as X11 input fields paint outside too. But that is just frame...

In practice, I do no undestand why you would want something like that... It is like requiring regular host platform windows to paint one over another.

But I guess you should be able to reuse frame overpaint to this:

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

using namespace Upp;

struct OverCtrl : public Ctrl, public CtrlFrame {
 virtual void FrameAddSize(Size& sz) {}
 virtual void FrameLayout(Rect& r) {}
 virtual void FramePaint(Draw& w, const Rect& r) {
  w.DrawRect(r.left - 10, r.top - 10, r.GetWidth() + 20, r.GetHeight() + 20, Blue());
  w.DrawRect(r, Red());
 }

 virtual int OverPaint() const { return 10; }

 OverCtrl() {
  SetFrame(*this);
 }
};

GUI_APP_MAIN
{
 OverCtrl ctrl;
 TopWindow win;

 win.Add(ctrl.LeftPos(40, 40).TopPos(40, 10));

 win.Run();
}
```

Mirek

Hello Mirek

Excellent: It works !

Quote:In practice, I do no understand why you would want something like that...

It is easy. Just a sample. Black rectangles are Ctrl ends:

If the Ctrl itself is a line or something derived from it (like an arrow) it is nice to see the ends not clipped, mainly with a big width.

Best regards
Koldo

File Attachments
1) Comparison.PNG, downloaded 944 times

Subject: Re: Paint outside Ctrl Rect
Posted by mirek on Wed, 25 Nov 2009 13:50:40 GMT
View Forum Message <> Reply to Message

koldo wrote on Wed, 25 November 2009 08:18Hello Mirek

Excellent: It works !

Quote:In practice, I do no understand why you would want something like that...

It is easy. Just a sample. Black rectangles are Ctrl ends:

If the Ctrl itself is a line or something derived from it (like an arrow) it is nice to see the ends not clipped, mainly with a big width.

Best regards
Koldo

Still not undestanding For me, the "BAD" is correct one....

Subject: Re: Paint outside Ctrl Rect
Posted by koldo on Wed, 25 Nov 2009 14:33:11 GMT
View Forum Message <> Reply to Message

Hello Mirek

If the Arrow is a Ctrl , I see this better like this instead of clipped.

Best regards
Koldo

Subject: Re: Paint outside Ctrl Rect
Posted by mirek on Wed, 25 Nov 2009 18:30:13 GMT
View Forum Message <> Reply to Message

koldo wrote on Wed, 25 November 2009 09:33Hello Mirek

If the Arrow is a Ctrl , I see this better like this instead of clipped.

Best regards
Koldo

In that case, why do not you just make it bigger? I mean, include 'overpaint' into its size.

Mirek

Subject: Re: Paint outside Ctrl Rect
Posted by koldo on Wed, 25 Nov 2009 20:49:53 GMT
View Forum Message <> Reply to Message

Hello Mirek

Sorry. I do not know what it is exactly OverPaint for.

In this case as this arrow can be entered in the layout designer, it is expected that the shape of the Ctrl both in the layout designer and in the final program has to be the same.

Best regards
Koldo

Subject: Re: Paint outside Ctrl Rect
Posted by mr_ped on Thu, 26 Nov 2009 07:52:19 GMT
View Forum Message <> Reply to Message

The control should adapt to arrow's width, and draw it a tad smaller, so it will end inside rectangle and no clipping will happen. That's the correct way how it should work, but I can see it may be a bit tricky to code that size adjustment by width.

edit: I would probably end with calculating new boundary rectangle as 1/2 of line width inside of the clip rectangle. This way the drawn arrow would be inside the clipping, and like 99+% of

available space would be used (but under certain arrow's angles like the one you posted it would not use 100% of available space). But I think visually it would work very well even in this simple way.

Subject: Re: Paint outside Ctrl Rect
Posted by koldo on Thu, 26 Nov 2009 09:20:54 GMT
View Forum Message <> Reply to Message

Hello mr_ped and Mirek

Just another example:

Left is from layout designer and right is from the program.

If when painting I move and resize the arrows, they will not match as in the designer view.

Best regards
Koldo

```
File Attachments
1) Draw.png, downloaded 907 times
```

Subject: Re: Paint outside Ctrl Rect
Posted by koldo on Thu, 26 Nov 2009 12:20:54 GMT
View Forum Message <> Reply to Message

Well, now arrow painting in layout designer is improved

This is also another opportunity to compare Draw with Painter quality.

```
File Attachments
1) Draw.png, downloaded 927 times
```