Subject: About DHCtrl and window handles...
Posted by mirek on Mon, 12 Nov 2007 19:39:35 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Mon, 12 November 2007 07:02luzr wrote on Mon, 12 November 2007 12:23
Well, the trouble is that this is (and has to be) platform specific.

Of course... but there is a thing that I don't like in Upp, and is that many platform specific methods are made public... I'd separate them from the public interface.

What is that supposed to solve?

BTW, there are people suggest to me to make ALL methods public and virtual...

Quote:
Ok, but I don't understand if EventProc is called on each control for all Windows or not.

Normally, only to top-level Ctrls (GetParent() == NULL).

Of course, existing DHCtrl recieves them too. So better to say, to all windows with handles, but handles for non-top-level windows are yet to be implemented in X11, that is the important part of the task...

Quote:
- Both A and B once, with A as parameter
- Only to A, as B don't need repaint

EventProc is already redirected to the window with given handle, if that answers your question....

Quote:
And, second question, how does MainWindos's EventProc dispatch the message to windowless controls ?

Nope. Dispatching to windowless ctrls is already done by platform independent code. Platform depended ends at distributing top-level events.

Quote:
And to windowed ones ?

There is a map in X11 platform specific code that maps handles to Ctrl *.

Quote:
Again an example :

A is a top level window, and B is a child of A, but with window handle too. What happens if I click on B ?
- B gets the event directly, A gets nothing

In Win32 DHCtrl, this is the case.

Quote:
And, last question (sorry, but it's a complicated matter...), another example :
 -A is a main window (with XWindow inside), B is a child of A (no matter if windowed or not...) and C is a child of B.

Hey, wait a moment. This is common misconception. Parentship is not the same thing as ownership.

All top-level widgets have no parents. Anyway, they can have owners.

Mirek

Subject: Re: About DHCtrl and window handles...
Posted by Novo on Mon, 12 Nov 2007 20:03:25 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 12 November 2007 14:39
BTW, there are people suggest to me to make ALL methods public and virtual...

That is the beginning of the end ...

Subject: Re: About DHCtrl and window handles...
Posted by mdelfede on Mon, 12 Nov 2007 20:53:15 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 12 November 2007 20:39mdelfede wrote on Mon, 12 November 2007 07:02luzr wrote on Mon, 12 November 2007 12:23
Well, the trouble is that this is (and has to be) platform specific.

Of course... but there is a thing that I don't like in Upp, and is that many platform specific methods are made public... I'd separate them from the public interface.

What is that supposed to solve?

Well, that should force derived controls to be platform-independent, if methods are made private, or at least that would force user code (not derived from Ctrl) to *not* use platform dependent stuffs.... IMHO those should be good reasons.
IMHO, core classes should incapsulate all platform dependent behaviour.

Quote:
BTW, there are people suggest to me to make ALL methods public and virtual...

Brrrrrrr !
So, if they don't bother of writing portable code, here's again the way :

#define private public
#include <all-Upp>

That's just for public, for 'virtual' I guess it's not so easy!

Quote:
Quote:
Ok, but I don't understand if EventProc is called on each control for all Windows or not.

Normally, only to top-level Ctrls (GetParent() == NULL).

Of course, existing DHCtrl recieves them too. So better to say, to all windows with handles, but handles for non-top-level windows are yet to be implemented in X11, that is the important part of the task...

So, as I'm trying to implement it (to some extents...), what I need to know is what happens now if I create an X11 handle on a child control, in respect with events.
I suppose X11 sends the event directly to window inside my control... so, my code *or* Upp has to have a way to handle the event, what is done usually in an event loop.
As you're keeping a list of X11 handles, I suppose that you've got a global event handler that dispatches in some way events to controls, right ? Because that's the central point... if Upp dispatches in some way the event, I must just catch it; if not, I must hook someway my code in your event loop... If I'm not making some big mistake, of course

Quote:
Quote:
- Both A and B once, with A as parameter
- Only to A, as B don't need repaint

EventProc is already redirected to the window with given handle, if that answers your question....

Not completely... I understand that your code redirects events to correct control, but what I don't know is if your code catches ALL events or only mainwindows events.

Better explained, if I create an X11 window that overlap a control, you're saying that my X11 window gets the event directly ? does UPP receive it also and dispatch it ?
Is maybe that one the purpose of having a global list of X11 windows handles ?
.....
Quote:
Quote:
And to windowed ones ?

There is a map in X11 platform specific code that maps handles to Ctrl *.

Ok, so I guess I did understand right... it's the Xwindow()[] map, right ? So, if I hook my handle inside it, my control gets Events as usual non-windowed controls ?

Quote:
Quote:
Again an example :
A is a top level window, and B is a child of A, but with window handle too. What happens if I click on B ?
- B gets the event directly, A gets nothing

In Win32 DHCtrl, this is the case.

So, I guess I'll have to duplicate this behaviour.

Quote:
Quote:
And, last question (sorry, but it's a complicated matter...), another example :
-A is a main window (with XWindow inside), B is a child of A (no matter if windowed or not...) and C is a child of B.

Hey, wait a moment. This is common misconception. Parentship is not the same thing as ownership.

All top-level widgets have no parents. Anyway, they can have owners.

Yes, I know the difference....
A child is usually an included control (or at least a dependent control), ownership is related to create/destroy it, not to the appearance nor to event handling.

Well, I'll try to go a bit further with my X11_DHCtrl

Ciao

Max

ops.... just another question :

Suppose this one :

MainWindow->A->B (-> represent Parent of)
B is windowed. I get the event... should I reflect to toplevel window ? I think that should the more consistent way to do the job...

Ciao

Max

Subject: Re: About DHCtrl and window handles...
Posted by mirek on Mon, 12 Nov 2007 22:21:36 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Mon, 12 November 2007 15:53luzr wrote on Mon, 12 November 2007 20:39mdelfede wrote on Mon, 12 November 2007 07:02luzr wrote on Mon, 12 November 2007 12:23
Well, the trouble is that this is (and has to be) platform specific.

Of course... but there is a thing that I don't like in Upp, and is that many platform specific methods are made public... I'd separate them from the public interface.

What is that supposed to solve?

Well, that should force derived controls to be platform-independent, if methods are made private, or at least that would force user code (not derived from Ctrl) to *not* use platform dependent stuffs.... IMHO those should be good reasons.
IMHO, core classes should incapsulate all platform dependent behaviour.

IMO, that is the bad way how to do things. I am pretty sure that "#ifdef PLATFORM_WIN32" is enough warning for anybody planning to use it, no need to stay in a way if you really need to do platform specific stuff.

Quote:
Ok, so I guess I did understand right... it's the Xwindow()[] map, right ? So, if I hook my handle inside it, my control gets Events as usual non-windowed controls ?

I suppose so, but frankly, I never really studied child windows in X11

In any case, X11 returns a target handle. That determines the widget whose EventProc is about to be called...

Quote:
Yes, I know the difference....
A child is usually an included control (or at least a dependent control), ownership is related to create/destroy it, not to the appearance nor to event handling.

Nope (or yes, but you are quoting another context and non-U++ teminology).

Owned window is a top-level window that has related owner.

For example, Find dialog in ide is owned by the main ide window. Such Find dialog is owned but really does not have parent (its GetParent is NULL).

Quote:
MainWindow->A->B (-> represent Parent of)
B is windowed. I get the event... should I reflect to toplevel window ? I think that should the more consistent way to do the job...

No.

Mirek

Subject: Re: About DHCtrl and window handles...
Posted by mdelfede on Mon, 12 Nov 2007 22:42:59 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 12 November 2007 23:21
Quote:
MainWindow->A->B (-> represent Parent of)
B is windowed. I get the event... should I reflect to toplevel window ? I think that should the more consistent way to do the job...

No.

Why not ?
Doesn't TopWindow take care of dispatching events to various Ctrl virtual funcs (like paint...) ?
The only other possibility I see is to reimplement all low level event dispatcher for my control...
seems to me a bit nonsense.... Where am I wrong ?

Subject: Re: About DHCtrl and window handles...
Posted by mirek on Mon, 12 Nov 2007 23:13:30 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Mon, 12 November 2007 17:42luzr wrote on Mon, 12 November 2007 23:21
Quote:
MainWindow->A->B (-> represent Parent of)
B is windowed. I get the event... should I reflect to toplevel window ? I think that should the more consistent way to do the job...

No.

Why not ?
Doesn't TopWindow take care of dispatching events to various Ctrl virtual funcs (like paint...) ?
The only other possibility I see is to reimplement all low level event dispatcher for my control... seems to me a bit nonsense.... Where am I wrong ?

Well, standard EventProc leaves that to U++. The idea is to override WindowProc / EventProc, do you hooks and then call Ctrl::WindowProc/EventProc...

Anyway, I guess your control will be of special kind anyway (like GLCtrl).

Mirek

Subject: Re: About DHCtrl and window handles...
Posted by mdelfede on Mon, 12 Nov 2007 23:47:55 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 13 November 2007 00:13

Anyway, I guess your control will be of special kind anyway (like GLCtrl).

Indeed But I think It's going to work now !

Ciao

Max

Subject: Re: About DHCtrl and window handles...

In order to avoid the need of make friends of GLCtrl all windowed x11 controls (and maybe it could serve for other purposes...)
I suggest to add these 2 protected members to GLCtrl :

```
////////////////////////////////////////////////////////////////////////////////
// Add subwindow to upp list of Xwindows
void Ctrl::AddUppXWindow(Window &w)
{
 int i = Xwindow().Find(None);
 if(i >= 0)
  Xwindow().SetKey(i, w);
 XWindow& cw = i >= 0 ? Xwindow()[i] : Xwindow().Add(w);
 cw.ctrl    = this;
 cw.exposed = true;
 cw.owner   = GetParent();
 cw.xic     = NULL;

} // END Ctrl::AddUppXWindow()


////////////////////////////////////////////////////////////////////////////////
// Removes subwindow to upp list of Xwindows
void Ctrl::RemoveUppXWindow(Window &w)
{
 int i = Xwindow().Find(w);
 if(i >= 0)
 {
  Xwindow().SetKey(i, None);
  Xwindow()[i].ctrl = NULL;
 }

} // END Ctrl::RemoveUppXWindow()
```

This code is duplicated both in X11 Topwindow AND in GLCtrl, and is coming in my windowed X11 control too...
BTW, after I finish my X11 DHCtrl, I guess it would be not bad to rewrite X11 GLCtrl part to be derived of it... I'm already doing it for testing purposes.

Ciao

Max

p.s.: I'm thinking again that the good behaviour for event handling should start from TopWindow parenting child controls and propagated to them. AFAIK now, as is DHCtrl done (and mine too...)

the parent of a windowed child control *don't* have any knowledge of events going to the child, as opposite as normal controls... maybe that can cause problems, I don't know yet.

Subject: Re: About DHCtrl and window handles...
Posted by mirek on Wed, 14 Nov 2007 09:47:19 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Tue, 13 November 2007 09:34In order to avoid the need of make friends of GLCtrl all windowed x11 controls (and maybe it could serve for other purposes...)
I suggest to add these 2 protected members to GLCtrl :

```
////////////////////////////////////////////////////////////////////////////////
// Add subwindow to upp list of Xwindows
void Ctrl::AddUppXWindow(Window &w)
{
 int i = Xwindow().Find(None);
 if(i >= 0)
  Xwindow().SetKey(i, w);
 XWindow& cw = i >= 0 ? Xwindow()[i] : Xwindow().Add(w);
 cw.ctrl    = this;
 cw.exposed = true;
 cw.owner   = GetParent();
 cw.xic     = NULL;

} // END Ctrl::AddUppXWindow()


////////////////////////////////////////////////////////////////////////////////
// Removes subwindow to upp list of Xwindows
void Ctrl::RemoveUppXWindow(Window &w)
{
 int i = Xwindow().Find(w);
 if(i >= 0)
 {
  Xwindow().SetKey(i, None);
  Xwindow()[i].ctrl = NULL;
 }

} // END Ctrl::RemoveUppXWindow()
```

This code is duplicated both in X11 Topwindow AND in GLCtrl, and is coming in my windowed X11 control too...

OK, np, good idea, adding that now.

Quote:
BTW, after I finish my X11 DHCtrl, I guess it would be not bad to rewrite X11 GLCtrl part to be derived of it... I'm already doing it for testing purposes.

Agree.

Quote:
p.s.: I'm thinking again that the good behaviour for event handling should start from TopWindow parenting child controls and propagated to them. AFAIK now, as is DHCtrl done (and mine too...) the parent of a windowed child control *don't* have any knowledge of events going to the child, as opposite as normal controls... maybe that can cause problems, I don't know yet.

Hard to say. Anyway, IMO, interface-wise it should be arranged that at the end of EventProc, the widget will call DHCtrl::EvenProc, which will do the all required stuff....

Mirek

Subject: Re: About DHCtrl and window handles...
Posted by mdelfede on Wed, 14 Nov 2007 14:21:56 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 14 November 2007 10:47
OK, np, good idea, adding that now.

good, so I'll remove my friend patch to Ctrlcore and fetch new sources.

Quote:
Quote:
BTW, after I finish my X11 DHCtrl, I guess it would be not bad to rewrite X11 GLCtrl part to be derived of it... I'm already doing it for testing purposes.

Agree.

The windowed control start working ok, and in parallel also the GLCtrl derived from it. I have just a question more : windowed controls (X11 & windows too) should not receive paint/mouse and so events if they are not initialized. Up to now, I resorted to add an isInitialized member to my DHCtrl, but then all code in event handlers should check it before painting/reacting, ecc.
There is not a flag already present in Ctrl class that can be used to automatize that ?
For example, I'd set a "no_event" flag in constructor of my dhctrl, and reset it after window is created. But then the core of Ctrl should look for such a flag and avoid sending events to the

uninitialized ctrl. (at least, no paint events).

Quote:

Hard to say. Anyway, IMO, interface-wise it should be arranged that at the end of EventProc, the widget will call DHCtrl::EvenProc, which will do the all required stuff....

Ok, I'm yet trying to fully understand the event chain, so I can't thell much more on the subject.
What I've seen in some other framework is a main event handler for each control (It could be your EventProc) that is called BEFORE events are dispatched to the control, and that can suppress the dispatching of them, for example, or modify them.
From what I've seen in my code, even if I set Ctrl EventProc to do nothing or to eat all events those are still passed to Paint() function, for example (not yet checked for mouse events). It's normal ?

Ciao

Max

Subject: Re: About DHCtrl and window handles...
Posted by mirek on Wed, 14 Nov 2007 21:18:08 GMT
View Forum Message <> Reply to Message

Quote:
The windowed control start working ok, and in parallel also the GLCtrl derived from it. I have just a question more : windowed controls (X11 & windows too) should not receive paint/mouse and so events if they are not initialized. Up to now, I resorted to add an isInitialized member to my DHCtrl, but then all code in event handlers should check it before painting/reacting, ecc.
There is not a flag already present in Ctrl class that can be used to automatize that ?
For example, I'd set a "no_event" flag in constructor of my dhctrl, and reset it after window is created. But then the core of Ctrl should look for such a flag and avoid sending events to the uninitialized ctrl. (at least, no paint events).

Search for "isopen" and "visible" in X11 code.

Quote:
Ok, I'm yet trying to fully understand the event chain, so I can't thell much more on the subject.
What I've seen in some other framework is a main event handler for each control (It could be your EventProc) that is called BEFORE events are dispatched to the control, and that can suppress the dispatching of them, for example, or modify them.
From what I've seen in my code, even if I set Ctrl EventProc to do nothing or to eat all events those are still passed to Paint() function, for example (not yet checked for mouse events). It's normal ?

IMO should not happen for your DHCtrl.

Otherwise, U++ really cares about top-level Ctrls only. Messages from these are passed to the library and all dispatching is performed by library (to normal, window-less widgets).

Mirek

Subject: Re: About DHCtrl and window handles...
Posted by mdelfede on Wed, 14 Nov 2007 22:55:26 GMT
View Forum Message <> Reply to Message

So far I've noticed one thing... I guess, it's not easy

If I put an X11 windows inside my ctrl, the ctrl don't get Expose events, as it's covered by the X11 window... or, better said, it gets them very few times when I'm adjusting the window pos with ctrl pos (when event happens in the middle of operation).

So, better than putting the window in upp windows list, would maybe better hook a routine in global event handler that do the job.
maybe... not sure about it.

Max

Subject: Re: About DHCtrl and window handles...
Posted by mirek on Fri, 16 Nov 2007 11:00:58 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Wed, 14 November 2007 17:55So far I've noticed one thing... I guess, it's not easy

Yes, I am afraid that sums it up pretty well

Mirek

Subject: Re: About DHCtrl and window handles...
Posted by mdelfede on Fri, 16 Nov 2007 15:50:53 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 16 November 2007 12:00mdelfede wrote on Wed, 14 November 2007 17:55So far I've noticed one thing... I guess, it's not easy

Yes, I am afraid that sums it up pretty well

Mirek

Well, that was referred to "understanding upp ctrl core code", not to write the control. It would be already finished since some days, but I usually like to write good code, not to add a bunch of

For now, I`ve the feeling that the ctrl class hierarchy could have taken advantage from a DHCtrl class from the beginning, being

Ctrl - DHCtrl - TopWindow

the logical one, putting in DHCtrl (both x11 and windows) the low level windows creation part. But it`s too late, I guess

Ciao

Max
