Subject: Re: Font and Image rendering slow
Posted by Oblivion on Sat, 01 Jul 2023 19:40:05 GMT
View Forum Message <> Reply to Message

Hi,

Thanks for the code.
Yesterday I have looked into the Java code and it seems fine and works well.

A quick tour through its U++ counterpart (I will dive deep, later), and here's some "odd" things (for U++ land, of course) I've seen so far (All understandable, as Upp can sometimes look alien even to regular C++ developers):

1) You seem to have duplicated code that is already in Upp: (Rectf, Sizef, Pointf, their operators, etc.)
2) Mixing std::string with Upp code: Not inherently bad, it is possible and sometimes necessary but Upp::String is more optimized and has a lot of optimized functionality you can utilize. Rule of thumb: Use Upp::String with Upp infrastructure whenever possible. Convert to Std only for glue code.

3) You are using BufferPainter, which is a pdf/print quality (subpixel) 2d graphics library for U++. It is great for a lot of things, but it is also very CPU-intesinve.It does not (AFAIK) utilize GPU's 2D hardware acceleration. But it can take advantage of multithreading. Try the below code, and you'll see the difference:

4) (post-post:) You also seem to use iterators a lot. Avoid them where possible. Upp's containers are designed to work with indices. In a wide range of use-cases they are lot faster & easier to use.

```
void frm_main::Paint(Draw& w) {
  Size sz = GetSize();
  ImageBuffer ib(scroll.GetViewSize());
  BufferPainter sw(ib, MODE_ANTIALIASED);

  sw.Co(); // Enable MT.
  sw.Clear(White());
  _icons_paint(sw, scroll.Get().x, scroll.Get().y, render_everything);
  sw.Finish(); // Join.

  w.DrawImage(0, 0, ib);
}
```

Now, of course the above code doesn't really fix the problem.
I am going study your code and try to come up with a diagnosis at least.

Best regards,