
Subject: AggCtrl

Posted by [dolik.rce](#) on Mon, 22 Sep 2008 12:39:53 GMT

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

Hello everyody!

I needed some fast and pretty graphics for my project, so I decided to write a Ctrl that would allow me to use Agg (<http://www.antigrain.com>) in Upp. After studying what was done in this topic before (DrawAgg and some other packages by Aris - gave me a lot of inspiration, thanks) and bit of fighting with Xlib and Agg libraries, I finnaly got working version.

It currently runs only on X11, but I plan to make win32 too (or to find some volunteer to do it). It also have some bugs concerning pixel formats (documented in source), but I hope to get rid of them soon. Most of the code is taken from original file agg_platform_support.cpp from Agg, just rewritten to work with Ctrl derived class.

The basic idea is that the Ctrl has it's own agg::rendering_buffer and everything is set up, so the only thing user have to do is to define member

void AggCtrl::onDraw() and put there all the code to draw into the Ctrl's buffer.

I upload it here with working example and I'll be very glad for any feedback...

Bye dolik.rce

Update: I've fixed the uninitialized variables (see next post) and uploaded the new files here...

Subject: Re: AggCtrl

Posted by [dolik.rce](#) on Mon, 22 Sep 2008 20:40:30 GMT

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

Hi,

I've just found that the code I posted above doesn't work properly in Release mode. Maybe some of you already found it too... I've tried to find different settings and here's how it worked:

GCC.Blitz.Gui.Shared: Not working

GCC.Debug_full.Gui.Release.Shared: Not working

GCC.Gui.Shared: Not working

GCC.Debug.Debug_full.Gui.Noblitz.Shared: OK

GCC.Debug.Debug_full.Gui.Shared: OK

For short: anything compiled in release mode crashed with "Out of memory" or "Invalid memory access" or tried to eat up all the memory on my box (got about 300MB in RAM + 800MB in swap before I killed it) In Debug mode everything works fine. I guess it must be some memory checks which are active only in release mode...

I have absolutly no idea what I do wrong I even tried to set full debug in release mode and run the app in debugger, but with no usefull results (btw: do the optimizations spoil debugging info?)

Could somebody give me a hint, please?

Thanks...

Update:

Never mind this message please... I've just found 2 uninitialized variables in constructor, which were causing the troubles

Subject: Re: AggCtrl
Posted by [mirek](#) on Tue, 23 Sep 2008 15:40:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Sounds pretty good. Anyway, I believe this belongs to "Bazaar"...

Mirek

Subject: Re: AggCtrl
Posted by [dolik.rce](#) on Tue, 23 Sep 2008 16:25:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Tue, 23 September 2008 17:40 Sounds pretty good. Anyway, I believe this belongs to "Bazaar"...

I hope it'll be good, I need it for my app Sorry for posting it on wrong place, I'm still not too confident about structure of this forum...

Anyway, I fixed the bug with non native pixel formats, it was just a stupid typo Some changes were made to the interface of this control, now user can overload three functions:

virtual void onDraw(Upp::Draw& dest);

virtual void onInit();

virtual void onResize(int width,int height);I think their meaning is clear enough...

The last change is in the example - draw a line is too easy with any lib or toolkit, so I put there vector drawing of a lion, which is used in many examples on Agg official site...

I guess now, it would be the time to make a win32 version...

And by the way, I'm still waiting for any feedback, I've tested it only on my xubuntu box...

Bye...

Subject: Re: AggCtrl
Posted by [dolik.rce](#) on Tue, 23 Sep 2008 21:49:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello everebody!

I hope this doesn't look like spamming But I've already finnished the win32 version of AggCtrl, so I'll put it here. It took much less effort than I expected. But please don't ask me how it works, I'm

not too sure

Since last post there's only few changes in the agg24 package and of course the win32 implementation.

I hope that now, when there's much more potential testers (I'm aware of the terrible fact that linux has only about 0,9% among the OS's) somebody will give me any report how it works or at least some criticism

File Attachments

1) [AggCtrl.tar.gz](#), downloaded 373 times

Subject: Re: AggCtrl

Posted by [Oblivion](#) on Tue, 23 Sep 2008 22:40:49 GMT

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

Hi,

Good job! I compiled the aggtest on Win32 (XP pro SP3), and it worked very well. The only thing is, the code refused to compile at the first time (on both MSVC9 and MinGW). Reason was, "if with no expression" at aggctrl.h lines 25 and 58:

```
#elif PLATFORM_WIN32
```

but after changing this two line to:

```
#elif defined(PLATFORM_WIN32)
```

code compiled successfully.

Regards.

Subject: Re: AggCtrl

Posted by [mirek](#) on Wed, 24 Sep 2008 08:32:07 GMT

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

Some initial comments:

- great job

- 370KB to upload here is quite lot - well, it is not really a problem if uploaded once, but I have deleted previous versions. Please, do so too. We have about 300MB for the forum now... Of course, this deserves to be moved into Bazaar anyway

- I think that you should try to "embed" agg24 into AggCtrl; maybe use "plugin/png" method (one file that includes all .cpp) to avoid package list to be too old. Or just put it into subdir...

- Now a little bit more design related issue:

I believe that correct "final solution" of this should be lower-level. I mean, we should definitely aim for ImageBuffer rather than full Ctrl.

One of fundamental advantages is that in future, drawing operations should be independent of GUI. Think about webserver that have to run without X11 but still have to produce .png files...

(Sure, current Draw does not allow this, but that should change).

Also, hypothetical AggDraw should reimplement all normal Draw methods (like DrawRect). The goal here is to allow existing code using Draw to be used with new renderer.

Mirek

Subject: Re: AggCtrl

Posted by [dolik.rce](#) on Wed, 24 Sep 2008 09:54:56 GMT

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

luzr wrote on Wed, 24 September 2008 10:32 Some initial comments:

- great job

Thanks

luzr wrote on Wed, 24 September 2008 10:32- 370KB to upload here is quite lot - well, it is not really a problem if uploaded once, but I have deleted previous versions. Please, do so too. We have about 300MB for the forum now... Of course, this deserves to be moved into Bazaar anyway OK, I promise to keep my uploads low next time. By moving to Bazaar you mean to the svn folder?

luzr wrote on Wed, 24 September 2008 10:32- I think that you should try to "embed" agg24 into AggCtrl; maybe use "plugin/png" method (one file that includes all .cpp) to avoid package list to be too old. Or just put it into subdir...

There are all the files included through agg24/agg24.h, but I guess it's not exactly what you mean... The problem is, that agg is highly modular - I wanted to give the user a choice to load either all files (with agg24.h) or just those headers that he actually needs. Now as I think of it, it's probably not that important, compiler should not include code which is not used, right? The second (and more important) reason to have all headers in this package is to let Assist++ to read them, and provide user all the features like code completion etc. I'm not too skilled in aggyet, so I use those features a lot to find the functions I need... The only change to agg24 package I would do is to put there separators and organize the code into sections like renderers, rasterizers and so on (same as Aris had it in his plugin/agg24 - I could maybe even use his package with some minor changes)

luzr wrote on Wed, 24 September 2008 10:32- Now a little bit more design related issue:

I believe that correct "final solution" of this should be lower-level. I mean, we should definitely aim

for ImageBuffer rather than full Ctrl.

One of fundamental advantages is that in future, drawing operations should be independent of GUI. Think about webserver that have to run without X11 but still have to produce .png files...

(Sure, current Draw does not allow this, but that should change).

Also, hypothetical AggDraw should reimplement all normal Draw methods (like DrawRect). The goal here is to allow existing code using Draw to be used with new renderer.

Mirek

I'm aware of the plans you have with Draw and Agg. I wrote this mostly as quick "patch" to get the functionality I need for my project (fast, platform independent rendering with antialiasing and easy alphablending). This is far from the "final solution", although I think it should not be too complicated to rewrite it to use ImageBuffer - AFAIK Aris already did it. I don't know much about the drawing internals of U++ (yet), but if I could convert the agg rendering_buffer to Ximage and win32 pixmap, I think it should be possible as well for any other image storage format - that means ImageBuffer or directly to png, bmp, etc...

The only problem would be (as usually) with printing. I read all the discussions here about printing and agg, but I can't think about anything else than you - just draw a huge bitmap and break it into rectangles... Btw: In future I may have to write postscript output for the pictures rendered by agg (sounds crazy, I know...), would that be of any help with the printing?

Oblivion wrote on Wed, 24 September 2008 00:40 Reason was, "if with no expression" at aggctrl.h lines 25 and 58:

```
#elif PLATFORM_WIN32
```

Thanks for feedback. This was the only change I made after shutting down Virtualbox with my WinXP. It worked in GCC this way and I supposed that all the preprocessors work the same... So I learned something new today

Bye
Honza

Subject: Re: AggCtrl

Posted by [mirek](#) on Wed, 24 Sep 2008 10:30:57 GMT

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

dolik.rce wrote on Wed, 24 September 2008 05:54

The only problem would be (as usually) with printing. I read all the discussions here about printing and agg, but I can't think about anything else than you - just draw a huge bitmap and break it into rectangles...

Actually, the plan was to store the drawing into some sort of recorder (like Drawing), then draw it many times, clipping out rectangles needed. The advantage is much less memory required.. (A4 page would otherwise be 72MB image...).

Quote:

Btw: In future I may have to write postscript output for the pictures rendered by agg (sounds crazy, I know...), would that be of any help with the printing?

Hard to say at this moment.

OTOH, AggDrawing will require AGG interface encapsulation, which in turn will make this much easier...

Mirek
