Subject: Creating sub menus without callback functions
Posted by prog13 on Mon, 23 Oct 2006 22:38:44 GMT

View Forum Message <> Reply to Message

Hi, I've stumbled across U++ some days ago while searching for an alternative GUI framework because of some issues I have with wxWidgets - and all in all I really liked what I saw. Now after beginning to port an application that reads in it's user interface layout from a text file I've got stuck on menu creation. The layout parser of that application needs to create the menu structure in standard depth-first-order or else some kind of overly complicated menu structure buffer would need to be implemented.
So what I'd need is instead of adding a sub menu with "bar.Add(caption, callback);" something like "bar.Add(caption, submenu);". But this doesn't seem to work with 'submenu' beeing of type MenuBar.
So is something like that possible? Or any method not using (seemlingly asynchronous) callback functions?

Subject: Re: Creating sub menus without callback functions
Posted by mirek on Tue, 24 Oct 2006 00:32:42 GMT

View Forum Message <> Reply to Message

No, current menu is designed to use callback.

BUT callbacks are quite versatile. This problem can be easily solved by passing additional parameter in the callback.

E.g. like this: (very barebone skeleton, just to get the idea):

```
struct MyApp {
    typedef MyApp CLASSNAME;

    Vector<String> top;
    Vector< Vector<String> > submenu;

    void Action(int top, int i) { /* some action */ }

    void SubMenu(Bar& bar, int top) {
      const Vector<String>& sm = submenu[top];
      for(int i = 0; i < sm.GetCount(); i++)
         bar.Add(sm[i], THISBACK2(Action, top, i));
    }

    void MainMenu(Bar& bar) {
      for(int i = 0; i < top.GetCount(); i++)
         bar.Add(top[i], THISBACK1(SubMenu, i));
    }
```

```
    ....
};
```

Subject: Re: Creating sub menus without callback functions
Posted by prog13 on Tue, 24 Oct 2006 11:09:46 GMT
View Forum Message <> Reply to Message

Ok thanks for the reply, I'm now saving the whole menu structure in a tree container and then using callbacks on that. It works now, although I had to work around THISBACK2(...)/callback2(...) using a helper struct. THISBACK2 doesn't seem to work (v610) - at least in this context. To verify this I modified the menu reference example:

```
 void SubMenu(Bar& bar, int a, int b)
 {
  for(int i = 0; i < 10; i++)
   bar.Add(AsString(i), THISBACK1(ShowNumber, i));
 }


 void Menu(Bar& bar)
 {
  bar.Add("Enable numbers", THISBACK(EnableNumbers))
    .Check(numbers_enabled);
  bar.Add(numbers_enabled, "Numbers", THISBACK2(SubMenu,1,2));
  bar.Add("Exit", THISBACK(Exit))
    .Key(K_CTRL_E);
 }
```

This gives the following compiler errors:

MSVC8:
C:\upp\reference\Menu\menu.cpp(33) : error C2780: 'Callback callback2(Callback2<P1,P2>,T1
,T2)' : expects 3 arguments - 4 provided
      C:\upp\uppsrc\Core/Callback.h(202) : see declaration of 'callback2'
C:\upp\reference\Menu\menu.cpp(33) : error C2780: 'Callback callback2(R (__cdecl *)(A,B),
 T1,T2)' : expects 3 arguments - 4 provided
      C:\upp\uppsrc\Core/Callback.h(197) : see declaration of 'callback2'
C:\upp\reference\Menu\menu.cpp(33) : error C2784: 'Callback callback2(const Object *,R (_
_thiscall O::*)(A,B) const,T1,T2)' : could not deduce template argument for 'R (__th
iscall O::*)(A,B) const' from 'void (__thiscall App::*)(Bar &,int,int)'
      C:\upp\uppsrc\Core/Callback.h(181) : see declaration of 'callback2'
C:\upp\reference\Menu\menu.cpp(33) : error C2784: 'Callback callback2(Object *,R (__thisc
all O::*)(A,B),T1,T2)' : could not deduce template argument for 'R (__thiscall O::*
```

)(A,B)' from 'void (__thiscall App::* )(Bar &,int,int)'
    C:\upp\uppsrc\Core/Callback.h(175) : see declaration of 'callback2'

GCC(MinGW):
C:\upp\reference\Menu\menu.cpp: In member function `void App::Menu(Bar&)':
C:\upp\reference\Menu\menu.cpp:33: error: no matching function for call to `callback2(App
 * const, void (App::*)(Bar&, int, int), int, int)'

Despite this working for me now and most smaller applications wouldn't read their interfaces from a file at runtime anyways, I  think it might be worth it to allow for a more "procedural" approach of menu creation for such situations - I haven't investigated the needed changes, so it could be too much work for the benifit.

---

## Subject: Re: Creating sub menus without callback functions
Posted by mirek on Tue, 24 Oct 2006 14:45:31 GMT
View Forum Message <> Reply to Message

Quote:
THISBACK2 doesn't seem to work (v610) - at least in this context.

Sorry about that, that variant really is not part of U++ (yet?)

Quote:
Despite this working for me now and most smaller applications wouldn't read their interfaces from a file at runtime anyways, I  think it might be worth it to allow for a more "procedural" approach of menu creation for such situations - I haven't investigated the needed changes, so it could be too much work for the benifit.

Well, the needed changes are exactly what you have done now. I think that callback based menu should be primary, because it is more generic - you can quite easily implement non-callback menu using callback variant, but opposite is not true.

Mirek

---

## Subject: Re: Creating sub menus without callback functions
Posted by slashupp on Sun, 29 Nov 2015 05:07:50 GMT
View Forum Message <> Reply to Message

... AND.. about nine years later...

bar.Add("Dummy Item", Callback(nullptr));
.. will do the trick ;)