Subject: Re: smoother drawing
Posted by mr_ped on Thu, 29 Jul 2010 15:06:56 GMT
View Forum Message <> Reply to Message

Did look into that DoPaint out of my curiosity and I think this will raise further questions from you, so to explain:

```
void App::DoPaint(Painter& sw)
{
 if(ctrl.painting) {
  PaintingPainter h(2000, 2000);
  DoPaint0(h);
  sw.Paint(h);
 }
 else
  DoPaint0(sw);
}
```

If you tick that "painting" checkbox, the app is not drawing to that BufferPainter sw directly, but it firstly does allocate Painting disguised into PaintingPainter class (because you will paint to that Painting with Painter).

Painting is UPP class to only store Draw API calls without execution, so you can replay it later upon different target.
So with that checkbox ticked, that demo code does draw the chosen example in "dry run" into Painting, after it is finished, then the Painting is executed upon that ImageBuffer by line "sw.Paint(h);". But you could still use the Painting "h", so you can for example allocate some higher resolution printing area target, and do printingarea.Paint(h); to get the very same painting in different quality by different "painter", for example you can use the old low quality Draw to paint that Painting.

I hope I'm clear enough to be understood within 2-3 rereads.

In case Painting is off, the examples are directly drawing to that ImageBuffer memory area, without storing the API calls anywhere, so it can't be replayed again.