Subject: How to simulate keyboard (and without focus)?
Posted by WebChaot on Tue, 28 Nov 2006 21:12:00 GMT
View Forum Message <> Reply to Message

Hi all!

Last time I asked how to catch keyboard input. Mirek directed me to the (simple) answer. Thanks!

But now I want to do the other way: I want to simulate keyboard input.

Is it possible to "send" a key via code?

The second question is: Is it possible to popup a static rect with some buttons - but without to get focus?

I need to develop a virtual keyboard for a touchscreen solution - therefore I dont want to loose focus on selected editfield while typing (=clicking) on the popped up keypanel.

Is that possible?

I hope, I didn't explain too confusing

Thanks in advance,

WebChaot.

Subject: Re: How to simulate keyboard (and without focus)?
Posted by mirek on Tue, 28 Nov 2006 22:05:24 GMT
View Forum Message <> Reply to Message

WebChaot wrote on Tue, 28 November 2006 16:12

Is it possible to "send" a key via code?

Just call Key method by your code (normally, it is called as result of Key event, but nothing can prevent your code doing so).

Quote:
The second question is: Is it possible to popup a static rect with some buttons - but without to get focus?

There is "NoWantFocus" flag - Ctrls with such flag cannot obtain focus by "standard" means (e.g. by TopWindow). They still can get it explicitly by your code, but that you can avoid..

I need to develop a virtual keyboard for a touchscreen solution - therefore I dont want to loose focus on selected editfield while typing (=clicking) on the popped up keypanel.

I see. Yes, it is OK. However, maybe even moving focus would be OK - virtual keyboard is mouse thing, I would just left focus processing as it is and simply send virtual keyboard events to GetFocusCtrl()->Key

I hope, I didn't explain too confusing :)

Actually, very understandable

More actually, that app sounds interesting. Keep us informed

Mirek

Subject: Re: How to simulate keyboard (and without focus)?
Posted by WebChaot on Tue, 28 Nov 2006 22:24:13 GMT
View Forum Message <> Reply to Message

>>> I see. Yes, it is OK. However, maybe even moving focus would be OK - virtual keyboard is mouse thing, I would just left focus processing as it is and simply send virtual keyboard events to GetFocusCtrl()->Key

I have to play around tomorrow. I want to let the edit cursor (does it have a name - the line which is blinking) stay visible. Therefore I thought, that the editfield should be focused all the time.

>>> More actually, that app sounds interesting. Keep us informed

Sure! As I wrote you some weeks ago, we are new in Ultimate++ and we decide to develop all our new applications with it. So now we started with our new version of a touchscreen application. We will send you more information and some screenshots, when there is something interesting to show. At the time we do some base development (touchscreen elements, advanced array control and much more).

Do you like to get technical feedback? Because we try to let U++ source as it is and only to create and / or overwrite our own classes (to be able to update U++ development releases without much work). But in some cases (3 times I think) we had to change U++ sources (very small changes!). Maybe it is possible to include this changes in next releases, because they are useful for others too?

Thanks,

WebChaot

Subject: Re: How to simulate keyboard (and without focus)?
Posted by mirek on Tue, 28 Nov 2006 22:29:01 GMT
View Forum Message <> Reply to Message

WebChaot wrote on Tue, 28 November 2006 17:24>>> I see. Yes, it is OK. However, maybe even moving focus would be OK - virtual keyboard is mouse thing, I would just left focus processing as it is and simply send virtual keyboard events to GetFocusCtrl()->Key

I have to play around tomorrow. I want to let the edit cursor (does it have a name - the line which is blinking) stay visible.

Well, the terminology names that blinking thing a "caret". Anyway, that is exactly the reason to keep normal focus processing and make your virtual keyboard "NoWantFocus". (Now thinking about it, maybe that is exactly what you wanted to do

Quote:
Do you like to get technical feedback? Because we try to let U++ source as it is and only to create and / or overwrite our own classes (to be able to update U++ development releases without much work). But in some cases (3 times I think) we had to change U++ sources (very small changes!). Maybe it is possible to include this changes in next releases, because they are useful for others too?

Of course! Actually, U++ develops fast, I hope you do not want to patch each single version

Of course, the reasons for patches have to be stated (and sometimes discussed).

Mirek
