Subject: Basic question regarding callbacks
Posted by 281264 on Mon, 21 Jun 2010 11:47:59 GMT
View Forum Message <> Reply to Message

I am completely new in Ultimate++, so one thousand apologizes if I ask very basic questions.

contained in the GUI Tutorial, which I am copying here for convenience:

```
#include <CtrlLib/CtrlLib.h>
using namespace Upp;
struct MyAppWindow : TopWindow {

   MenuBar menu;

   void Exit() {
      if(PromptOKCancel("Exit MyApp?"))
         Break();
   }

   void SubMenu(Bar& bar) {
      bar.Add("Exit", THISBACK(Exit));
   }

   void MainMenu(Bar& bar) {
      bar.Add("Menu", THISBACK(SubMenu));
   }

   typedef MyAppWindow CLASSNAME;

   MyAppWindow() {
      Title("My application with menu").Sizeable();
      AddFrame(menu);
      menu.Set(THISBACK(MainMenu));
   }

};

GUI_APP_MAIN
{
   MyAppWindow app;
   app.Run();
}
```

Question 1: What is really happening when menu.Set(THISBACK(MainMenu)); is executed?.

Question 2: I guess that menu.Set is the method void Set(Callback1<Bar&> menu) described in

the Menu Class help; if that is ok, the expected argument is a Callback1, which subsequently ask for TWO arguments [as indicated in template <class Object, class M, class P, class T> Callback callback1(Object *object, void (M::*method)(P), T arg)]. Why are we passing only the methid MainMenu and not both the ManiMenua and the expected argument menu?.

Question 3: How is MainMenu executed from menu.Set(THISBACK(MainMenu)) if we do not pass the argument bar?.

Question 4: I tried to debug the code and added some breakpoints (for example in manu.Set(..). But the debugger do not work and enters in an infinite loop; what is happening?

Sorry again for my clumsiness and my lack of knowledge of C++.

Best wishes and many thanks.

Subject: Re: Basic question regarding callbacks
Posted by cbpporter on Mon, 21 Jun 2010 12:12:13 GMT
View Forum Message <> Reply to Message

Q1: To understand what is happening there you need to understand function pointers and delegates. If you have a function pointer, you can then call that function through the pointer and transfer execution to an arbitrary piece of code. But in OOP, you can't have a pointer to a method of the object. Calling the method has no meaning if you don't know on what object to call it. This is where delegates come in. Basically, a delegate is an ordered pair formed from an object and a method of it that is going to get called. This is a very superficial explanation, but I hope that if will get you started.

THISBACK is just a shortcut to create a delegate where the object is "this".

So when you are using the Set method, that method will call the method MainMenu of your MyAppWindow instance (your current this) and it will pass it a Bar object, which in you case is the "menu" variable. In MainMenu method, you add the top level menus to you menubar, and pass them additional callbacks which get called for each item. This seems confusing at first because you are using the same mechanism both to construct menus (structure) and to define behavior. The difference is only in you intention, there are not separate mechanisms.

Q2: Since you are dealing with callbacks/delegates, not function pointers, it is normal to have one extra argument there for the object which has the method. So if first argument is the object, and the second is your intended argument, you get a Callback1. And you are not passing that argument manually because Set does that.

Q3: The argument bar is passed by the Set method and in your case it is "menu", an instance of MenuBar which is in the hierarchy that allows it to be passed as an argument to that function. The type is Bar as a common ancestor so you can use the same methods to construct both menubars and toolbars (and theoretically any kind of controls that are compatible with a Bar: one could imagine a TabCtrl with the same interface or an "Outlook" kind o pane control).

Q4: I don't know exactly where you get the loop, but it is normal to eventually enter an infinite loop. This is how GUI application work. They wait in a loop for user or system messages.

Subject: Re: Basic question regarding callbacks
Posted by 281264 on Mon, 21 Jun 2010 15:33:09 GMT
View Forum Message <> Reply to Message

Thank you. With your explanation the matter is clearer.

I have some additional questions..

1.- "delgates"..I was not aware about the existence of this. is it standard C++?.

2.- debugging: what I mean is if I insert a breakpoint in, shall we say, Break(); the programs does stop but the yellow arrow in the debugger does not appear.

Additional note: when debugging a simple C++ console application, the debugger shows trash values (i.e. non sense values) of many variables. What is the reason of this???. Sometimes I have the sensation that the debugger (GCC) does not properly.

Best wishes.

Subject: Re: Basic question regarding callbacks
Posted by mrjt on Mon, 21 Jun 2010 16:24:44 GMT
View Forum Message <> Reply to Message

- delegates:
For a global function (as you would have in C) you need only a pointer to the function to call it. If the function is a member function of a class (as you might get in C++) you also need to know which object to call the function against and so you must additionally store an object pointer. This combination is a 'delegate', but it's just a name for a common type of structure.

-debugging
GCC breakpoints (in TheIDE) don't work if the program is running, it's a real pain. You need to set them before execution or when the debugger has already stopped.

To inspect variables you need to use the immediate window (Ctrl+Q while debugging). I tend to do my debugging on Windows first for these reasons.

Subject: Re: Basic question regarding callbacks
Posted by 281264 on Mon, 21 Jun 2010 16:59:12 GMT

View Forum Message <> Reply to Message

Thank you.

Additional note: when debugging a simple C++ console application, the debugger shows trash values (i.e. non sense values) of many variables. What is the reason of this???.

Concerning the debugger I am attaching a screenshot showing a debugging process of a simple C++ console program. The programs is stopped after the assignment of a value of 10 (integer) to

output: it is nonsense!.

File Attachments

1) GCC debugger.doc, downloaded 400 times

Subject: Re: Basic question regarding callbacks
Posted by dolik.rce on Mon, 21 Jun 2010 18:39:14 GMT
View Forum Message <> Reply to Message

Hi,

This looks like a compiler optimization. The compiler recognizes that value of a is actually not used before you assign it the result of that call to p(10,20).

Just to make sure: What version of U++ are you using? You can find it in Assist>About... Since you said you are using mingw which is not distributed with U++ anymore, I guess you have version 605, am I right? If so, please upgrade to something newer. Just have a look in the Download section for links Also, if you downloaded that ancient mingw release, you should consider upgrading the mingw to newer version (have a look in this thread).

Bye,
Honza

PS: Could you use some other file format than .doc next time? Some of us don't have Office/openoffice installed Pdf or any image format would be nice, thanks.

Subject: Re: Basic question regarding callbacks
Posted by 281264 on Mon, 21 Jun 2010 20:32:26 GMT
View Forum Message <> Reply to Message

Thank you.

However the problem persists. Regardless of the location of the breakpoints, the debugger is not

The version of U++ I am using is the latest one. GCC is my compiler (I was not aware about MinGW is not loner used).
