Subject: Cairo
Posted by kodos on Tue, 06 Jan 2009 14:37:35 GMT
View Forum Message <> Reply to Message

Hi,

After I read that you are working on a AGG draw control I want to post my unfinished but working cairo control.

It works on Windows and Linux, supports double buffering, easy PDF and SVG output, a Draw compatible drawing class.

To compile the code you have to replace the png.upp file in your uppsrc dir with the one in the zip file.

Included in the zip file are also 2 demo programs.

File Attachments
1) cairo.zip, downloaded 530 times

Subject: Re: Cairo
Posted by mirek on Tue, 06 Jan 2009 19:32:20 GMT
View Forum Message <> Reply to Message

kodos wrote on Tue, 06 January 2009 09:37Hi,

After I read that you are working on a AGG draw control I want to post my unfinished but working cairo control.

Well, I am afraid that providing control is the same mistake as has been done in previous attempts.

We do not need control.

What we REALLY need is to draw to ImageBuffer. And provide meaningful way how to store such drawing and how to print it (like DrawingDraw).

This really is multitarget effort - and one important target is to support drawing in non-GUI apps (think webservers).

Once you have rendered raster in ImageBuffer, providing Ctrl is trivial.

Mirek

Subject: Re: Cairo
Posted by mirek on Tue, 06 Jan 2009 19:45:29 GMT
View Forum Message <> Reply to Message

PS.: Thanks anyway, it is a good resource to interface finetuning. (In reality, what I have now is pretty similar).

Subject: Re: Cairo
Posted by kodos on Tue, 06 Jan 2009 22:23:54 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 06 January 2009 20:32
What we REALLY need is to draw to ImageBuffer. And provide meaningful way how to store such drawing and how to print it (like DrawingDraw).

This really is multitarget effort - and one important target is to support drawing in non-GUI apps (think webservers).

Once you have rendered raster in ImageBuffer, providing Ctrl is trivial.

Mirek

The Linux control is actually just a wrapper around my cairo class which renders to an image buffer and after that the image buffer is rendered to a control. I think cairo also supports printing, but never tested that one .

This are the 4 methods that are currently implemented to set/create a cairo surface:

void SetSurface(cairo_surface_t *surface, int width, int height);
void SetSurface(ImageBuffer &img);
void CreateSvgSurface(String const &filename, double widthInPoints = 595, double heightInPoints = 841);
void CreatePdfSurface(String const &filename, double widthInPoints = 595, double heightInPoints = 841);

Subject: Re: Cairo
Posted by mirek on Wed, 07 Jan 2009 07:14:40 GMT
View Forum Message <> Reply to Message

kodos wrote on Tue, 06 January 2009 17:23
The Linux control is actually just a wrapper around my cairo class which renders to an image buffer and after that the image buffer is rendered to a control.

Ops, sorry then

Quote:
I think cairo also supports printing, but never tested that one

Well, the nice thing about that is that cairo or agg printing is irrelevant, as long as we have means to store painting sequence (into Drawing like object) -> we can print Images already, means we can band and render any advanced graphics quite easily too.

BTW, it would be nice to do some benchmark comparisons, now that we have both:)

Mirek

Subject: Re: Cairo
Posted by kodos on Wed, 07 Jan 2009 08:14:12 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 07 January 2009 08:14
BTW, it would be nice to do some benchmark comparisons, now that we have both:)

Yes, sure I'll rewrite the benchmarks in cairo if you give me some from AGG

BTW: the CairoDemo in the zip is already a pretty good benchmark

---

Subject: Re: Cairo
Posted by mirek on Wed, 07 Jan 2009 10:36:18 GMT
View Forum Message <> Reply to Message

For now, the startpoint is uppdev/SDrawTest

I guess interface are quite similar - it should not be any problem for you. The only problem being, SDraw is not really finished yet

Mirek

Subject: Re: Cairo
Posted by mirek on Wed, 07 Jan 2009 10:38:21 GMT
View Forum Message <> Reply to Message

CiaroDemo in Win32:

Linking...
   Creating library u:\out\MSC9.Debug.Debug_full.Gui\CairoDemo.lib and object u:\out\MSC9.Debug.Debug_full.Gui\CairoDemo.exp
cairolib.obj : error LNK2019: unresolved external symbol _png_set_filler referenced in function _write_png
cairolib.obj : error LNK2019: unresolved external symbol _png_set_write_user_transform_fn referenced in function _write_png
cairolib.obj : error LNK2019: unresolved external symbol _png_set_packswap referenced in function _write_png
cairolib.obj : error LNK2019: unresolved external symbol _png_set_read_user_transform_fn referenced in function _read_png
cairolib.obj : error LNK2019: unresolved external symbol _png_set_gray_to_rgb referenced in function _read_png
cairolib.obj : error LNK2019: unresolved external symbol _png_set_packing referenced in function _read_png
cairolib.obj : error LNK2019: unresolved external symbol _png_set_tRNS_to_alpha referenced in function _read_png
cairolib.obj : error LNK2019: unresolved external symbol _png_set_expand_gray_1_2_4_to_8 referenced in function _read_png
cairolib.obj : error LNK2019: unresolved external symbol _png_set_palette_to_rgb referenced in function _read_png
u:\out\MSC9.Debug.Debug_full.Gui\CairoDemo.exe : fatal error LNK1120: 9 unresolved externals

Mirek

Subject: Re: Cairo
Posted by mirek on Wed, 07 Jan 2009 10:38:59 GMT
View Forum Message <> Reply to Message

(which reminds me that SDraw does not work in Linux now - needs minor text rendering related fix).

Subject: Re: Cairo
Posted by kodos on Wed, 07 Jan 2009 11:06:20 GMT
View Forum Message <> Reply to Message

Hm, did you update your png.upp file? If yes I think you have to manually rebuild the png package because TheIDE doesn't recognize the change.

Quote:
(which reminds me that SDraw does not work in Linux now - needs minor text rendering related fix).

Text rendering is also the problematic part of cairo, currently there is just support for the "toy api" in my implementation. YOu shouldn't really use it in real projects. The best option would be pango, but that would be again another library and the license is probably not that great for U++. LGPL.

Subject: Re: Cairo
Posted by mirek on Wed, 07 Jan 2009 11:30:03 GMT
View Forum Message <> Reply to Message

kodos wrote on Wed, 07 January 2009 06:06
Text rendering is also the problematic part of cairo, currently there is just support for the "toy api" in my implementation. YOu shouldn't really use it in real projects. The best option would be pango, but that would be again another library and the license is probably not that great for U++. LGPL.

In fact, while not everything text-related is implemented yet, it turned out to be surprisingly simple to support current U++ text/font capabilites in SDraw.

Mirek

Subject: Re: Cairo
Posted by kodos on Wed, 07 Jan 2009 11:44:04 GMT
View Forum Message <> Reply to Message

I have to admit that I'm not really into all the text rendering stuff. So I'm not sure how hard it would be to implement the U++ text implementation in cairo.

Subject: Re: Cairo
Posted by kodos on Tue, 13 Jan 2009 14:38:12 GMT
View Forum Message <> Reply to Message

Here is a package with the lion rendering and the Pythagoras tree with SDraw and cairo. Although the SDraw Pythagoras tree doesn't work and I don't know what's wrong. Somehow the rotation isn't applied.

File Attachments
1) DrawBench.zip, downloaded 448 times

Subject: Re: Cairo
Posted by mirek on Tue, 13 Jan 2009 16:53:55 GMT
View Forum Message <> Reply to Message

The problem is that you cannot apply (at least in current SDraw) the path transformation during

path definition. And you had that Rotate after Move (Move starts path, any Fill or Stroke or Clip ends it).

Fixed:

```
void DoRect(SDraw &sw, double size)
{
 if(size < 0.5)
  return;

 sw.Move(0, 0);
 sw.Line(0, size);
 sw.Line(size, size);
 sw.Line(size, 0);
 sw.Fill(Blue());

 sw.Begin();
 sw.Translate(0, size);
 sw.Rotate(M_PI/4.);
 DoRect(sw, size / M_SQRT2);
 sw.End();

 sw.Begin();
 sw.Translate(size / 2, 1.5 * size);
 sw.Rotate(-M_PI/4.);
 DoRect(sw, size / M_SQRT2);
 sw.End();
}

void DrawPythagorasTree(Size sz, SDraw *sw, Cairo *ca)
{
 double size = 128;

 if (sw)
 {
  sw->Begin();
  sw->Translate(sz.cx / 2 - size / 2, sz.cy);
  sw->Scale(1, -1);

  DoRect(*sw, size);
  sw->End();
 }
```

Mirek

Subject: Re: Cairo
Posted by mirek on Tue, 13 Jan 2009 16:55:49 GMT
View Forum Message <> Reply to Message

Well, these preliminary results look promising:

TIMING Cairo lion : 156.00 ms - 7.80 ms (156.00 ms / 20), min: 7.00 ms, max: 9.00 ms, nesting: 1 - 20
TIMING Cairo tree : 2.85 s - 142.70 ms (2.85 s / 20), min: 139.00 ms, max: 146.00 ms, nesting: 1 - 20
TIMING AGGUPP lion : 46.00 ms - 2.30 ms (46.00 ms / 20), min: 1.00 ms, max: 3.00 ms, nesting: 1 - 20
TIMING AGGUPP tree : 627.00 ms - 31.35 ms (627.00 ms / 20), min: 29.00 ms, max: 34.00 ms, nesting: 1 - 20

I have heard AGG is fast. It really is

Mirek

EDIT: I have noticed that in tree, I left AGG to go up to size 0.5, so it was at disadvantage. With the same requirements, it is much faster than that.. (above numbers are updated to size < 1).

Subject: Re: Cairo
Posted by kodos on Tue, 13 Jan 2009 17:04:22 GMT
View Forum Message <> Reply to Message

Ah thanks for the fix.

Yes AGG is faster. But I still like cairo better because it supports other surfaces (PDF, PS, SVG, Glitz) and it is actively developed. As I understand the current situation with AGG we are stuck with version 2.4.

Subject: Re: Cairo
Posted by mirek on Tue, 13 Jan 2009 17:55:56 GMT
View Forum Message <> Reply to Message

Well, what we need is rendering engine for ImageBuffer.

SVG, PDF surfaces are relatively simple and planned (PDF as part of current PdfDraw)

BTW, I am trying to model the SDraw interface to be close to SVG requirements - that should make export/import quite simple.

As for active development, it slowly becomes "agg inspired" more and more:) And 2.4 license is

perfect for our needs.

BTW, one significant problem with cairo is that (AFAIK) its .so requires X11. Showstopper for web applications.

Mirek

Subject: Re: Cairo
Posted by kodos on Tue, 13 Jan 2009 19:18:40 GMT
View Forum Message <> Reply to Message

I think cairo can also be used on a headless server, by just compiling the image surface.

But I have no problem to maintain the cairo package for myself. It suits my needs very well, and is already quite tightly integrated in my current program
I just have to implement some text rendering, probably with the help of the SDraw source

---

Subject: Re: Cairo
Posted by mirek on Tue, 13 Jan 2009 21:54:55 GMT
View Forum Message <> Reply to Message

kodos wrote on Tue, 13 January 2009 14:18I think cairo can also be used on a headless server, by just compiling the image surface.

But I have no problem to maintain the cairo package for myself. It suits my needs very well, and is already quite tightly integrated in my current program
I just have to implement some text rendering, probably with the help of the SDraw source

Sure. In fact, it should be possible to implement SDraw (well, it will rather be renamed to "Painter") with cairo background quite simply.

Mirek
