
Subject: "Suicide Buttons Array" or how to re-index callbacks or...?

Posted by [fudadmin](#) on Mon, 29 May 2006 10:08:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi, I'm stuck with this piece of code:

```
#include <CtrlLib/CtrlLib.h>

struct App : TopWindow {
    StaticRect      work;
    Array<Button>    buts;
    //change count to change work area width
    static int const count = 10;
    static int const elw = 40;
    static int const vw=500;
    static int const ww = count*elw+50;
    static int const wh = 290;

    void Suicide(int i){
        //i=GetNewIndex(i);
        buts.Remove(i);
        //ReIndex();
    }

    typedef App CLASSNAME;

    App() {
        work.SetRect(50,50,ww,wh);
        work.Color(SRed);
        Add(work);

        for(int i=0;i<=count;i++){
            buts.Add();
            buts[i].SetRect(5+i*elw,20,30,20);
            buts[i].SetLabel(AsString(i));
            work.Add(buts[i]);
            buts[i].WhenAction=callback1(this,Suicide,i); //incorrect due to lost index!...
        }

    }
};

GUI_APP_MAIN
{
    App().Title("Suicide Buttons Array").Sizeable().Zoomable().Run();
}
```

My problem is not for buttons (this code is just visual representation of the problem. I know that I can hide buttons or remove them from parent Ctrl's. And I know that I can arrange remove buttons with "outside management").

But my problem is something like "how to re-index "suicide" callbacks with dynamic containers". Or what else can be used in such cases?

(Somehow I feel that my custom index management is not the best case with ntl libraries...) Any ideas?

Subject: Re: "Suicide Buttons Array" or how to re-index callbacks or...?

Posted by [fudadmin](#) on Mon, 29 May 2006 18:01:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

P.S. Or what could be a more effective way than:

```
void ReIndex(){
    for(int i=0;i<buts.GetCount();i++){
        buts[i].SetLabel(AsString(i));
        buts[i].WhenAction=callback1(this,Suicide,i);
    }
}
```

.S. Or what could be a more effective way than:

Subject: Re: "Suicide Buttons Array" or how to re-index callbacks or...?

Posted by [mirek](#) on Tue, 30 May 2006 21:42:01 GMT

[View Forum Message](#) <> [Reply to Message](#)

fudadmin wrote on Mon, 29 May 2006 14:01P.S. Or what could be a more effective way than:

```
void ReIndex(){
    for(int i=0;i<buts.GetCount();i++){
        buts[i].SetLabel(AsString(i));
        buts[i].WhenAction=callback1(this,Suicide,i);
    }
}
```

.S. Or what could be a more effective way than:

<CODE]

```
void ReIndex(){
    for(int i=0;i<buts.GetCount();i++){
        buts[i].SetLabel(AsString(i));
        buts[i].WhenAction=callback1(this,Suicide,i);
    }
}
```

</CODE

every time after removing?

Or maybe to mark them as dead and then to filter actions everywhere else?

Are there any better ways?

Well, I believe this is not very callback-specific... it is simply the problem of unique identification of item in the array...

Solution depends on what you exactly want to achieve. What you propose is not that bad solution - if you NEED index, I see (at the moment) no other algorithmic way.

Alternative solution would have to introduce another form of unique id per button - e.g. you could post pointer to button instead of index, or you could use UUID and Index....

Mirek
