Subject: Painter and viewports
Posted by mdelfede on Sat, 29 Nov 2014 11:30:23 GMT
View Forum Message <> Reply to Message

Hi, I'm using painter to draw inside a big working area, let's say of 20'000 (millimeters) size, given by a rectangle from (-10000, -10000) to (10000, 10000).
To do this, I write:

```
PaintingPainter sw(20000, 20000);
sw.Translate(-10000, -10000);
```

I guess it's correct.
Now I want to take a viewport of my big area and display it inside a control, like following picture:

Viewport is given by its origin and a zoom/scale factor.
I can't find a way to do it....

File Attachments
1) painter-Model.png, downloaded 815 times

Subject: Re: Painter and viewports
Posted by Didier on Mon, 01 Dec 2014 19:07:09 GMT
View Forum Message <> Reply to Message

Hello Massimo,

I think Painter and Drawing has all you need.
You don't need to create PaintingPainter sw(20000, 20000); with such a big size.
All you need to do is:

```
PaintingPainter sw(viewPortXsize, viewPortYsize);
sw.Offset(viewportOrigin);
.... paint you're data
sw.end();
```

Ideally you may not even need to allocate a PaintingPainter ==> if Draw interface is sufficient you can reuse the draw instance of the control passed to ViewportCtrl::Paint()

If you really need to use painter, I use the following code (or something close to it), look at GraphDraw::Paint() method (in svn repo).

```
ViewportCtrl::Paint(Draw& w) {
```

```
 ImageBuffer ib( Size() ); // Ctrl size
 Upp::Fill( ib.Begin(), bckgColor, ib.GetLength() ); // if you use transparent colors, you will need
this
 BufferPainter bp(ib, drawMode);
 bp.Offset(viewportOrigin);
 .... paint you're data
 sw.end();
 DrawImage(0, 0, bp);
}
```

In fact, the data is first drawn to an image (using painter) and then the image is draw to the Ctrl
Not very efficient, but I don't know a better way to achieve this when you use Painter.

Subject: Re: Painter and viewports
Posted by mdelfede on Mon, 01 Dec 2014 21:03:50 GMT
View Forum Message <> Reply to Message

Hi Didier,

thank you for answer.

I use Painter because I need to work in floating point units, scale and so on,
and I don't want to replicate all that stuff ;)
Btw, I use a big painting area because I just do drawing regens when I modify it,
then I "move" the viewport over it to pan/zoom, so I spare some graphic calculations.

I found a solution, anyways.
Now I've got the problem of transparency.

I need to paint 2 transparent drawings over a background, i.e.:

```
 // create an imagebuffer to paint inside
 ImageBuffer ib(sz);

 // place it on requested viewport
 BufferPainter bp(ib);
 bp.Scale(scale, scale);
 bp.Translate(-x1 - WORK_AREA / 2, -y1 + WORK_AREA / 2);

 // paint work area
 INTERLOCKED_(doc->RegenMutex()) {
  One<PaintingPainter> &workArea = doc->GetWorkArea();
  if(!workArea.IsEmpty())
```

```
  bp.Paint(*workArea);
}

// paint overlayer
bp.Paint(*doc->GetOverlayArea());

// display it
w.DrawImage(0, 0, ib);
```

workArea and overlayArea are both cleared with RGBAZero() color.
But when I paint on bp object (which has a white background...) I get garbage.
If I clear the painters with a color, all is ok, but I loose the overlay stuff.

Subject: Re: Painter and viewports
Posted by Didier on Mon, 01 Dec 2014 22:34:28 GMT
View Forum Message <> Reply to Message

I think what is missing is :

```
// create an imagebuffer to paint inside
 ImageBuffer ib(sz);
 Upp::Fill( ib.Begin(), bckgColor, ib.GetLength() );   // ********** if the ImageBuffer is not filled, transparency gives garbage
 // place it on requested viewport
 BufferPainter bp(ib);
```

I stumbled accross the same problems when trying to use transparency in background with GraphCtrl (took me some time to figure it out ... and also needed an Upp bug correction)

Subject: Re: Painter and viewports
Posted by mdelfede on Tue, 02 Dec 2014 07:27:35 GMT
View Forum Message <> Reply to Message

Hi,

I solved (ugly) painting on bp an empty, non-transparent painter filled with background color just before
painting the 2 layers on it. It works.... but I don't like very much that solution.

Subject: Re: Painter and viewports
Posted by Didier on Tue, 02 Dec 2014 18:13:09 GMT
View Forum Message <> Reply to Message

Hello Massimo,

backgroundColor can be transparent.
The need for this is that the BP memory contains random data if you don't initialize it.
So when using transparency ... problems start appearing since then the random data gets involved in final result :(

I don't see any other solution
