Subject: How to use callbacks with "embedded" controls...
Posted by forlano on Sun, 16 Apr 2006 16:54:26 GMT
View Forum Message <> Reply to Message

Quote:
Waiting for your next task
Aris

Here I am,
I think I need other few things to understand before I can proceede alone (well... almost alone).
One is the following (perhaps the title of the thread is not correct).

We have a marvellous threestate OptionImage of which all the world is envious. I've attached to it a callback (it costed me one hour of useless temptives before to found the exact syntax... from other side very obvious at the end)

```
void Avail3(One<Ctrl>& ctrl)
{    ctrl.Create<OptionImage>().ThreeState().SetImage(imgYes(), imgNo(),
imgMaybe()).WhenAction=callback(Avail3Action);
}
```

where the callback is

```
void Avail3Action()
{ int int_row;
 String s ;
 PromptOK("ImageButton action");
}
```

What I desire is that this callback can recognize the state of the ImageButton that has activated it. So that it can prompt the state of the pressed ImageButton to inform the user about what he did (it is a very delicate operation).
Unfortunately this callback is not a method of the class so it has no access to the ArrayCtrl. Otherwise I could easily look in the cell and retrieve the state of the pressed button.
I'm sure there is way to pass as argument of Avail3Action() just the pressed button. It should be already available in some form in Avail3() so that it can be passed even to Avail3Action.
Unfortunately I'm lost with the syntax like One<Ctrl>& ctrl.
I've even tried to let be the previous callback two new methods of the class to have access to all the instances, but the compiler cried in corrispondence of the argument One<Ctrl>& ctrl passed to the metod Avail3().

Have you any hint?
Thank you,
Luigi

Edit: The title was changed...

Subject: Re: How to use callbacks with "embedded" controls...
Posted by fudadmin on Mon, 17 Apr 2006 00:54:38 GMT
View Forum Message <> Reply to Message

First of all, don't create too long code "sausages", esp. in the wrong places...

Second, don't spend your hours in the wrong bushes by creating a bicycle. That means, always check if your parents have those wheels...
I'm not very sure but I guess:

Step1.

```
void VegaMain::Init()
{
 info="Ready";

 tab1.arr.WhenEnterRow = THISBACK(UpdateInfo);
 tab1.arr.WhenCtrlsAction = THISBACK(UpdateInfo);
}
```

Step2. in VegaStatus.cpp add this:

```
void VegaMain::UpdateInfo()
{
 info = "Wait..Counting...Busy...";
 int nTotal = tab1.arr.GetCount();
 int nAvail = tab1.GetAvail(); //might optimize later...
 info = NFormat("Ready: Total players: %d - Available: %d", nTotal, nAvail);
}
```

Step3
update VegaMain.h

```
 void UpdateInfo();
```

Step4
and also in the same VegaMain.h

```
class VegaTab1 : public WithVegaTab1Layout<TopWindow> {
public:
 void SortRecord(int column);
 void AddPlayer();
// void ShowNumberRecord(); you don't need it here...
 int GetAvail();  //new!!!
```

Step5
in VegaTab1.cpp

```
int VegaTab1::GetAvail()
```

```
{
 int n=0;
 for(int i=0; i<arr.GetCount(); i++) {
  if ( arr.Get(i, "Avail") ) n=n+1;
 }
 return n;
}
```

Step5
remove your 1 hour wasted time and spoiled callback from Avail3
Step6
make "info" label in the layout at least double width.
Step7
Ctrl_F5.

You can apply this logic with Prompts if you like...
Step8
Report any success or failure...

Subject: Re: How to use callbacks with "embedded" controls...
Posted by forlano on Mon, 17 Apr 2006 07:33:13 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Mon, 17 April 2006 02:54You can apply this logic with Prompts if you like...
Step8
Report any success or failure...

Every thing work very well! You have anticipated my wishes with the GetAvail method.
My mistake is that I continue to think in C style where all the functions are global and I can jump from one place to another.
Instead now I must put the callback in the right place. It is not immediate but I'll learn.

One more note. I've modified the GetAvail() method to show the status of all the Option image
```
int VegaTab1::GetAvail()
{ int n=0;
 String s="value=" ;

 for(int i=0; i<arr.GetCount(); i++) {
  if ( arr.Get(i, "Status") ) n=n+1;
  s += AsString( arr.Get(i, "Status") ) + " ";
 }
 PromptOK( s );
 return n;
}
```
I was surprised to observe that the state "Maybe" return no value! I mean, 'V'='true' return 1, 'X'='false' return 0, but
'?'=maybe seemd return nothing? In fact I can't see its value. Is it normal? Maybe it is 'null'. I

believed to obtain some thing like 2 or -1.
Luigi

Subject: Re: How to use callbacks with "embedded" controls...
Posted by fudadmin on Mon, 17 Apr 2006 08:03:35 GMT
View Forum Message <> Reply to Message

If something is empty, it means it's empty! I do not question such things. Just accept as they are .
P.S in fact, I think, you can study nuller classes etc. if you wish...

Subject: Re: How to use callbacks with "embedded" controls...
Posted by forlano on Mon, 17 Apr 2006 08:19:51 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Mon, 17 April 2006 10:03If something is empty, it means it's empty! I do not question such things. Just accept as they are .
P.S in fact, I think, you can study nuller classes etc. if you wish...
Yesterday I've upgraded the U++ and only now I've seen that the documentation is increased. In particular about three state I have just read
Quote:
Widget providing the selection of 2 or alternatively 3 states (true, false, Null). Value of Option is either 0, 1, or Null. When setting Value to Option, string "1" is interpreted as true state, other non-Null strings as false.

I accept it without complain.

Subject: Re: How to use callbacks with "embedded" controls...
Posted by fudadmin on Mon, 17 Apr 2006 09:04:39 GMT
View Forum Message <> Reply to Message

forlano wrote on Mon, 17 April 2006 08:33
My mistake is that I continue to think in C style where all the functions are global and I can jump from one place to another.
Instead now I must put the callback in the right place. It is not immediate but I'll learn.

Luigi

I noticed one your mistake that you tend to think and want to install callbacks (and access methods) from down up or from inside out. It should be opposite way, IMHO. I mean, you should imagine your classes (or instances) like russian matrioshka doll, if you know such term.
And in most cases you should start your access analysis from the biggest doll then going inside to the smallest ones (members or children). This way you find then right place for your callback assignment.
In paralel you can think about your callback function body (not its place!), and make it even

before you install a callback and test it with one more temporary function.

This is my method of programming. Others might offer different approaches.

If that helps...

Subject: Re: How to use callbacks with "embedded" controls...
Posted by forlano on Mon, 17 Apr 2006 09:19:34 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Mon, 17 April 2006 02:54
Report any success or failure...

I'm very sorry to remain again on the same argument, but I'm obeserving a strange behaviour and before to say it is a bug it is better to verify.
It regards the OptionImage inside the array. When I click in each cell of the array except the OptionImage the cursor of the array move on the clicked row. This is OK because I can read the selected row and retrieve its data. Instead when I click on the OptionImage the cursor doesn't change and remains where it was. This is not good because my code cannot intercept the clicked row and in turn the clicked optionimage button.
Is it normal? It seems that the button has overrided the property of that column as it where not in an array.
Is there an alternative way to listen the optionimage button? (if not let's neglect it).
This is the code that prompt the status of the clicked button.

```
int VegaTab1::GetAvail()
{ int n;
 String s, player, s1;

   // n = arr.GetCursor(); // doesn't change, so I try the next line
 n = arr.GetClickRow(); // nothing
   // arr.SetCursor( n);

   s=AsString( arr.Get(n, "Avail") );

   player = arr.Get(n, "Name");
   if (s=="1") PromptOK("You have made available player " + player);
   else if (s=="0") PromptOK("You have made NOT available player " + player);
   else PromptOK("You have assigned a BYE to player " + player);


   n = 0;
 for(int i=0; i<arr.GetCount(); i++) {
  if ( arr.Get(i, "Avail") ) n=n+1;
  //s1 += AsString( arr.Get(i, "Avail") ) + " ";
 }
   // PromptOK( s1 );
```

```
 return n;
}
```

Subject: Re: How to use callbacks with "embedded" controls...
Posted by fudadmin on Mon, 17 Apr 2006 09:55:36 GMT
View Forum Message <> Reply to Message

I think extra controls should be made
1. "transparent" for the events or
2. we need to synchronize them.
Need to think and/or to study.

Subject: Re: How to use callbacks with "embedded" controls...
Posted by fudadmin on Mon, 17 Apr 2006 10:15:03 GMT
View Forum Message <> Reply to Message

Don't worry too much about that now. Think next problems. The solution will come...

Subject: Re: How to use callbacks with "embedded" controls...
Posted by fudadmin on Wed, 19 Apr 2006 04:26:33 GMT
View Forum Message <> Reply to Message

Ok, I think, this should help:

Step1. in OptionImage.cpp

```
#include "OptionImage.h"

//added new method...
void OptionImage::ReFocus(){
 SetFocus();
}

OptionImage::OptionImage()
{
 WhenAction = THISBACK(ReFocus); //added...
}
...
```

Step2. in OptionImage.h

```
class OptionImage : public Option {
 void ReFocus(); //added -let's make it private...
```

```
protected:
 Image imgYes, imgNo, imgMaybe;
public:
...


Step3. in VegaTab1.cpp

int VegaTab1::GetAvail()
{

 int n=0;
 for(int i=0; i<arr.GetCount(); i++) {
  if ( arr.Get(i, "Avail") ) n=n+1;
 }

 //try this here. Later we'll rearrange...
 arr.ChildGotFocus();   //lets move the focus after calculations...
 //do you tests below...

 return n;
}
```

Subject: Re: How to use callbacks with "embedded" controls...
Posted by forlano on Wed, 19 Apr 2006 07:16:25 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Wed, 19 April 2006 06:26Ok, I think, this should help:
[snip]

Thanks a lot Aris,

I'll try it this afternoon. Yesterday I started to read the first 6 chapters of the C++ Eckel's book. Then I've understood what you meant regarding the #include and extern (declaration and definition). Now I can see the nightmare I produced in my first version of the application.

Luigi

Subject: Re: How to use callbacks with "embedded" controls...
Posted by forlano on Wed, 19 Apr 2006 14:44:57 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Wed, 19 April 2006 06:26Ok, I think, this should help: [snip]

Aris,

I tried the code and it works nicely, but... there is a little drawback.
If you remind I wanted to intercept the clicked embedded OptionImage to get its status and prompt the user with a message (and write in the infoctrl too). Now I can do it because, thanks to you, the focus go on the row in which the button is. This is the code that shows its status:

```
 int VegaTab1::GetAvail()
{ int n = 0;
 String s, player, s1;

 for(int i=0; i<arr.GetCount(); i++) {
  if ( arr.Get(i, "Avail") ) n=n+1;
 }
 //try this here. Later we'll rearrange...
 arr.ChildGotFocus();   //lets move the focus after calculations...
 //do you tests below...

        n = arr.GetCursor();
 s=AsString( arr.Get(n, "Avail") );
     PromptOK( s );
 return n;
}
```

Now, when I click the optionimage, the PromptOK(s) appear TWO times, instead of one, and it is annoying. This is due both do the click and the refocus connected connected to WhenCtrlsAction :

```
void VegaMain::Init()
{ info1="Ready";
 tab1.arr.WhenEnterRow = THISBACK(UpdateInfo);
 tab1.arr.WhenCtrlsAction = THISBACK(UpdateInfo);
}
```

that call GetAvail:

```
void VegaMain::UpdateInfo()
{ int int_row, nTotal, nAvail;

 nTotal = tab1.arr.GetCount();
 nAvail = tab1.GetAvail();
 info1 = NFormat("Total players: %d - Available: %d", nTotal, nAvail);
}
```

I've noticed that if you comment
// tab1.arr.WhenCtrlsAction = THISBACK(UpdateInfo);
then the consecutive click on the option image will not produce a PromptOK dialog.
So now I'm a bit confused on the place where I have to put the hands to correct this behaviour.
I hope to have been clear.
Luigi

Subject: Re: How to use callbacks with "embedded" controls...
Posted by fudadmin on Wed, 19 Apr 2006 15:25:33 GMT
View Forum Message <> Reply to Message

Quote:
I've noticed that if you comment
// tab1.arr.WhenCtrlsAction = THISBACK(UpdateInfo);
then the consecutive click on the option image will not produce a PromptOK dialog.
So now I'm a bit confused on the place where I have to put the hands to correct this behaviour.
I hope to have been clear.
Luigi

Yes, that's why I commented "//try this here. Later we'll rearrange..."

1. You need to think about the sequence:
1.1. arr.WhenCtrlsAction=THISBACK(CursorOnFocus?);
1.2. with inside arr.ChildGotFocus();
then
1.3. warn (Prompt) user with the possibility to cancel or confirm the change or only to display warning in status.

2. and only then let update and invoke maybe some other arr.When???... update info

Try or leave it to me...

Subject: Re: How to use callbacks with "embedded" controls...
Posted by fudadmin on Wed, 19 Apr 2006 15:35:14 GMT
View Forum Message <> Reply to Message

I would also think about keeping track in a separate DB table of Available- not available Periods and who made and for what reason... Hm?

Subject: Re: How to use callbacks with "embedded" controls...
Posted by forlano on Wed, 19 Apr 2006 19:36:03 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Wed, 19 April 2006 17:25Try or leave it to me...

I've tried but without success. Perhaps I must read some more chapters about C++. At this moment I do not catch the logic behind the GUI class and how they react at some action.
In particular this optionimage that was not automatically followed by the focus and need so much attention is making me crazy!

When I click on it I want a simply warning ("The status changed!") and nothing else (no confimation) and the avail players should be calculated once more. Very simple, but this embedded control seems the evil inside the array. In the original C version it was out of the array... if it will continue to create problems I'll kill it and move in the edit mask; at its place I'll use a stupid colored label... yes I promise I'll do it!
Quote:
I would also think about keeping track in a separate DB table of Available- not available Periods and who made and for what reason... Hm?

I've not understood what you mean. Any way, later, the pressure of the SaveMenu item, should produce two parallel things:
1. save on a file all the array data, and
2. transfer all the array data in a minimalist database structure that will support the output of the elaboration.

I was forgetting, equivalent to the SaveMenu are the actions performed by the Modify button and by the ... optionimage button, yes again it! Each modification I want be permanent.

What do you think? I'll move the optionimage button or we give it the last possibility?

Subject: Re: How to use callbacks with "embedded" controls...
Posted by forlano on Wed, 19 Apr 2006 22:46:27 GMT
View Forum Message <> Reply to Message

forlano wrote on Wed, 19 April 2006 21:36
What do you think? I'll move the optionimage button or we give it the last possibility?

This night I wanted to see how looks the application without the embedded ctrl in the array. It is attached. Now in this version everything works very nice and the label info is updated regularly just with the previous code.
Only one annoying problem: my bitmap are 19x19 pixel and are not painted in good manner.
... and a little problem: from the array now is not well visible the status of each player. So I think to write in red (imgNo), blank(imgYes) and blue (imgMaybe) the symbol status of each player.
Any comments?

Luigi

File Attachments
1) Vega2.rar, downloaded 1926 times

Subject: Re: How to use callbacks with "embedded" controls...
Posted by fudadmin on Thu, 20 Apr 2006 03:02:25 GMT
View Forum Message <> Reply to Message

Yes, you can read a few more chapters. Also, it's useful writing some notes when you find anything new. One more tip - read everything with an approach that later you will need to explain or teach someone else efficiently. If you have enthusiasm you could write one more chapter of Ultimate's tutorial.

P.S. I'll write more and present you with once more "refurbished" Vega very soon...
