Subject: Strange behaviour of my modal dialog
Posted by forlano on Tue, 25 Apr 2006 20:57:27 GMT
View Forum Message <> Reply to Message

I didn't know where to post this thread because I'm sure it is not a bug. Nevertheless i'm observing strange things, very strange, at least for me.

I've declared and defined a class, say Tournament. Then I've an instance TD of Tournament the which constructor set, for example

```
...
NAMETORNEO = "my Tournament";
...
```

Now I load a file and update NAMETORNEO with the value read from file, for example

```
...
TD.NAMETORNEO = "international";
...
```

I tested that TD.NAMETORNEO has been adjourned and now it is "international", no doubt about it please.
Now comes the strange things. I run a modal dialog to set a new tournament. The dialog is run by:

```
void VegaMain::NewTournament()
{   NewTournamentDlg newt;
    //PromptOK( TD.NAMETORNEO + "1" ); // show "international"
    newt.Run();
    //PromptOK( TD.NAMETORNEO + "2" ); // show "international"
}
```

Here is some lines of the constructor of NewTournamentDlg :

```
NewTournamentDlg::NewTournamentDlg()
{   int i, j;
    CtrlLayout(*this, "New Tournament");

    editTourn <<= TD.NAMETORNEO;
    PromptOK( TD.NAMETORNEO ); // TD.NAMETORNEO is again "my Tournament" !!!
...
```

I believed to set in editTourn the new value of TD.NAMETORNEO, i.e. "international", instead there is again the very old one, "my Tournament"! Why? After all newt born after I changed the value of TD.NAMETORNEO.

Perhaps it is normal but I need to synchronize NewTournamentDlg with the rest of the world. How to do?

I hope you have understood what I meant.
Luigi

Subject: Re: Strange behaviour of my modal dialog
Posted by mirek on Tue, 25 Apr 2006 21:10:15 GMT
View Forum Message <> Reply to Message

Maybe I did not get it completely, but I guess the mistake you have done is that you think that "<<=" somewhat "binds" the variable with EditField (or other Ctrl).

However, that is not true - it is simple assignment, it sets the value of EditField, but the value of "TD.TOURNAMENT" is further unchanged by the EditField. If you need to have it changed, you have to perform "backward" assignment like

TD.NAMETONEO = ~newt.editTourn;

Of course, doing so for each variable would be boring, so U++ gives you better options.

First of all, you can consider whether you really need to have separate variable for tournament name. Sometimes you can simply use the dialog to store data and do not store to variable at all. (Data are stored in widgets regardless it is "open").

Another option is to use "CtrlRetriever". This provides a kind of binding you perhaps have expected:

```
CtrlRetriever r;
r(newt.editTorneo, TD.TORNEO)
 (newt.anotherField, TD.ANOTHER)
....
;
newt.Run();
r.Retrieve();
```

First past, using operator(), simply performs

ctrl <<= val;

and stores references to both to the list. Then Retrieve method goes through those references and performs

val = ~ctrl;

Primitive, but effective

Mirek

Subject: Re: Strange behaviour of my modal dialog
Posted by forlano on Tue, 25 Apr 2006 21:33:34 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 25 April 2006 23:10Maybe I did not get it completely, but I guess the mistake you have done is that you think that "<<=" somewhat "binds" the variable with EditField (or other Ctrl).

However, that is not true - it is simple assignment, it sets the value of EditField, but the value of "TD.TOURNAMENT" is further unchanged by the EditField.
I was aware of it. I wanted put in EditField the value TD.TOURNAMENT and nothing else. I wanted to know the current value of TD.TOURNAMENT. Later another method will get what I've written in the EditField and then update TD.TOURNAMENT. My surprise is that TD.TOURNAMENT that finish in the EditField is the very old one. It seems the widget is not aware that TD.TOURNAMENT has already changed at some moment.

Quote:
If you need to have it changed, you have to perform "backward" assignment like

TD.NAMETONEO = ~newt.editTourn;

Of course, doing so for each variable would be boring, so U++ gives you better options.

First of all, you can consider whether you really need to have separate variable for tournament name. Sometimes you can simply use the dialog to store data and do not store to variable at all. (Data are stored in widgets regardless it is "open").

This is very interisting! I was not aware of it. In fact I get tired to use a parallel structure to store variables (I was afraid to loose them when the widget disappeared!)
Quote:Another option is to use "CtrlRetriever". This provides a kind of binding you perhaps have expected:

```
CtrlRetriever r;
r(newt.editTorneo, TD.TORNEO)
 (newt.anotherField, TD.ANOTHER)
....
;
newt.Run();
r.Retrieve();
```

First past, using operator(), simply performs

ctrl <<= val;

and stores references to both to the list. Then Retrieve method goes through those references and performs

val = ~ctrl;

Primitive, but effective

Mirek

I'm going to try it!

Luigi

Subject: Re: Strange behaviour of my modal dialog
Posted by forlano on Tue, 25 Apr 2006 21:52:30 GMT
View Forum Message <> Reply to Message

I tried the retrieve method and it is very nice. Irrispective of how the widget is closed you get what there was at its closure.
So the widget continue to stay alive but invisible.

Is there a method to destroy a widget definitely? (just to prevent somethig I do not want to do).

Luigi

Subject: Re: Strange behaviour of my modal dialog
Posted by mirek on Tue, 25 Apr 2006 22:30:33 GMT
View Forum Message <> Reply to Message

forlano wrote on Tue, 25 April 2006 17:52I tried the retrieve method and it is very nice. Irrispective of how the widget is closed you get what there was at its closure.
So the widget continue to stay alive but invisible.

Is there a method to destroy a widget definitely? (just to prevent somethig I do not want to do).

Luigi

Ahh, you are mixing two issues: Retrieve is there to store data *separately* (means, in paralel data structure).

To destroy widget and its value definitely, you simply destroy it (by going out of scope or by calling delete for heap-based).

Mirek

Subject: Re: Strange behaviour of my modal dialog
Posted by forlano on Wed, 26 Apr 2006 16:13:07 GMT

Hello,

I'm sorry to bother the forum once more with the same issue. Unfortunately I can't bypass this stupid problem that stopped me. I feel it is not a bug but the mistake it is not at all evident to me. I've prepared a minimalist example that shows what the problem is. They are 5 files very short. Who is interested can run the package attached (unzip in vegaTest). Here I show each file with the hope that the problem can be fixed on the fly.

File 1, is the *.h with the declaration of the two classes that are used by the program, one of them is a widget:

```
#include <CtrlLib/CtrlLib.h>
#define LAYOUTFILE      <testVega/Vega.lay>
#include <CtrlCore/lay.h>

class NewTournamentDlg : public WithNewTournamentLayout<TopWindow> {
public:
 typedef NewTournamentDlg CLASSNAME;
 NewTournamentDlg();
 ~NewTournamentDlg() {;}
};

class RoundData {
public:
  String NAMETORNEO;
  RoundData();
 ~RoundData(){;}
};

// maybe the problem is the line below?
class RoundData TD; //is it seen in each *.cpp file? Is it a definition or a declaration?
```

File 2 is the definition of the method of the above widget. It put (it should put!) in an EditString the value TD.NAMETORNEO that has been set elsewhere:

```
#include "VegaMain.h"

NewTournamentDlg::NewTournamentDlg()
{   int i, j;
 CtrlLayout(*this, "");

    editTourn <<= TD.NAMETORNEO;
}
```

and this is the correspondent .lay file:

```
LAYOUT(NewTournamentLayout, 268, 92)
 ITEM(EditString, editTourn, NotNull(true).LeftPosZ(16, 228).TopPosZ(36, 19))
END_LAYOUT
```

File 4, define the constructor of the non widget class:

```
#include "VegaMain.h"

RoundData::RoundData()
{ NAMETORNEO = "my torneo";
}
```

So "my torneo" is the name with which born an instance of the previous class. And now the last file, main.cpp, where happen strange things:

```
#include "VegaMain.h"

void NewMask()
{ TD.NAMETORNEO = "luigi";
        NewTournamentDlg newt;
 newt.Run(); //appear "my torneo"
 PromptOK( TD.NAMETORNEO); //appear "luigi"
}

GUI_APP_MAIN
{   TopWindow w;
 Button b;
 w.Add(b);
 b.SetLabel("push me").LeftPos(10, 100).TopPos(10, 30);
 b.WhenAction = callback(NewMask);
 w.Run();
}
```

As said in a previous post, after the previous:
TD.NAMETORNEO = "luigi";
I expect to see it inside the editstring of the widget because of its constructor. Instead appear "my torneo". It looks as TD is not seen by newt. How can I do visibile TD to newt? In C I could make TD global, but now?

What I am missing to do?
Thank you,
Luigi

File Attachments

1) testVega.rar, downloaded 1666 times

Subject: Re: Strange behaviour of my modal dialog
Posted by mirek on Wed, 26 Apr 2006 17:06:56 GMT
View Forum Message <> Reply to Message

forlano wrote on Wed, 26 April 2006 12:13
// maybe the problem is the line below?
class RoundData TD; //is it seen in each *.cpp file? Is it a definition or a declaration?

This indeed is not quite right. You should place

extern RoundData TD;

into header (declaration) and

RoundData TD;

into any .cpp file (definition). After fixing that, it started working.

Mirek

Subject: Re: Strange behaviour of my modal dialog
Posted by forlano on Wed, 26 Apr 2006 18:31:50 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 26 April 2006 19:06This indeed is not quite right. You should place

extern RoundData TD;

into header (declaration) and

RoundData TD;

into any .cpp file (definition). After fixing that, it started working.

Mirek
.SetSize(MAX);

Thanks a lot!
I was becoming crazy. This stupid problem stopped me for an entire day (night included).
There is an explanation for this. After I read some notes about C++ that complained about the "extern" used in C, and after Aris complained too, , for my extensive use of "spaghetti" I become a Taleban and I wanted to remove all the "extern" from my program.
Perhaps I've projected the application not in the best way if now I need it, so I've altered the beauty of the Aris' code.
I'll post soon the first part of the application where there are all the things I've learned so far.

Luigi
