Subject: Touchscreen and buttons
Posted by Giorgio on Tue, 08 Aug 2017 10:10:45 GMT
View Forum Message <> Reply to Message

Hi there,
I am experimenting with a touch screen on a raspberry. I developed the application on Windows, but it compiled on raspbian with almost no additional effort - just to fix less than 20 lines of code - and I think that is quite remarkable.

The original application was meant to be used with mouse and keyboard, so I have to tune the User Interface to make the application usable: first problem I encountered is with buttons.

If I use a "custom" button (i.e. a button created by me and to which I assigned a function with btn.WhenPush = THISBACK(myFunction);) the button works as expected. If I use "built-in" buttons (i.e. the button ok and cancel of a CtrlLayoutOKCancel window) they do not work: when I press them on the touch screen, I see that they are pressed and released, but nothing happens. Someone else encountered this problem?

As a workaround, I tried to assign the method Close() to the ok and cancel buttons, but - due to my newbie-ity - I did not figure out how to do that; this is the original code, working without a problem with keyboard and mouse:

```
#include "ListOp.h"

ListOp::ListOp(MLav _macrolav)
{
 CtrlLayoutOKCancel(*this, t_("Select operator"));
 std::vector<S_OPERATORS> mylist = db_data.FindOperators(_macrolav);
 for (size_t i=0; i<mylist.size(); i++)
  SwOperators.Set((int)i, mylist[i].ID, mylist[i].NAME);
 ok.Ok();
 cancel.Cancel();
}
```

I tried adding the following:

```
 [...]

 ok.Ok();
 ok.WhenPush = THISBACK(this.Close);
 cancel.Cancel();
 cancel.WhenPush = THISBACK(this.Close);
```

...and this:

[...]

```
ok.Ok();
ok.WhenPush = THISBACK(this->Close);
cancel.Cancel();
cancel.WhenPush = THISBACK(this->Close);
```

...but the code does not compile.

Any suggestion is appreciated.
Regards,
Gio

Subject: Re: Touchscreen and buttons
Posted by rafiwui on Tue, 08 Aug 2017 11:26:06 GMT
View Forum Message <> Reply to Message

What does the compiler complain about?
How are ok and cancel defined?

You don't need this in the THISBACK:

```
ok.WhenPush = THISBACK(Close);
cancel.WhenPush = THISBACK(Close);
```

And if you want to exit the app you should use Breaker I guess:

```
ok.WhenPush = Breaker();
cancel.WhenPush = Breaker();
```

Subject: Re: Touchscreen and buttons
Posted by Giorgio on Tue, 08 Aug 2017 12:16:44 GMT
View Forum Message <> Reply to Message

Hi rafiwui,
thanks for the prompt answer.

The buttons "cancel" and "ok" are defined in the .lay file.

Using Close as you suggested compiles and runs, but the application does not works as intended.
A little bit of a context. When launched, GUI_APP_MAIN creates and launches an instance of a

class (HomeScreen) that manages and creates the main window of the application. The HomeScreen class, before launching the main window, displays a window (ListOp) asking what is the user that is working. This ListOp windows is the one that has the ok and cancel buttons. In the old scenario, after the user selected the operator and hit ok, the main window was created. Now, if I add the instrucrion ok.WhenPush = THISBACK(Close) the application ends. On the shell from which I launched the application I read "terminate called after throwing an instance of 'Upp::ExitExc'". Does that mean that I have to invoke another method instead of Close?
Thanks,
Gio

Subject: Re: Touchscreen and buttons
Posted by Giorgio on Tue, 08 Aug 2017 12:20:33 GMT
View Forum Message <> Reply to Message

I've got myself an answer: if i put ok.WhenPush = THISBACK1(Break, 1); it works as intended.
