Subject: More GLCtrl
Posted by kodos on Tue, 14 Oct 2008 16:51:32 GMT
View Forum Message <> Reply to Message

Hi,

I found again some problems in the GLCtrl. It seems as it isn't that easy to fix a problem in there without introducing a new one
So I now made a test case that should test most of the things. Also included is my extended GLCtrl class that features the functions GLInit, GLResize and GLDone on windows and an helper class to to use picking in OpenGL. But that shouldn't change the problems that you can see in the test case.

Here are the things that are currently broken:
-) Hide/Show does nothing on Windows Vista
-) The slider control doesn't refresh itself, if you move it on Windows Vista
-) The slider flickers on Linux, it looks like OpenGL wants to draw over the slider, but wouldn't it be better if frames would still be handled from U++ and the DHCtrl should fit inside the frames?
-) GLDone is not called on Linux after removing the control from the form. (This one is because the call should be made from BeforeTerminate from DHCtrl but this function is never called. The method that is used in the windows code also doesn't work because the State method is private on linux)

I hope the test case helps to find the problems
And I would be glad if the picking helper class would made it to U++, so I don't have to maintain it on my laptop and pc

File Attachments

1) OpenGl.zip, downloaded 537 times

Subject: Re: More GLCtrl
Posted by mirek on Fri, 17 Oct 2008 13:47:42 GMT
View Forum Message <> Reply to Message

kodos wrote on Tue, 14 October 2008 12:51Hi,

I found again some problems in the GLCtrl. It seems as it isn't that easy to fix a problem in there without introducing a new one

To tell the truth, GLCtrl is a nightmare for me - mostly because I am not that good at maintaining stuff that I do not actually use... (And I do not use Vista as well...).

Mirek

Subject: Re: More GLCtrl
Posted by kodos on Fri, 17 Oct 2008 14:24:13 GMT
View Forum Message <> Reply to Message

I understand that.

I think the biggest problem right now is that the DHCtrl (or more accurately the native control inside DHCtrl) is using the whole space of the DHCtrl, and the frames try to draw over it. I don't think this is the right behavior because as I understand it the area that the frames use shouldn't be usable from the control itself.
So the best solution imho would be if the "native control" inside the DHCtrl would be resized with respect to the frames. But I don't know how complicated this change would be. I am a bit lost in all the layout code

Subject: Re: More GLCtrl
Posted by mirek on Fri, 17 Oct 2008 15:04:01 GMT
View Forum Message <> Reply to Message

kodos wrote on Fri, 17 October 2008 10:24I understand that.

I think the biggest problem right now is that the DHCtrl (or more accurately the native control inside DHCtrl) is using the whole space of the DHCtrl, and the frames try to draw over it. I don't think this is the right behavior because as I understand it the area that the frames use shouldn't be usable from the control itself.
So the best solution imho would be if the "native control" inside the DHCtrl would be resized with respect to the frames. But I don't know how complicated this change would be. I am a bit lost in all the layout code

Yes, you are perhaps right...

Mirek

Subject: Re: More GLCtrl
Posted by mirek on Sun, 19 Oct 2008 16:20:03 GMT
View Forum Message <> Reply to Message

I believe all should be fix now.

Note: DHCtrl was never intended to have children... Therefore, rather than rewriting half of CtrlCore, I have solved the issue by restructruing GLCtrl - it has now DHCtrl as child (and derives from ParentCtrl).

Thanks for hint and a quality testcase - I have put it to upptst for further use

Mirek

Subject: Re: More GLCtrl
Posted by kodos on Sun, 19 Oct 2008 16:53:53 GMT
View Forum Message <> Reply to Message

Ah, nice idea. I really didn't thought about that solution.
I will test it as soon as possible. Perhaps we could add an assertion somewhere that tests if somebody tries to add a frame to a DHCtrl.

Subject: Re: More GLCtrl
Posted by mdelfede on Mon, 20 Oct 2008 12:26:49 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 19 October 2008 18:20I believe all should be fix now.

Note: DHCtrl was never intended to have children... Therefore, rather than rewriting half of CtrlCore, I have solved the issue by restructruing GLCtrl - it has now DHCtrl as child (and derives from ParentCtrl).

Thanks for hint and a quality testcase - I have put it to upptst for further use

Mirek

Eh, do you remember when some monthes ago I told you CtrlCore would need some strong refactoring ?
Writing Linux GLCtrl was a nightmare because of the impossibility of having childs of windowed controls. I introduced them with some dirty hacks on Ctrl stuffs trying to not break previous behaviours, but just to Linux part.... and it was quite complicated because of optimizations in ctrlcore classes. And I know it's not a perfect implementation.....
I've seen some similar (but differently handled) hacks on windows DHCtrl code.

But I agree with you, it would be a *big* effort to clean up CtrlCore classes.

For other people not knowing the problem, OpenGL needs an underlying OS window (X11 or Windows). The problem arose because UPP controls don't have an underlying window (besides top ones) for speed (and other) purposes and were not foreseen to contain controls with a window handler.
All CtrlCore classes assume that no child control can have a window handle, so they use often the fact that there's no parent window to assume the control is a top one; in case of DHCtrl that's wrong. All hacks on linux part were aimed to remove this assumption.

Max

EDIT : btw, IMHO, the best would be to go in the direction on which child windowed controls are allowed. That's done on Linux part, it was not easy and it's not completely well implemented, but can be done. That would be a more uniform behaviour and would allow to embed some fancy windowed controls without the need of a wrapper class, and allowing them to be also containers for upp controls. (Linux DHCtrl can do almost all about it).

Max

Subject: Re: More GLCtrl
Posted by kodos on Mon, 20 Oct 2008 18:02:56 GMT
View Forum Message <> Reply to Message

Ok, I have tested the new code and it works great so far except that there are no more events on the GLCtrl itself

I have fixed the mouse events with the following code:

virtual Image MouseEvent(int event, Point p, int zdelta, dword keyflags) { return ctrl->MouseEvent(event, p, zdelta, keyflags); }

in the GLPane class. But I haven't found a function like that for the key events, and I don't know if I broke something else

---

Subject: Re: More GLCtrl
Posted by mirek on Mon, 20 Oct 2008 18:51:44 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Mon, 20 October 2008 08:26luzr wrote on Sun, 19 October 2008 18:20I believe all should be fix now.

Note: DHCtrl was never intended to have children... Therefore, rather than rewriting half of CtrlCore, I have solved the issue by restructruing GLCtrl - it has now DHCtrl as child (and derives from ParentCtrl).

Thanks for hint and a quality testcase - I have put it to upptst for further use

Mirek

Eh, do you remember when some monthes ago I told you CtrlCore would need some strong refactoring ?
Writing Linux GLCtrl was a nightmare because of the impossibility of having childs of windowed controls.

IMO you do not really appreciate nightmares if it would actually be directly possible....

Quote:
But I agree with you, it would be a *big* effort to clean up CtrlCore classes.

IMO, worst thing is that it would be *wrong* effort.

Most of complication in CtrlCore are caused by necessary "impedance mismatch" between U++ and underlaying host system API. "clean up" would most likely produced code that is nice, but does not work. Making it work would then bring us to the current status again...

Mirek

Subject: Re: More GLCtrl
Posted by mirek on Mon, 20 Oct 2008 19:01:14 GMT
View Forum Message <> Reply to Message

kodos wrote on Mon, 20 October 2008 14:02Ok, I have tested the new code and it works great so far except that there are no more events on the GLCtrl itself

I have fixed the mouse events with the following code:

virtual Image MouseEvent(int event, Point p, int zdelta, dword keyflags) { return ctrl->MouseEvent(event, p, zdelta, keyflags); }

in the GLPane class. But I haven't found a function like that for the key events, and I don't know if I broke something else

Ops, correct. Hopefuly fixed (check soon). (You forget to translate for view offset, BTW...)

I think keyboard input can be effectively solved by adding "NoWantFocus" to GLPane constructor (there is no way how GLPane could get focus after that point...).

Mirek

Subject: Re: More GLCtrl
Posted by mdelfede on Mon, 20 Oct 2008 20:17:45 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 20 October 2008 20:51mdelfede wrote on Mon, 20 October 2008 08:26luzr wrote on Sun, 19 October 2008 18:20I believe all should be fix now.

Note: DHCtrl was never intended to have children... Therefore, rather than rewriting half of CtrlCore, I have solved the issue by restructruing GLCtrl - it has now DHCtrl as child (and derives from ParentCtrl).

Thanks for hint and a quality testcase - I have put it to upptst for further use

Mirek

Eh, do you remember when some monthes ago I told you CtrlCore would need some strong refactoring ?
Writing Linux GLCtrl was a nightmare because of the impossibility of having childs of windowed controls.

IMO you do not really appreciate nightmares if it would actually be directly possible....

Quote:
But I agree with you, it would be a *big* effort to clean up CtrlCore classes.

IMO, worst thing is that it would be *wrong* effort.

Most of complication in CtrlCore are caused by necessary "impedance mismatch" between U++ and underlaying host system API. "clean up" would most likely produced code that is nice, but does not work. Making it work would then bring us to the current status again...

Mirek

Well.... I'd not call it a "wrong" effort. It would make control behaviour more uniform between windowed and non-windowed controls, from the user's point of view.
BTW, by now it's not a big concern, being GLCtrl the only windowed child control in UPP... but it can change in future
IMHO to have the "correct" behaviour is not impossible, but it does indeed require some work, which maybe isn't worth the effort at the moment.

Max

Subject: Re: More GLCtrl
Posted by kodos on Tue, 04 Nov 2008 19:13:06 GMT
View Forum Message <> Reply to Message

Hi,

I have refactored the GLCtrl a little bit so that the platform specific code is just in the GLPane and not in GLCtrl class. I have also fixed a bug on Linux with the resize event not being fired.
While debugging this problem I have found that the methods AfterTerminate and Resize from the linux DHCtrl are never actually called. I think we should remove them.

File Attachments
1) GLCtrl.zip, downloaded 437 times

Subject: Re: More GLCtrl
Posted by mirek on Wed, 05 Nov 2008 07:53:42 GMT
View Forum Message <> Reply to Message

Thanks, applied.

Mirek

Subject: Re: More GLCtrl
Posted by cocob on Fri, 07 Nov 2008 21:43:34 GMT
View Forum Message <> Reply to Message

I have always some problems with GLCtrl on windows vista

Its ok on Linux32 Linux64 ans winXP
but on vista in all build mode (excepted debug) with MSC9 and MINGW
the refresh() call seems to not work properly.

Unfortunatly i didn't managed to create a small test-case.

cocob

Subject: Re: More GLCtrl
Posted by kodos on Fri, 07 Nov 2008 22:06:20 GMT
View Forum Message <> Reply to Message

Hm, I develop on Vista and I don't see a problem with refresh(). Are you sure you use the latest svn uppsrc?

Subject: Re: More GLCtrl
Posted by cocob on Sat, 08 Nov 2008 09:23:51 GMT
View Forum Message <> Reply to Message

Yes it is the last svn build (589). There is no problems for me on opengl reference exemple, but with my app recently build for the first time on vista. (no problems on other platforms or in vista in debug mode). How can i debug the refreshing process ?

Subject: Re: More GLCtrl
Posted by cocob on Sat, 08 Nov 2008 11:31:04 GMT
View Forum Message <> Reply to Message

it looks like a non-initialized variable (because it works in debug mode) But i can't find it...

The most surprising is that now it works with MINGW, but with MSC9 i have always the problem
cocob

Subject: Re: More GLCtrl
Posted by mirek on Sat, 08 Nov 2008 13:29:05 GMT
View Forum Message <> Reply to Message

cocob wrote on Fri, 07 November 2008 16:43I have always some problems with GLCtrl on windows vista

Its ok on Linux32 Linux64 ans winXP
but on vista in all build mode (excepted debug) with MSC9 and MINGW
the refresh() call seems to not work properly.

Unfortunatly i didn't managed to create a small test-case.

cocob

You can always create a huge one

Mirek

Subject: Re: More GLCtrl
Posted by cocob on Mon, 10 Nov 2008 14:02:55 GMT
View Forum Message <> Reply to Message

After some investigations, The bug is present on all windows plateform with all compilers. And it was working with an old version of GLCtrl. I will try to find the modification in svn

Subject: Re: More GLCtrl
Posted by cocob on Mon, 10 Nov 2008 14:48:22 GMT
View Forum Message <> Reply to Message

I have solved my problem by adding

virtual void Refresh() { pane.Refresh(); }

In GLCtrl class. But i don't know why this is necessary.

cocob

Subject: Re: More GLCtrl
Posted by kodos on Mon, 02 Feb 2009 18:11:54 GMT
View Forum Message <> Reply to Message

cocob wrote on Mon, 10 November 2008 15:48I have solved my problem by adding

virtual void Refresh() { pane.Refresh(); }

In GLCtrl class. But i don't know why this is necessary.

cocob

I think that fix is actually quite useful, as it resolves the last flickers on Vista for me

Subject: Re: More GLCtrl
Posted by mirek on Sun, 08 Feb 2009 07:00:19 GMT
View Forum Message <> Reply to Message

kodos wrote on Mon, 02 February 2009 13:11cocob wrote on Mon, 10 November 2008 15:48I have solved my problem by adding

virtual void Refresh() { pane.Refresh(); }

In GLCtrl class. But i don't know why this is necessary.

cocob

I think that fix is actually quite useful, as it resolves the last flickers on Vista for me

Well, the only trouble is that Refresh is not virtual...

Anyway, added that as non-virtual method.

Mirek