Subject: How to dynamically add remove tabs
Posted by nicesai on Mon, 25 Sep 2023 18:20:30 GMT
View Forum Message <> Reply to Message

I am doing something like this:

```
   void addTab() {
 WithTabLay<ParentCtrl> *tabLay  = new WithTabLay<ParentCtrl>();
       CtrlLayout(*tabLay);
       tabLay->HSizePosZ(0, 0).VSizePosZ(0, 0);
       tabs.Add(*tabLay, MyImgs::add20(), "Tab1");

       tabLay->closeBut.WhenPush = [=] {
        tabs.Remove(*tabLay);
        tabLay->~WithTabLay();
        delete tabLay;
       };
   }
```

As you can guess,
WithTabLay is the tab layout created in the UI editor.
closeBut is a button within the tab that I want to use to remove the tab.

Question-1: Is it necessary to call the destructor after removing the tab?

```
tabLay->~WithTabLay();
```

Question-2: Is it necessary to delete the allocated memory? This line causes the applicatton to crash. So probably I should not, if so, how can I free the allocated memory?

```
delete tabLay;
```

Questions-3: Is there any better way to do this without using pointers? If I create the Tab on stack, its free'ed at the end of the function (from RAII I suppose). And the Tab will not be visible.

Subject: Re: How to dynamically add remove tabs
Posted by Oblivion on Tue, 26 Sep 2023 22:46:41 GMT
View Forum Message <> Reply to Message

Hello nicesai,

Quote:Question-1: Is it necessary to call the destructor after removing the tab?
tabLay->~WithTabLay();

No, not unless you want to manually destroy the pane object (tablay).

Quote:Question-2: Is it necessary to delete the allocated memory? This line causes the applicatton to crash. So probably I should not, if so, how can I free the allocated memory?
delete tabLay;

Well, if you manually allocate the widget on the heap (something we usually avoid in U++, it is not a good practice), yes you need to delete the object when you're done with it (or it will leak)

Quote:Questions-3: Is there any better way to do this without using pointers? If I create the Tab on stack, its free'ed at the end of the function (from RAII I suppose). And the Tab will not be visible.

In general, if we want to create multiple (a lot of, or indeterminate number of) tabs, we'd usually use containers (e.g. could be Array<WithTabLay<ParentCtrl>>, in this case). Containers take care of mem alloc/dealloc. THAT is the way to go.

An example (just to give you an idea. Not the only way to do it)

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

using namespace Upp;

struct MyApp : TopWindow {
 TabCtrl tabs;
 Array<Button> panes;
 Button bt;
 MyApp()
 {
  Title("Tabs test");
  Sizeable().Zoomable().CenterScreen().SetRect(0, 0, 800, 600);
  Add(tabs.HSizePosZ().VSizePosZ(0, 28));
  Add(bt.SetLabel("Add New Tab").HSizePosZ().BottomPosZ(2, 24));

  auto AddPane = [this] {
   int n = tabs.GetCount();
   Button& b = panes.Add();
   b.SetLabel("Pane " + AsString(n) + " (click to delete)");
   b << [this] { // DelPane
    panes.RemoveIf([this](int i) { return &panes[i] == tabs.GetItem(~tabs).GetSlave(); });
    tabs.Remove(~tabs);
   };
   tabs.Add(b.SizePos(), "Tab " + AsString(n));
  };

  bt << [&] { AddPane(); };
```

```
  for(int i = 0; i < 5; i++)
   AddPane();

 }
};

GUI_APP_MAIN
{
 MyApp().Run();
}
```

Best regards,
Oblivion

Subject: Re: How to dynamically add remove tabs
Posted by nicesai on Fri, 29 Sep 2023 19:36:32 GMT
View Forum Message <> Reply to Message

well.

The following part was non intuitive, probably couldn't have guessed at first glance looking at the APIs, but glad I asked and thanks for showing it.
panes.RemoveIf([this](int i) { return &panes[i] == tabs.GetItem(~tabs).GetSlave();

Subject: Re: How to dynamically add remove tabs
Posted by Oblivion on Sat, 30 Sep 2023 15:14:09 GMT
View Forum Message <> Reply to Message

By the way,

The TabCtrl is a bit rudimentary widget. If you need a fully-fledged tab managing, I suggest you try TabBar instaed. It has a ton of features and specialized versions of it (It is basically the tab widget used in TheIDE, editor).

See the example: https://www.ultimatepp.org/reference$FileTabsExample$en-us.h tml

The example demonstrates the capabilities of TabBar package.
