Subject: Raster Control
Posted by mdelfede on Thu, 20 Mar 2008 20:58:42 GMT
View Forum Message <> Reply to Message

I've created an optimized raster control, usable to view any kind of raster image.
Capabilities :

- Middle mouse button press -- image PAN
- Mouse Wheel -- scroll up/down
- Shift + mouse Wheel -- zoom in/out
- Multi paage raster view with optional thumbnails
- quite fast zoom/pans operations even for big images

The control has been developed mainly a viewer for a fax application, but can be useful to other purposes.
Having optimized ring-buffer scrolling, it's not suited to build an image editor; replacing ImageCache stuff with a trimmed version without ring-buffer can make suitable also for editing.

Code is in Bazaar svn repository, starting from SVN 190.

Here a screenshot :

Ciao

Max

File Attachments
1) RasterTest.jpg, downloaded 1076 times

Subject: Re: Raster Control
Posted by mrjt on Tue, 01 Apr 2008 09:22:50 GMT
View Forum Message <> Reply to Message

I've just run RasterCtrlTest and got the following compilation errors (very easy to fix obviously):
c:\uppsvn\bazaar\rasterctrl\rasterctrl.cpp(79) : error C4716: 'RasterCtrl::Open' : must return a value
c:\uppsvn\bazaar\rasterctrl\rasterctrl.cpp(110) : error C4716: 'RasterCtrl::ShowThumbnails' : must return a value

It also seems to crash when loading a single image smaller than the window, as Layout gets called before imageScale is set (I believe).

Subject: Re: Raster Control
Posted by mdelfede on Tue, 01 Apr 2008 12:09:35 GMT
View Forum Message <> Reply to Message

mrjt wrote on Tue, 01 April 2008 11:22I've just run RasterCtrlTest and got the following compilation errors (very easy to fix obviously):
c:\uppsvn\bazaar\rasterctrl\rasterctrl.cpp(79) : error C4716: 'RasterCtrl::Open' : must return a value
c:\uppsvn\bazaar\rasterctrl\rasterctrl.cpp(110) : error C4716: 'RasterCtrl::ShowThumbnails' : must return a value

yes, sorry... the error checking was not finished... I'll update svn today.
BTW, I guess we should enable strict warnings on GCC... mine didn't notice the error.

Quote:
It also seems to crash when loading a single image smaller than the window, as Layout gets called before imageScale is set (I believe).

mhhh... I can't reproduce that. Can you tell me how to do it ? I tested with small icons and all seems ok.

Max

Subject: Re: Raster Control
Posted by mrjt on Tue, 01 Apr 2008 13:11:19 GMT
View Forum Message <> Reply to Message

I'm opening an image that doesn't require thumbnails to be shown and is smaller than the window. This causes 3/4 divide by ZERO errors for various reasons. I appreciate that this is kind of the opposite to how the control is supposed to be used though.

- In RasterThumbsCtrl::CalcScale:

```
int RasterThumbsCtrl::CalcScale(int imScale, int rasterWidth, int maxPageHeight)
{
 // calculates scale factor based on max thumb width
 // slightly smaller than ctrl width
 int maxScaledWidth = iscale(GetSize().cx, THUMBS_HSIZE_MUL, THUMBS_HSIZE_DIV);
 return iscale(maxScaledWidth, 1000, rasterWidth);

} // END RasterThumbsCtrl::CalcScale()
```

With no thumbnails GetSize().cx is 0 since you have previously set the splitter pos to 0. imageScale is then set to 0 and causes a DIVIDE_BY_ZERO. Adding the following after the call to CalcSize in RasterBaseCtrl::Layout

```
 if (!imageScale) {
  inside = false;
  return;
```

}fixes it.

- Also RasterBaseCtrl::Layout. You need !0 checks for vScollMax and hScrollMax on the two lines that do:
hScrollBar.Set(iscale(hScrollPos, scaledRasterWidth, hScrollMax)); and the vertical equivalent. I don't really understand what's going on here, but I can't see how this could ever work since vScrollMax should always start as 0.

- RasterBaseCtrl::PaintCache. Add a check for imageScale == 0 at the start of the function.

- ImageCache::Fill, caused when you resize the window down to nothing. Change first line to:
if(imageBuffer.IsEmpty() || !imageBuffer.GetSize().cx || !imageBuffer.GetSize().cy)

Then everything works very nicely .

You also may want to investigate Upp::Scroller. Not sure if this'll help in your case but it's worth a look.

James

Subject: Re: Raster Control
Posted by mdelfede on Tue, 01 Apr 2008 13:38:10 GMT
View Forum Message <> Reply to Message

Ok, thanx for suggestions !
I didn't notice the divide by 0 errors... maybe I've got them but didn't make troubles on running demo.

About scrolling, my control repaints inside cache ONLY the scrolled in part of image, that's the speed up stuff.
BUT, it repaints the whole cache on ctrl on Paint() routine.... I don't know if it's worth the effort to repaint only neede areas... For my use, the control is fast enough, and IMO checking to see what needs to be repaint from cache to ctrl is a quite big work.

I'll correct bugs later this nigh on svn. Thanx again for signaling them!

BTW.... what's your use for my control ? or was it just for testing ?

Ciao

Max

Subject: Re: Raster Control
Posted by mrjt on Tue, 01 Apr 2008 14:13:56 GMT

View Forum Message <> Reply to Message

I was just curious. It a very nice control though

Subject: Re: Raster Control
Posted by mdelfede on Tue, 01 Apr 2008 16:52:34 GMT
View Forum Message <> Reply to Message

I still don't get divide by 0 errors.... I've tested with many small images, icons and so on, but nothing.
How can you reproduce the problem ?

Max

Subject: Re: Raster Control
Posted by mrjt on Tue, 01 Apr 2008 17:08:23 GMT
View Forum Message <> Reply to Message

Interesting. If you compile with MingW, and presumably GCC as well, you don't get the error. If you look at iscale:

```
int iscale(int x, int y, int z)
{
#ifdef __NOASSEMBLY__
 return int(x * (double)y / z);
#else
 __asm
 {
  mov  eax, [x]
  imul [y]
  idiv [z]
 }
#endif
}
```

When using MingW __NOASSEMBLY__ is defined, with MSVC it's not. And only the assembler crashes, even though there is still a clear divide by 0. This really should be fixed.

Subject: Re: Raster Control
Posted by mdelfede on Tue, 01 Apr 2008 19:05:35 GMT
View Forum Message <> Reply to Message

mrjt wrote on Tue, 01 April 2008 19:08
When using MingW __NOASSEMBLY__ is defined, with MSVC it's not....

Quite interesting stuff, yes... I think fp math can cope with 0/0 problem, using NANs, maybe.
Well, how can I tell GCC to undefine _NOASSEMBLY_ without patching upp code ?

Max

EDIT : well, I've seen why NOASSEMBLY is defined for GCC... asm syntax is different from MSC to GCC.
IMHO the best way would be to do some alternate assembly code instead of dropping it in GCC....
For example, along with NOASSEMBLY we should define a GCC_ASSEMBLY and MSC_ASSEMBLY and react in consequence.
For now I'll make a small patch on my 'mathutils.cpp' stuff...

Max

Subject: Re: Raster Control
Posted by mr_ped on Tue, 01 Apr 2008 19:57:24 GMT
View Forum Message <> Reply to Message

Not only it is divided into C/ASM, but both code paths are different.

The ASM version works purely with integers, while the C version with (double) will lead to decimal division at least, if not multiply (I'm not that big C guru to fully understand how that casting to double will propagate to the rest of calculations).

To get maybe identical version in C you may try:

return ((x * y) / z);

I'm not sure it will compile absolutely identically, but I think the chances are very good... feel free to compile into ASM firstly and check the result of GCC ASM (-S command line switch?).

Subject: Re: Raster Control
Posted by mdelfede on Tue, 01 Apr 2008 20:14:47 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Tue, 01 April 2008 21:57Not only it is divided into C/ASM, but both code paths are different.

The ASM version works purely with integers, while the C version with (double) will lead to decimal division at least, if not multiply (I'm not that big C guru to fully understand how that casting to double will propagate to the rest of calculations).

To get maybe identical version in C you may try:

return ((x * y) / z);

I'm not sure it will compile absolutely identically, but I think the chances are very good... feel free to compile into ASM firstly and check the result of GCC ASM (-S command line switch?).

Well, maybe will be the same, maybe not.
The problem solved by Iscale is rounding working with integers.
To have no precision loss, you should have a temporary integer with twice the integer width, whichever it is... OR use some MULDIV advanced algorithm.
Because of that some assembly language has a built-in muldiv instruction.
The big problem is that in C you've no guarantee of integer sizes, you just know that :
char <= short <= int <= longint <= longlongint
so, you can't know what size to use in intermediate calculation.... because of that in Iscale they use double.
In GCC32 you should have int(32), long(32) and long long(64), so using longlong would be ok.... but I guess that in GCC64 the int is 64 byte wide (maybe), so it won't work either.
BTW, nor the asm code will work without some cheks.

IMHO, the right approach would be to typedef some integer sizes in Core.h :

typedef char int8
typedef short int16
typedef int int32
typedef longlong int64

depending on machine, then use

int32 Iscale(int32 a, int32 b, int32 c)

so it's compiler sake to give needed warnings.

Max

typedef

Subject: Re: Raster Control
Posted by mdelfede on Tue, 01 Apr 2008 20:59:20 GMT
View Forum Message <> Reply to Message

Well, corrected in SVN, now it should work ok on MSC too.

BTW, in the meantime, I've made a patched and (IMHO) better version of Iscale functions, I'll post them in another thread.

Ciao

Max

Subject: Re: Raster Control
Posted by mr_ped on Tue, 08 Apr 2008 15:34:12 GMT
View Forum Message <> Reply to Message

I'm trying it from 2008.1beta2 installation, and I hit crash right after opening first image I found on my disc.

It turned out the very first image I found was gif, and you didn't add plugin/gif package, so the RasterCtrl\RasterCtrl.cpp:55 line
raster = StreamRaster::OpenAny(imageStream);
ends with NULL assigned to raster.

Than the next for loop
 for(curPage = 0; curPage < raster->GetPageCount(); curPage++)
will crash due to referencing null pointer.

I think there should be some error handler instead of crash in case the input stream did not result into valid raster object.

Another funny thing I encountered was opening that gif finally (after I did add plugin/gif package), it fits the width of window (it's scaled up a bit) with vertical scrollbar, than I used View / Zoom 10%, than View / Page width, and I do get both vertical and horizontal scrollbars suddenly. After using View / Page width second time, the horizontal scrollbars is gone and I have the "default" view just like after opening the image.

Subject: Re: Raster Control
Posted by mdelfede on Wed, 09 Apr 2008 08:28:04 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Tue, 08 April 2008 17:34I'm trying it from 2008.1beta2 installation, and I hit crash right after opening first image I found on my disc.

well, 2008.1beta2 has an outdated version, anyways... bugs pointed out by mrjt (which caused crash on MSC) are corrected in svn. Please try it, just pick Bazaar from svn, if you don't like to work with upp-svn package.
BTW I stopped upgrading svn packages for the moment cause some lettercase mismatches in uvs2 that brings some problems in linux.

Quote:
It turned out the very first image I found was gif, and you didn't add plugin/gif package, so the RasterCtrl\RasterCtrl.cpp:55 line
raster = StreamRaster::OpenAny(imageStream);

ends with NULL assigned to raster.

Than the next for loop
 for(curPage = 0; curPage < raster->GetPageCount(); curPage++)
will crash due to referencing null pointer.

I think there should be some error handler instead of crash in case the input stream did not result into valid raster object.

I agree with you, the package isn't yet refined, in particular to error handling. I'll add some error checks soon.... sorry for the inconvenience but I made the package for a particular need, not yet with production quality in mind
About the gif plugin, I'm not sure if I should add it to RasterControl package or leave it to users if they need it... But that apply to tiff too. IMHO would be better to leave grafic formats out of rastercontrol package and let users include just what they need.

Quote:
Another funny thing I encountered was opening that gif finally (after I did add plugin/gif package), it fits the width of window (it's scaled up a bit) with vertical scrollbar, than I used View / Zoom 10%, than View / Page width, and I do get both vertical and horizontal scrollbars suddenly. After using View / Page width second time, the horizontal scrollbars is gone and I have the "default" view just like after opening the image.

uh ? I didn't notice that behaviour.... it should hide scrollbars when not needed. A known problem is that it can loose the space used by scrollbars (it makes it necessary to calculate the sizes both for control with and without scrollbars to solve it, it's planned but I'm looking for a good (and quick) solution.
But scrollbars hiding should work... can you post a sample image ?

Finally, I'm now working to the hard(ware) part of my fax application, the modem/serial port/threads stuffs, which is quite complicated. When it'll ok, I'll join together with RasterControl so I'll test it much more in depth....

Max

Subject: Re: Raster Control
Posted by mr_ped on Wed, 09 Apr 2008 08:59:17 GMT
View Forum Message <> Reply to Message

Here's the image I was testing with.
(a wallpaper, so I believe the copyright owner will not mind this post)

File Attachments
1) pvp37.gif, downloaded 430 times