Subject: AGG
Posted by mirek on Sun, 10 Jun 2007 09:24:39 GMT
View Forum Message <> Reply to Message

Quote:
For me i hope it will be possible to integrate it into Draw instead of GDI.
In the code i don't want to know that for drawing lines is responsible agg or GDI, it schould be transparent.
But in AGG mode the interface should have a bit more options for drawing lines of different widths with any pen etc .

Well, this issue was already resolved and while there could be Draw variant, I strongly believe that AGG should be kept as separate engine. The main reason is that it has much more rich capabilities than Draw, OTOH we want to keep Draw interface minimal (so that it can be easily implemented anywhere). I think in your message, you see that too

What we need is the seamless integration. The real problem there (not that apparent at the first look) is, in nutshell, printing.

That means you should be capable to store AGG painting into Drawing and then put it to the printer, in high resolution (600dpi is U++ standard).

That more or less rules out storing it as raster. There is quite reasonable interface ready in Draw capable of solving this issue - DrawData. In current Draw, it is capable of drawing rescaled images directly from compressed image formats, which was e.g. needed in one of my application, where 300dpi scanned image of full A4 page is printed behind some text produced by application (form-filler style of app). DrawData even tries to "compress" the raster data while sending them to the printer by replacing large uni-color areas with DrawRect and it also performs banding to reduce the memory requirements.

See Draw/DrawData and Draw/DrawRasterData.

Subject: Re: AGG
Posted by mirek on Sun, 10 Jun 2007 10:29:51 GMT
View Forum Message <> Reply to Message

Related archived thread:

http://www.ultimatepp.org/forum/index.php?t=msg&th=579&a mp;start=0&

Also, looks like we are stuck with AGG 2.4 version.

Mirek

Subject: Re: AGG
Posted by Novo on Mon, 11 Jun 2007 02:30:06 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 10 June 2007 06:29Also, looks like we are stuck with AGG 2.4 version.

Insider's information: the author of AGG can make a separate license for UPP. You'll need just ask him and explain reasons.

Subject: Re: AGG
Posted by mirek on Mon, 11 Jun 2007 08:00:29 GMT
View Forum Message <> Reply to Message

Novo wrote on Sun, 10 June 2007 22:30luzr wrote on Sun, 10 June 2007 06:29Also, looks like we are stuck with AGG 2.4 version.

Insider's information: the author of AGG can make a separate license for UPP. You'll need just ask him and explain reasons.

That sounds good, thanks - but I see that a little bit troublesome - I cannot imagine separate license that would work in our case (because it would make AGG sources released under "free" license AFAIK...).

Mirek

Subject: Re: AGG
Posted by Novo on Mon, 11 Jun 2007 13:54:11 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 11 June 2007 04:00
it would make AGG sources released under "free" license AFAIK...

AGG was previously "free", and, as I know, old customers got separate licenses. If you really want to use AGG in UPP just ask the author. He is a good guy.

Subject: Re: AGG
Posted by mirek on Mon, 11 Jun 2007 15:28:36 GMT
View Forum Message <> Reply to Message

Ok, how should it look like (IMO) interface-wise:

```
void SomePaintRoutine(Draw& w)
{
   AGGDraw agg(w, 10, 10, 200, 200);
   agg.DrawRect(...); // AGGDraw is derived from Draw
   agg.DrawBezierCurve(...); // But also has some specific methods..
   // ... destructor of AGGDraw "flushes" it into draw
}
```

Now the important part -> this should work with Drawing and PdfDraw and PrinterDraw too, in that case drawing ops get recorded into binary stream and "flush to draw" will be performed by DrawData.

I guess there should be check whether we are painting to the screen, in that case the image would be drawn directly into ImageBuffer and applied using DrawImage.

Intitial implementation of course can use this ImageBuffer path only (not be bothered with Drawing / DrawData).

Subject: Re: AGG
Posted by arturbac on Tue, 19 Jun 2007 21:12:42 GMT
View Forum Message <> Reply to Message

Design that don't work at all
What I do wrong ?

File Attachments
1) aggdraw_design.7z, downloaded 478 times

Subject: Re: AGG
Posted by fudadmin on Fri, 22 Jun 2007 02:27:43 GMT
View Forum Message <> Reply to Message

Hi, this this my old agg package. Maybe some people find it useful...

File Attachments
1) agg_aris_main.tar.gz, downloaded 559 times

Subject: Re: AGG
Posted by fudadmin on Fri, 22 Jun 2007 02:48:50 GMT
View Forum Message <> Reply to Message

...this is a testcase for upp_agg_svg...

Sorry, it's old and messy - I was trying xml parsing with upp before uncovered upp Xml capabilities (or maybe they didn't exist at that time...) but it works and shows the potential.

SvgView::Paint(Draw& w) - could give you some ideas to experiment with different aspects of agg connection to upp.
ImagBuffer::Rectangalize(Draw& w, bool norect) is the main thing, IMHO, to sort out Mirek's mentioned printing problem:
it finds mono-color rectangles and can be used to send DrawRect to printer instead of pixels. (it might not in this example but I had tested printing, too)
Ok, I might be back...

File Attachments
1) XmlView_svg2.tar.gz, downloaded 496 times

Subject: Re: AGG
Posted by fudadmin on Fri, 22 Jun 2007 02:54:55 GMT
View Forum Message <> Reply to Message

These are svg_examples to play with. They are a bit corrected by me (according to standards, imo) and might differ from those of agg website.
P.S. you should unpack the examples into ~/svg_examples/ to test with XmlView_svg2

File Attachments
1) svg_examples.tar.gz, downloaded 538 times

Subject: Re: AGG
Posted by arturbac on Fri, 22 Jun 2007 08:37:37 GMT
View Forum Message <> Reply to Message

Thanks.

Subject: Re: AGG
Posted by fudadmin on Fri, 22 Jun 2007 23:56:33 GMT
View Forum Message <> Reply to Message

arturbac wrote on Fri, 22 June 2007 09:37Thanks.
You welcome.
A. Some more considerations about agg or what people expect from agg:
1. the questions I didn't answer to myself:
"Would be it possible (or desired.. or required) too "agg-ise" existing:
a) the whole application
b) only some controls
c) only some elements of controls
with e.g let's say (kind/part of "chameleon"?)

Draw::SetEngine("AGG"/"VNC"/"HOST"/"SDL/"GLX"..., SetScreen("whole_screen", 10times x 10times), ElementsFilter(ArrayMap("RichEdit","text")))
d) all text
...
or only use AggDraw in e.g virtual Paint() for each users needs in new controls to draw something (like Mirek's proposal?) ?

2. In case of controls- "how to switch/connect between subpixel accuracy and existing "no double point" coordinates (something like existing "Splitter"...?)

3. Would all those upp AggDraw::DrawSomething(...) wrapper things just be an extra bloat when at the moment with my Imag buffer (kind of canvas?) I have all the flexibility of agg?
-partial answer to this: ok, maybe people would not need to learn "agg way of doing things" but is it worth?

B. if accepted, "agg_aris_main" package should be renamed agg2.4_uppBase1 (or similar) and go to under /plugins and not needed to be changed except:
1. maybe - agg_svg_upp (svg related files) should be extracted into a separate package? - Because (at least me) it is going to be adjusted for Upp::Xml
2. maybe - someone (or me in some future) will want to adjust agg2.4_uppBase1 to upp containers (then agg2.4_uppBase2?)
3. I would suggest then to have different agg_upp_bind (or agg2.4UppDraw) packages and/or variants (I will have my variants anyway)
4. related to my naming scheme - I'm seriously thinking about a mechanism (regular exp aliases? and/or version control) allowing showing and switching between different branches/versions in the theide.

Aris

Subject: Re: AGG
Posted by arturbac on Sat, 23 Jun 2007 18:12:20 GMT
View Forum Message <> Reply to Message

I have 2 questions :
1.Why You use ImagBuffer instead of ImageBuffer from Upp ?
2.What exactly do Rectangelize ?

Subject: Re: AGG
Posted by fudadmin on Sat, 23 Jun 2007 20:11:58 GMT
View Forum Message <> Reply to Message

arturbac wrote on Sat, 23 June 2007 19:12I have 2 questions :
1.Why You use ImagBuffer instead of ImageBuffer from Upp ?
2.What exactly do Rectangelize ?

1.
a) You can try ImageBuffer. At least at that time (something a year ago... when I was creating upp-agg things) ImaggeBuffer seemed to me like having not enough functionality.
b) Also, I had more extra methods in it and achieved some agg functionality without agg (like moving lines by fractions of a pixel(tried 1/10th), line smoothing etc.).
c) I had idea to use it as memory canvas and paint different things (with and without agg) to it and at the end of a cycle:

2. Rectangalize - to save graphics bandwidth and speed up throwing it to screen (or printer) by not sending pixel by pixel but employing hardware's DrawRect (this idea was proposed by Mirek if you browse forum archieves). Especially a big speed up with printers if you ever tried to print BIG rendered images...

P.S. The thing which Mirek wanted me to do was recording and replay of agg drawings in WMF style... But despite a lot of efforts my brain never came to that realization... . Maybe joint efforts can produce that...

Subject: Re: AGG
Posted by mirek on Sun, 24 Jun 2007 09:02:35 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Sat, 23 June 2007 16:11
2. Rectangalize - to save graphics bandwidth and speed up throwing it to screen (or printer) by not sending pixel by pixel but employing hardware's DrawRect (this idea was proposed by Mirek if you browse forum archieves). Especially a big speed up with printers if you ever tried to print BIG rendered images...

Actually, you got this point wrong back then...

What is needed is in the fact the ability to render the image in bands. In other words, to draw the clip of image only. Together with "WMF-style" recording, U++'s DrawData function will be able to draw the image in only small portions (bands) and will perform the "rectanglizations" (this code is already in Draw package and tested to work in some of mine commercial applications to directly draw supported image formats like .png).

OTOH, I think this whole issue can wait a bit. Painting on the screen seems to be the primary problem now anyway....

Mirek

Subject: Re: AGG
Posted by fudadmin on Sun, 24 Jun 2007 11:30:27 GMT

luzr wrote on Sun, 24 June 2007 10:02
...
Actually, you got this point wrong back then...

You mean, now I understand it, do you?

Quote:
What is needed is in the fact the ability to render the image in bands. In other words, to draw the clip of image only. Together with "WMF-style" recording, U++'s DrawData function will be able to draw the image in only small portions (bands) and will perform the "rectanglizations" (this code is already in Draw package and tested to work in some of mine commercial applications to directly draw supported image formats like .png).
Yes, I just made your ingenious piece of code to work with agg and blend with upp Ctrls (with controlled transluency layers). But I also wanted to detect refreshable rectangles from the whole application and pump through my ImagBuffer system. Maybe the other way round and wrong approach...

Quote:OTOH, I think this whole issue can wait a bit.
Do you mean, agg?
Quote:
Painting on the screen seems to be the primary problem now anyway....

Could you elaborate? You mean agg painting or upp painting? (if upp, is this the reason upp blinks horribly with beryl when resizing?)

P.S XmlView_svg2 which I posted is not all what have done with upp and agg. Some other things were e.g inserting resizable png and svg imges (EDIT: actually as links not the whole images) into UWord and exporting qtf into xhtml (not HTML).
But the biggest drawback of theide (or maybe my inability) to have good version control switching is that I have to spend too much time adjusting my changes to official versions.

Subject: Re: AGG
Posted by mirek on Sun, 24 Jun 2007 11:59:44 GMT

Quote:
Do you mean, agg?

I mean the printing problem.

Mirek

Subject: Re: AGG
Posted by fudadmin on Mon, 09 Jul 2007 23:55:34 GMT
View Forum Message <> Reply to Message

ok, if you'd like this style:

```
  AggDrawer agd(RectC(0,0,sz.cx, sz.cy));
  agd.SetBackground(agg::rgba(0, 1.0, 1.0, ~optR));
//  agd.SetBackground(agg::rgba(1.0, 0.5, 0.5));

  agd.DrawLine(0,0,100,100,10);
  agd.SetBrushColor(agg::rgba(0, 0, 0));
//  agd.DrawLine(100,0,200,100,10);
  agd.DrawEllipse(sz2.cx, sz2.cy, sz2.cx-10, sz2.cy-10, 3);

  agd.End(w, ~optR);
```

here is the plugin:

File Attachments
1) plugin_agg24.tar.gz, downloaded 583 times

Subject: Re: AGG
Posted by fudadmin on Tue, 10 Jul 2007 00:00:02 GMT
View Forum Message <> Reply to Message

here is bind package to play with.

File Attachments
1) agg_upp_bind.tar.gz, downloaded 699 times

Subject: Re: AGG
Posted by fudadmin on Tue, 10 Jul 2007 00:09:14 GMT
View Forum Message <> Reply to Message

here is test bind.

P.S If I was Mirek, I would advice aris about pixel format files and add that plugin to upp ...

File Attachments
1) agg_upp_bind_test.tar.gz, downloaded 481 times

Subject: Re: AGG
Posted by mirek on Tue, 10 Jul 2007 18:40:33 GMT

fudadmin wrote on Mon, 09 July 2007 20:09here is test bind.

Good job!

Mirek

Subject: Re: AGG
Posted by arturbac on Mon, 16 Jul 2007 15:37:44 GMT

Thanks to fudadmin i got AGGDraw up to work with ImageBuffer & DRAW_OP scheme, I was so close
thanks again fudadmin.

I used fudadmin agg24plugin becouse it nice separates agg from aggdraw

Everything in aggraw is hidden from public in private impl. so agg headers will not be included into clients sources.

Not we can work a bit on it, at least working start.

File Attachments

1) aggdraw_v0.00.7z, downloaded 487 times

Subject: Re: AGG
Posted by mirek on Mon, 16 Jul 2007 21:40:22 GMT

Great news!

Looks like a good idea to make it in two steps:

First step as "RAW" ImageBuffer producing thing.

And later, if needed, we can add second step (the one capable of printing).

I guess, for the first step we can just use plain agg interfaces plus simple Draw interfaces (so that you can actually use existing Draw code too).