Subject: Painter improvements
Posted by mirek on Sun, 24 Dec 2023 14:32:47 GMT
View Forum Message <> Reply to Message

I have spent last month trying to improve Painter, especially MT performance.

New tricks is increasing the number of paths that can be processed together, improving text rendering using new MakeValueTL thread local caching and unlike before, span fills (those with nonuniform color like images or gradients) can now be rendered at the same time (multiple span filled paths can be renderd concurrently with solid fills, before only solid fills could be mixed together). Also inition Clear command is postoned to this rendering phase, which should improve cache locality. As a result, some of PainterExamples are significantly faster in MT, typical is

In addition I have fixed a problem with Image placement being 0.5 pixel off (took 2 weeks to figure that out...) and add new attribute to specify advanced filtering kernels like bicubic or lancsoz 3.

Mirek

Subject: Re: Painter improvements
Posted by Tom1 on Wed, 27 Dec 2023 12:42:37 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Thanks, this is very interesting. However, unfortunately, when built with MSBT22x64 the text rendering is very broken. It shows on PainterExamples (and elsewhere too). Can you check this?

Best regards,

Tom

EDIT: In Core/ValueCache.h add String() conversion for return fixes the text rendering issue. MSBT actually warns about automatic conversion here:

```
template <class K, class M>
String MakeKey_(const K& k, const M& m)
{
 StringBuffer key;
 RawCat(key, StaticTypeNo<K>());
 RawCat(key, StaticTypeNo<M>());
 key.Cat(k());
 return String(key); // << Add String() here!
}
```

I will do further tests...

Subject: Re: Painter improvements
Posted by mirek on Wed, 27 Dec 2023 13:08:20 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Wed, 27 December 2023 13:42Hi Mirek,

Thanks, this is very interesting. However, unfortunately, when built with MSBT22x64 the text rendering is very broken. It shows on PainterExamples (and elsewhere too). Can you check this?

Best regards,

Tom

EDIT: In Core/ValueCache.h add String() conversion for return fixes the text rendering issue. MSBT actually warns about automatic conversion here:

```
template <class K, class M>
String MakeKey_(const K& k, const M& m)
{
 StringBuffer key;
 RawCat(key, StaticTypeNo<K>());
 RawCat(key, StaticTypeNo<M>());
 key.Cat(k());
 return String(key); // << Add String() here!
}
```

I will do further tests...

Thank you!

Subject: Re: Painter improvements
Posted by Tom1 on Wed, 27 Dec 2023 15:47:57 GMT
View Forum Message <> Reply to Message

Mirek,

It seems that the rendering speed has decreased to about half on my hardware:

Processor 12th Gen Intel(R) Core(TM) i9-12900K 3.20 GHz
Installed RAM 32,0 GB (31,7 GB usable)
System type 64-bit operating system, x64-based processor

I will send you a download link to the Sample50k.Painting (a vector based map as a Painting) file separately.

Here's the code for testing:

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

using namespace Upp;


struct MyApp : TopWindow {
 Painting painting;

 MyApp(){
  Maximize();
  LoadFromFile(painting, GetDesktopFolder() + DIR_SEPS + "Sample50k.Painting");
 }

 virtual void Paint(Draw& dw)
 {
  Size sz = GetSize();
  ImageBuffer ib(sz);
  BufferPainter painter(ib);
  painter.Co(true);
  painter.Clear(White());

  int64 t0=usecs();
  painter.Paint(painting);
  painter.Finish();
  Title(Format("Render took %lld usecs",usecs(t0)));
  Image im(ib);
  dw.DrawImage(0,0,im);
 }


 bool Key(dword key, int count){
  switch(key){
   case K_SPACE:
    Refresh();
    return true;
  }
  return false;
 }
};

GUI_APP_MAIN
{
 MyApp app;
 app.MaximizeBox().MinimizeBox().Sizeable();
 app.Run();
}
```

I'm getting about 30 milliseconds for the old (17045) Painter and about 60 milliseconds for the new one.

Hope you're using an UHD/4k or larger display.

Best regards,

Tom

Subject: Re: Painter improvements
Posted by mirek on Wed, 27 Dec 2023 16:13:24 GMT
View Forum Message <> Reply to Message

Just curious: Are you stroking texts to get thin white margins around glyphs?

Subject: Re: Painter improvements
Posted by Tom1 on Wed, 27 Dec 2023 16:47:05 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 27 December 2023 18:13Just curious: Are you stroking texts to get thin white margins around glyphs?
As a matter of fact, I am. It makes it much easier to read them when there are underlaying items getting covered. Is there a more optimized way to do that?

Best regards,

Tom

Subject: Re: Painter improvements
Posted by mirek on Wed, 27 Dec 2023 22:43:15 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Wed, 27 December 2023 16:47Mirek,

I'm getting about 30 milliseconds for the old (17045) Painter and about 60 milliseconds for the new one.

Tom

Please retry. Great benchmarking example, interesting reason for the difference...

Now optimisations were mostly about source spans, so do not expect much improvement, but at

least it should not be slower now.

Mirek

Subject: Re: Painter improvements
Posted by Tom1 on Thu, 28 Dec 2023 11:43:48 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Thanks! Now it performs equally with the old version.

I can create more detailed maps (as Paintings) if you need them for optimization of Painter.

Best regards,

Tom

EDIT: Well, at least almost as good. The new is at 35 ms on the average and the old 17045 runs in 30 ms.
