Subject: How to find the top-most window the mouse is over (Linux)?
Posted by James Thomas on Tue, 23 Jan 2007 14:32:23 GMT

I haven't been able to find a way using just U++ to find the highest window that the mouse pointer is currently over (as returned by GetMousePos).

To get around this in Windows I use the API call WindowFromPoint to get the HWND of the highest window under the mouse, but I'm having trouble reproducing this for XWindows. The closest match I have found is XQueryPointer, but this only seems to check against a specified window's parent/children, plus it fails whenever I call it. Example:
XQueryWindow(Xdisplay, main_window->GetWindow(), &Xroot, &Xchild,
   &root_x, &root_y, &win_x, &win_y, &mask);

Is there a way to do this without native API calls? If not I would be very grateful if someone with more experience with XWindows (that isn't difficult - this is the first XWindows call I've tried) could point me in the right direction.

Cheers,
JT

Subject: Re: How to find the top-most window the mouse is over (Linux)?
Posted by mirek on Tue, 23 Jan 2007 23:14:56 GMT

static Ctrl *Ctrl::GetMouseCtrl();

returns (child) widget with mouse.

TopWindow   *Ctrl::GetTopWindow();

returns window child widget is in. Therefore:


```
Ctrl *GetMouseWindow()
{
   Ctrl *q = Ctrl::GetMouseCtrl();
   return q ? q->GetTopWindow() : NULL;
}
```


should solve your problem.

Mirek

Subject: Re: How to find the top-most window the mouse is over (Linux)?
Posted by James Thomas on Wed, 24 Jan 2007 11:25:13 GMT

View Forum Message <> Reply to Message

Thanks, but unfortunately I've tried those and they don't solve my problem. I should have been more specific.

I'm implementing dragging of data within the app between different windows/ctrls and when dragging is in progress I want the mouse pointer set to an icon indicating the type of data, even when over a ctrl that has no drag-drop interface. The only way (I know of) to achieve this is by overloading CursorImage on the source control and using SetCapture to route all CursorImage events to it. This means that GetMouseCtrl (or a mouse hook) will always return the source ctrl and not the one the mouse is actually over.

I have also tried this using vaious other methods, but I like this implementation for two further reasons:
1) While dragging all mouse events can be caught by the drag aware control, which avoids certain 'state' problems.
For instance if the user starts a drag with the left mouse button down and then releases the button while the mouse is outside any application window (where there is no way to get notified of the event) the app still needs to cancel the drag-drop process.
2) It provides a nice interface between the 'source' and 'target' controls.

However, this means I need a way of determining which control is under the mouse when the left mouse button is released even when it is not actually recieving the event. I accomplish this in Windows using the API call above and everything works perfectly, but I am unable to make the XWindows API work similarly.

Am I going about this wrong? Or perhaps there is a better way of implementing this that I haven't thought of? For instance, if there was a way to either hook onto CursorImage events or set the default mouse pointer (as far I can tell this currently impossible - Image::Arrow() is specified directly all over the place) it would solve one my problems and I could work around the rest.

I hope this makes sense. If I have time today I might make a test app to illustrate the problem.

Subject: Re: How to find the top-most window the mouse is over (Linux)?
Posted by mirek on Wed, 24 Jan 2007 12:34:24 GMT

View Forum Message <> Reply to Message

James Thomas wrote on Wed, 24 January 2007 06:25the source control and using SetCapture to route all CursorImage

Ooops. Yes SetCapture spoils the fun

Quote:
For instance, if there was a way to either hook onto CursorImage events or set the default mouse

pointer (as far I can tell this currently impossible - Image::Arrow() is specified directly all over the place) it would solve one my problems and I could work around the rest.

Actually, it is possible to hook mouse messages:

static void   Ctrl::InstallMouseHook(MouseHook hook);

See CtrlLib/ToolTip.cpp for usage example.

OTOH, I do not think this is a good idea for D&D.

Quote:
I hope this makes sense. If I have time today I might imlpement an app to illustrate the problem.

Definitely makes sense.

Mirek

Subject: Re: How to find the top-most window the mouse is over (Linux)?
Posted by James Thomas on Wed, 24 Jan 2007 12:42:08 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 24 January 2007 07:34
Actually, it is possible to hook mouse messages:

static void   Ctrl::InstallMouseHook(MouseHook hook);

See CtrlLib/ToolTip.cpp for usage example.

OTOH, I do not think this is a good idea for D&D.

Definition of MouseHook:
typedef bool (*MouseHook)(Ctrl *ctrl, bool inframe, int event, Point p, int zdelta, dword keyflags);
I can intercept the CursorImage message with a mouse hook, but there is no way that I can see of returning the Image I want.

But I agree that this would not be a sensible way of doing it anyway.

Cheers,
JT

Subject: Re: How to find the top-most window the mouse is over (Linux)?
Posted by James Thomas on Wed, 24 Jan 2007 16:01:21 GMT

Since I thought it would be interesting and I wanted to test out an implementation using templates (somewhat inspired by WithDropList) I've now written a test app that demonstrates what I'd like to achieve and shows two methods of implementing drag-drop between controls using GetData/SetData.

My real-world app obviously requires a more complicated implementation than this but the method I use is identical.

Written for 612-dev-3 and only tested on XP. Should compile with GCC but one method won't work in Linux (which is the whole problem).

p.s. Sorry for having all the function bodies in the headers. I can't get MSVC8 to link templated classes with the body outside the class and I'm too lazy to fix it.

File Attachments
1) DragTest.zip, downloaded 396 times