Subject: Canvas widget created (Callbacks are great:-)
Posted by ren42 on Mon, 21 Aug 2006 19:38:58 GMT
View Forum Message <> Reply to Message

Hi everybody,
while I'm coding my private app, I missed an useful way to paint
in a rectangular area. So I decided to develop a canvas class
derived from Ctrl to get what i want. I worked since 7 weeks with
UPP and I think it's time to give something back, because UPP is
the best c++IDE I've ever seen. And I try a lot of IDE's
The Attachment include a complete Example. Simply copy it to your
MyApps Nest, create a new Package "uppCanvas" and hit Ctrl-F5.

And now some more Infos:

Canvas provides an easy way to draw on rectangular
area embedded in a parentcontrol of your choice e.g. a StaticRect(so far I tested). But how is it possible to receive mouseclicks or to
paint on Canvas without altering or override the virtual methods in a child class of Canvas? The solution are Callbacks.
Consider a callback as a pointer to a method of Canvas called from another place (the parent window).
In order to draw on the Canvas the parent window must provide and assign methods with matching signaturs to the Callbacks of
Canvas. e.g.:

```
//Declaration
struct App : public TopWindow{
 typedef App CLASSNAME; //Needed by THISBACK macro
 Canvas canvas;        //Prepair your brushes

 App();
 //Same signatur as canvas.paint(Draw& w)
 void OnCanvasPaint(Draw& w); //Want to paint on canvas
 //Paint on Click see below
 void OnCanvasMouseLeft(Point p, dword keyflags);
};

//Definition
App::App(){
 .
 .
 // The magic assignment:-)
 canvas.OnCanvasPaint=THISBACK(OnCanvasPaint);
 // The left mousebutton click is now routed to the canvas
 canvas.OnMouseLeft=THISBACK(OnCanvasMouseLeft);
}
```

```
void App::OnCanvasPaint(Draw& w)
{// Your paintings on Canvas goes here:
  Rect r=canvas.GetSize();
 // Draw a Line
  w.DrawLine(0,0,r.Width(),r.height(),1,Black());
}
/*
And not still enough, Canvas offers a way to paint not only
in a Paint Callback. For this purpose you can use the DrawSheet()
function. To make it short here is an Example:
*/
void App::OnCanvasMouseLeft(Point p, dword keyflags)
{ //Paint an ugly ellipse:-)
  Rect r=canvas.GetSize();
  DrawingDraw& ds=canvas.DrawSheet();
  ds.DrawEllipse(p.x,p.y,20,20,Red(),5,Green());
  canvas.EndSheet();//draw it now
}
```

Consider the Drawsheet as an additional paintlayer you can paint at.
Look at main.cpp to see it all together:-)
Happy coding...

File Attachments

1) uppCanvas.rar, downloaded 6400 times

Subject: Re: Canvas widget created (Callbacks are great:-)
Posted by ren42 on Mon, 21 Aug 2006 20:11:55 GMT
View Forum Message <> Reply to Message

Hi, I made a mistake, sorry

Quote:
Simply copy it to your
MyApps Nest, create a new Package "uppCanvas" and hit Ctrl-F5.

Simply copy it to your MyApps Nest is ok.
You need not to create a new Package. If you Select main Package
click the option "All Packages" and the Package uppCanvas will
appear.

ren

Subject: Re: Canvas widget created (Callbacks are great:-)
Posted by mirek on Mon, 21 Aug 2006 22:07:48 GMT
View Forum Message <> Reply to Message

Nice Drawing/DrawingDraw example

Mirek

Subject: Re: Canvas widget created (Callbacks are great:-)
Posted by ren42 on Tue, 22 Aug 2006 10:05:03 GMT
View Forum Message <> Reply to Message

Quote:Nice Drawing/DrawingDraw example Smile

Thanks a lot

I will develop it further in the next 1 or 2 weeks. I think of
scrollbars, more callbacks of course, and maybe multiple Layers.

Happy coding, ren