
Subject: gtk rendering improvements, BufferPainter::PaintOnceHint

Posted by [mirek](#) on Thu, 03 Jun 2021 09:33:39 GMT

[View Forum Message](#) <> [Reply to Message](#)

I have found some bottleneck in gtk rendering and possible improvements:

- Paint was called too often (basically, too soon, after each event) due to event processing code, this is now resolved.
- SetSurface was optimized to use `cairo_image_surface_create_for_data`, which is about 3 times faster than previous code
- With SetSurface fast, when the Image is supposed to be painted just once, SetSurface is the fastest route. To this end I have added PaintOnceHint to ImageBuffer.
- PainterDraw is now using this new hint for much improved performance...

Hopefully this does not break anything...

Mirek

Subject: Re: gtk rendering improvements, BufferPainter::PaintOnceHint

Posted by [Tom1](#) on Thu, 03 Jun 2021 14:01:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Mirek,

Thanks! Sounds interesting. I'll need to take a closer look at this. At first sight it seems to work fine!

Best regards,

Tom

Subject: Re: gtk rendering improvements, BufferPainter::PaintOnceHint

Posted by [Tom1](#) on Fri, 04 Jun 2021 11:25:03 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Mirek,

I did some testing of SetSurface on Linux/Gtk and noticed that it does not work well when using only a part of a larger image. Offset does not seem to work at all there. On Windows, it works very well and can update a 4k screen at 40-50 FPS with SetSurface and 13 FPS with DrawImage. On Linux I get 11-12 FPS for DrawImage.

Please try this testcase:

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

```
using namespace Upp;
```

```
class ResponsiveDrag : public TopWindow {
```

```
public:
```

```
Image im;
```

```
Point offset;
```

```
ResponsiveDrag(){
```

```
Sizeable().MaximizeBox().MinimizeBox();
```

```
Maximize();
```

```
offset=Point(0,0);
```

```
leftdown=false;
```

```
}
```

```
virtual void Paint(Draw &w){
```

```
static int64 last;
```

```
int64 now=usecs();
```

```
double fps=1000000/(now-last);
```

```
last=now;
```

```
// In Windows SetSurface is about 3x faster than DrawImage on 4k UHD screen
```

```
// However, in Gtk SetSurface fails to display the image correctly
```

```
// and offset is not taken into account at all
```

```
SetSurface(w,Rect(GetSize()),~im,im.GetSize(),offset); // Only works on Windows
```

```
// w.DrawImage(Rect(GetSize()),im,Rect(offset,GetSize())); // Works on both Linux and Windows
```

```
w.DrawText(10,10,Format("FPS = %.1f",fps));
```

```
}
```

```
virtual void Layout(){
```

```
Size sz(GetSize()*3);
```

```
ImageBuffer ib(sz);
```

```
BufferPainter p(ib);
```

```
p.Clear(SColorPaper());
```

```
for(int i=0;i<50;i++){
```

```
p.Move(Random(sz.cx),Random(sz.cy));
```

```
p.Line(Random(sz.cx),Random(sz.cy));
```

```
p.Stroke(3,SColorText());
```

```
}
```

```
im=ib;
```

```
offset=Point(GetSize());
```

```
Refresh();
```

```
}
```

```
bool leftdown;
```

Point leftdownp;

```
virtual void LeftDown(Point p, dword keyflags){
    SetCapture();
    leftdown=true;
    leftdownp=p;
}
```

```
virtual void LeftUp(Point p, dword keyflags){
    ReleaseCapture();
    leftdown=false;
    offset=Point(GetSize());
    Refresh();
}
```

```
virtual void MouseMove(Point p, dword keyflags){
    if(leftdown){
        offset=Point(GetSize())+leftdownp-p;
        Refresh();
    }
};
```

```
GUI_APP_MAIN
{
    ResponsiveDrag().Run();
}
```

Best regards,

Tom

Update: PaintOnceHint() almost doubles the DrawImage() FPS on Linux/Gtk to about 20 FPS. On Windows PaintOnceHint() does not have any effect on DrawImage() performance.

Subject: Re: gtk rendering improvements, BufferPainter::PaintOnceHint
Posted by [mirek](#) on Fri, 04 Jun 2021 12:55:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

Tom1 wrote on Fri, 04 June 2021 13:25Hi Mirek,

I did some testing of SetSurface on Linux/Gtk and noticed that it does not work well when using only a part of a larger image. Offset does not seem to work at all there. On Windows, it works very well and can update a 4k screen at 40-50 FPS with SetSurface and 13 FPS with DrawImage. On Linux I get 11-12 FPS for DrawImage.

Update: PaintOnceHint() almost doubles the DrawImage() FPS on Linux/Gtk to about 20 FPS. On Windows PaintOnceHint() does not have any effect on DrawImage() performance.

Uhm, src and offset is not even really implemented, sorry, that is my mess... Frankly, for the dominant use of SetSurface, it really even is not needed. I think removing that SetSurface variant would be enough; you can always achieve setting subpart with Offset and Clip....

Also PaintOnceHint really is only used with Gtk now. Maybe it could be used to optimize MacOS, but that is todo..

Mirek

Subject: Re: gtk rendering improvements, BufferPainter::PaintOnceHint
Posted by [Tom1](#) on Fri, 04 Jun 2021 14:20:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

Please do not remove the SetSurface() variant with src and offset. My scrolling histogram view (on Windows) is absolutely dependent on that as it works on an image buffer which is configured as a larger ring buffer. It is very performance sensitive for fast and smooth scrolling of the view and the current SetSurface() has been the reliable work horse here.

I would certainly like to see it work on Linux too rather than drop out.

Best regards,

Tom

Subject: Re: gtk rendering improvements, BufferPainter::PaintOnceHint
Posted by [mirek](#) on Fri, 04 Jun 2021 15:55:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

Tom1 wrote on Fri, 04 June 2021 16:20Hi,

Please do not remove the SetSurface() variant with src and offset. My scrolling histogram view (on Windows) is absolutely dependent on that as it works on an image buffer which is configured as a larger ring buffer. It is very performance sensitive for fast and smooth scrolling of the view and the current SetSurface() has been the reliable work horse here.

But it should work exactly the same with Offset and Clip... DrawImage is curretyly using this trick already, see Draw/Draw.cpp:134

Subject: Re: gtk rendering improvements, BufferPainter::PaintOnceHint

Posted by [Tom1](#) on Mon, 07 Jun 2021 10:27:05 GMT

[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Fri, 04 June 2021 18:55Tom1 wrote on Fri, 04 June 2021 16:20Hi,

Please do not remove the SetSurface() variant with src and offset. My scrolling histogram view (on Windows) is absolutely dependent on that as it works on an image buffer which is configured as a larger ring buffer. It is very performance sensitive for fast and smooth scrolling of the view and the current SetSurface() has been the reliable work horse here.

But it should work exactly the same with Offset and Clip... DrawImage is currently using this trick already, see Draw/Draw.cpp:134

Hi Mirek,

OK, I see. Did some testing on this. Using Clipoff()+SetSurface() yields similar performance on Windows as SetSurface() with src alone. It just requires a bit more complex code on this side than the original version.

Best regards,

Tom

Subject: Re: gtk rendering improvements, BufferPainter::PaintOnceHint

Posted by [jjacksonRIAB](#) on Tue, 10 Aug 2021 13:56:41 GMT

[View Forum Message](#) <> [Reply to Message](#)

It seems to have improved the problem with screen tearing I was having while scrolling GLCtrls in a container. Cool. Container boundaries are still not obeyed though so the GLCtrl draws on top of everything, but it's a good improvement. Thanks, Mirek!
