

---

Subject: smoother drawing

Posted by [cullam](#) on Thu, 29 Jul 2010 12:33:04 GMT

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

---

Hi guys. I've been creating a re-sizable display in UPP that uses some drawing functionality. Everything is working fine, but I was wondering if there was any way to use anti-aliased lines and edges, and possibly gradient fills. Obviously, using OpenGL is a possible option, but I was hoping to avoid it, since I'm only making simple 2D vector drawings, and there seems to be quite a lot of work to get Open GL up and running in an app.

Basically, what I've done so far works fine, but I'd like to be able to make "smoother" drawings, that don't look like they were whipped up in paintbrush!

Thanks a bunch.

-cullam

---

---

Subject: Re: smoother drawing

Posted by [mr\\_ped](#) on Thu, 29 Jul 2010 13:10:22 GMT

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

---

Did you use the new Painter or how do you draw your current vectors with UPP? The new painter did work very well for me, although I didn't use all things you describe.

---

---

Subject: Re: smoother drawing

Posted by [cullam](#) on Thu, 29 Jul 2010 13:52:06 GMT

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

---

I'm not entirely sure. I basically did what was in the drawing example. My app uses GUI\_APP\_MAIN, with my main class inheriting from WithNameLayout<TopWindow>. I then just overwrote the Paint(Draw& w) function. So, whichever version of painter that is, that's what I used. The actual primitive drawing functions I end up basing my own drawing code on are DrawEllipse(), DrawLine(), etc.

---

---

Subject: Re: smoother drawing

Posted by [mr\\_ped](#) on Thu, 29 Jul 2010 14:10:58 GMT

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

---

check Bazaar PainterSVG\_demo, it does use uppsrc/Painter package too.

Or just add uppsrc/Painter package to your project, add

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

in the code, and go ahead.

Search also forum for painter, watch out for Mirek's (luzr) posts about his experiments when he was creating it.

---

It's not really difficult to use, but the examples/docs are a bit lacking, as it's relatively new part of upp. (but the API is sort of drop-in replacement for previous UPP Draw API, so docs about that one will help even with new Painter package)

---

---

Subject: Re: smoother drawing  
Posted by [mr\\_ped](#) on Thu, 29 Jul 2010 14:12:08 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

I'm not sure and too lazy to check, but I think for your case you need to add Painter package, and use something like PainterDraw class (or DrawPainter) and apply that on that Draw& w.

edit: the DrawEllipse and other API calls should stay the same, so your code should work without change, just find out the way to switch it to new Painter object. In case you fail (unlikely), write here, somebody else will surely give you more exact advice.

---

---

Subject: Re: smoother drawing  
Posted by [mr\\_ped](#) on Thu, 29 Jul 2010 14:27:35 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Check probably at first examples/PainterExamples  
(I didn't see it before, because I had some old examples/reference directories at hdd from before the Painter, had to update first)

---

---

Subject: Re: smoother drawing  
Posted by [cullam](#) on Thu, 29 Jul 2010 14:42:05 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Aha! This is looking much more promising!

So, I'm looking through the code in the example, but without any comments in it, I'm not entirely sure what I'm seeing.

Examples.h

```
#ifndef _PainterExamples_Examples_h_
```

```
#define _PainterExamples_Examples_h_
```

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

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

using namespace Upp;

#define IMAGECLASS TestImg
#define IMAGEFILE <PainterExamples/Test.img>
#include <Draw/iml_header.h>

#define LAYOUTFILE <PainterExamples/Examples.lay>
#include <CtrlCore/lay.h>

void RegisterExample(const char *name, void (*ex)(Painter& pw));

struct App : TopWindow {
    virtual void Paint(Draw& w);

    SplitterFrame split;
    ArrayCtrl list;
    FrameBottom< WithCtrlLayout<StaticRect> > ctrl;

    typedef App CLASSNAME;

    void DoPaint0(Painter& sw);
    void DoPaint(Painter& sw);
```

```

void Print();

void Benchmark();

void Sync();

void ToSlider(EditDouble *e, SliderCtrl *slider);

void ToEdit(EditDouble *e, SliderCtrl *slider);

void Pair(EditDouble& e, SliderCtrl& slider);

void ToSlider();

void Reset();

void Serialize(Stream& s);

App();

~App();

};

#endif

```

All the drawing classes in this example appear to take a reference to a Painter object, but I'm not quite sure where I get this object in the first place. In my app, my main is just:

```

#include "VectorsGUI.h"

#define VERSION "2010070801"
#define FILE_BASE "SQX-Drive"
#define FILE_TITLE String( FILE_BASE )

GUI_APP_MAIN
{
  VectorsGUI(FILE_TITLE, FILE_BASE, VERSION ).Run();
}

```

... with everything of interest happening in my other classes. I believe the Paint() function I'm using came from inheriting my most important class from WithVectorsGUILayout<TopWindow>. What

do I need to modify/do/include to be able to use Painter() instead? Is it some way of getting a Painter& to pass into Paint(), rather than a Draw&?

Thanks for the help. And it is very helpful to know that what I'm looking to do CAN be done simply! Now I just need to figure out how...

---

---

Subject: Re: smoother drawing

Posted by [mr\\_ped](#) on Thu, 29 Jul 2010 14:56:45 GMT

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

---

Ok, some upp "magic" for you, or how I would proceed further:

Select the PainterExamples package

Open Examples.h

.. notice the line with virtual void Paint(Draw& w); declaration, after all you told me you draw your things there, right? So move on that line with cursor on the "Paint" word.

Now Alt+J.

You should end in main.cpp at line 80 with function definition.

And that was quite easy to understand for me, because I recently did some code with Image/ImageBuffer/RasterImage/Painter and other upp classes of that family. For you it will be maybe more cryptic, so here we go:

```
Size sz = GetSize(); //size of painting area of window
if(ctrl.transparent) { //this will create the color chessboard background if you tick that checkbox in
app, otherwise the background is white
  for(int y = 0; y < sz.cy; y += 32)
    for(int x = 0; x < sz.cx; x += 32)
      w.DrawRect(x, y, 32, 32, (x ^ y) & 32 ? Color(254, 172, 120) : Color(124, 135, 253));
  //notice it's drawn into the old Draw thing without new Painter
}
ImageBuffer ib(sz); //this is raw RGBA memory array for SW rendering (allocating the "canvas"
for Painter)

//here's the Painter class finally - BufferPainter is nothing more than
//Painter class extended to paint into ImageBuffer, which was created a line above
BufferPainter sw(ib, ctrl.quality); //init it with desired ImageBuffer and desired antialiasing mode
(from GUI control)
DoPaint(sw); //some custom function which does the actual drawing and I didn't bother to look
inside
//of course just go on the DoPaint with cursor and hit Alt+J to see what's there, if you are curious
enough
//but basically you just need to do sw.drawEllipse, etc.. as you wish
```

```
w.DrawImage(0, 0, ib);  
//and the resulting SW rendered RGBA image is rendered back to OS's window area represented  
by the original Draw& w instance.
```

Looks simple to me, but I'm used to UPP, so keep asking if you don't get anything, I will try to explain better. (but later, going off now)

---

---

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.

---

---

Subject: Re: smoother drawing  
Posted by [cullam](#) on Thu, 29 Jul 2010 16:11:32 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Awesome! OK, I'm working on adding this to my app. First I need to figure out what quality setting I need. Basically, I need to figure out what it's looking for, and what I need to give it. But that should be simple enough. If I get that happening, then the compiler should stop complaining about `BufferPainter sw(ib, ctrl.quality);`, since I'll no longer be handing it a non-existent `ctrl` object.

When I get that done, do I basically just need to implement a `MyApp::DoPaint()` function? Or is there another inheritances I need to sort out to make that connection? Thanks so much for all the help, gang!  
-cullam

---

---

Subject: Re: smoother drawing  
Posted by [cullam](#) on Thu, 29 Jul 2010 17:43:36 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Success!!! The stuff I'm drawing now is all smooth and anti-aliased. Thanks so much for the help, people!  
-cullam

---

---

Subject: Re: smoother drawing  
Posted by [mr\\_ped](#) on Fri, 30 Jul 2010 08:08:20 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

You are welcome. (actually I find this stuff in UPP a total breeze, and enjoy to toy with it a lot, I wish my other projects were so much direct fun)

---

---

Subject: Re: smoother drawing  
Posted by [koldo](#) on Fri, 30 Jul 2010 19:58:58 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hello

Bazaar/Controls4U uses massively Paint and Layout Designer.

In this page you have a summary showing samples of all the controls as seen in Layout Designer and as seen in real programs using Paint.

[http://www.ultimatepp.org/src\\$Controls4U\\$Controls4U\\$en-us.ht ml](http://www.ultimatepp.org/src$Controls4U$Controls4U$en-us.ht ml)

The difference is evident!

---