Subject: THISBACK and inheritance
Posted by rafiwui on Mon, 03 Jul 2017 14:15:22 GMT
View Forum Message <> Reply to Message

I had an interesting issue with a MenuBar and a THISBACK.
I made three classes: MenuBarWindow, FnWindow and MyWindow.
 - MenuBarWindow only has the MenuBar attribute and the method I want as its callback.
 - FnWindow inherits from MenuBarWindow and has some buttons for the F1-F8 keys and the corresponding callback methods.
 - MyWindow inherits from FnWindow and is the main window in my application.

Here is an example code:

```
class MenuBarWindow : public TopWindow
{
protected:
    MenuBar m_menuBar;

public:
    typedef MenuBarWindow CLASSNAME;
    MenuBarWindow(int _sizeX, int _sizeY);

protected:
    virtual void MenuBarMain(Bar& _bar) = 0;
};

MenuBarWindow::MenuBarWindow(int _sizeX, int _sizeY)
{
    SetRect(0, 0, Zx(_sizeX), Zy(_sizeY + m_menuBar.GetStdHeight(m_menuBar.GetFont())));
    AddFrame(m_menuBar);
    m_menuBar.Set(THISBACK(MenuBarMain));
}


class FnWindow : public MenuBarWIndow
{
protected:
    Button b_fn1;
    // ...
    Button b_fn8;

public:
    typedef FnWindow CLASSNAME;
    FnWindow(int _sizeX, int _sizeY);

private:
    virtual void OnClickFn1() = 0;
    // ...
```

```
    virtual void OnClickFn8() = 0;
};

FnWindow::FnWindow(int _sizeX, int _sizeY)
    : MenuBarWindow(_sizeX, _sizeY)
{
    // Adding buttons here...

    b_fn1 <<= THISBACK(OnClickFn1);
    // ...
    b_fn8 <<= THISBACK(OnClickFn8);
}

// MyWindow header...
MyWindow::MyWindow(int _sizeX, int _sizeY)
    : FnWindow(_sizeX, _sizeY)
{
    // do sth...
}
```

In the MyWindow files are declarations and definitions for the virtual methods.

If I run this the program stops and shows me this and exits the application when you continue:

If I remove virtual void MenuBarMain(Bar& _bar) = 0; and move m_menuBar.Set(THISBACK(MenuBarMain)); from MenuBarWindow to MyWindow everything works fine.

At first I thought this is because the method is pure virtual and needs an implementation in the class from where I call it or use THISBACK but then I should get this strange behaviour after my changes as well because I do the same with the button callbacks but they work perfectly.

So does anyone has an idea why this is happening? Is this a generell C++ thing or is it a U++ thing?

File Attachments

1) UppCallbackProblem.PNG, downloaded 729 times

Subject: Re: THISBACK and inheritance
Posted by mirek on Wed, 05 Jul 2017 12:07:18 GMT
View Forum Message <> Reply to Message

rafiwui wrote on Mon, 03 July 2017 16:15I had an interesting issue with a MenuBar and a THISBACK.
I made three classes: MenuBarWindow, FnWindow and MyWindow.
 - MenuBarWindow only has the MenuBar attribute and the method I want as its callback.

- FnWindow inherits from MenuBarWindow and has some buttons for the F1-F8 keys and the corresponding callback methods.
- MyWindow inherits from FnWindow and is the main window in my application.

Here is an example code:

```
class MenuBarWindow : public TopWindow
{
protected:
    MenuBar m_menuBar;

public:
    typedef MenuBarWindow CLASSNAME;
    MenuBarWindow(int _sizeX, int _sizeY);

protected:
    virtual void MenuBarMain(Bar& _bar) = 0;
};

MenuBarWindow::MenuBarWindow(int _sizeX, int _sizeY)
{
    SetRect(0, 0, Zx(_sizeX), Zy(_sizeY + m_menuBar.GetStdHeight(m_menuBar.GetFont())));
    AddFrame(m_menuBar);
    m_menuBar.Set(THISBACK(MenuBarMain));
}


class FnWindow : public MenuBarWIndow
{
protected:
    Button b_fn1;
    // ...
    Button b_fn8;

public:
    typedef FnWindow CLASSNAME;
    FnWindow(int _sizeX, int _sizeY);

private:
    virtual void OnClickFn1() = 0;
    // ...
    virtual void OnClickFn8() = 0;
};

FnWindow::FnWindow(int _sizeX, int _sizeY)
    : MenuBarWindow(_sizeX, _sizeY)
{
    // Adding buttons here...
```

```
    b_fn1 <<= THISBACK(OnClickFn1);
    // ...
    b_fn8 <<= THISBACK(OnClickFn8);
}

// MyWindow header...
MyWindow::MyWindow(int _sizeX, int _sizeY)
    : FnWindow(_sizeX, _sizeY)
{
    // do sth...
}
```

In the MyWindow files are declarations and definitions for the virtual methods.

If I run this the program stops and shows me this and exits the application when you continue:

If I remove virtual void MenuBarMain(Bar& _bar) = 0; and move m_menuBar.Set(THISBACK(MenuBarMain)); from MenuBarWindow to MyWindow everything works fine.

At first I thought this is because the method is pure virtual and needs an implementation in the class from where I call it or use THISBACK but then I should get this strange behaviour after my changes as well because I do the same with the button callbacks but they work perfectly.

So does anyone has an idea why this is happening? Is this a generell C++ thing or is it a U++ thing?

I see nothing wrong. Please, as this really looks like a small package experiment, follow the suggestion and compress the whole package and attach it to the post so that we can test it quickly.

Mirek

Subject: Re: THISBACK and inheritance
Posted by rafiwui on Thu, 06 Jul 2017 07:53:52 GMT
View Forum Message <> Reply to Message

Ok here are the packages:
One with the bug and the other one working so you can see how it should look like.

File Attachments
1) TestApplication.7z, downloaded 297 times
2) TestApplicationWorking.7z, downloaded 292 times

Subject: Re: THISBACK and inheritance
Posted by Oblivion on Thu, 06 Jul 2017 18:25:26 GMT
View Forum Message <> Reply to Message

Hello,

It is a "C++ thing." :)
As far as I can see, you are calling a pure virtual method (or storing its address -which is undefined at that moment) from base constructor.
Calling a pure virtual method from base constructor is undefined behaviour. (Since at that point the vtable entry for the said method is not yet assigned. It is basically "0" (undefined)). That's why you dont't even get an explicit error message.

```
MenuBarWindow::MenuBarWindow(int _sizeX, int _sizeY)
{
   SetRect(0, 0, Zx(_sizeX), Zy(_sizeY + m_menuBar.GetStdHeight(m_menuBar.GetFont())));
   AddFrame(m_menuBar);
   m_menuBar.Set(THISBACK(MenuBarMain)); // <-- Undefined behaviour.
}
```

Why did you define a menu handler virtual anyway? We are using callbacks to define menus, so you won't really need virtual functions for this purpose.
if you want to store m_menuBar member in MenuBarWindow class that's fine, but since it is a protected member, it can be accessed from the derived classes:

```
MyWindow::MyWindow(int _sizeX, int _sizeY)
   : FnWindow(_sizeX, _sizeY)
{
       AddFrame(m_menuBar);                          // This should
 m_menuBar.Set(THISBACK(MenuBarMain));           // work.
}
```

Also I see that you use some obscure internal macro(s) explicitly. You dont need them at all.

Best regards,
Oblivion

Subject: Re: THISBACK and inheritance
Posted by rafiwui on Thu, 06 Jul 2017 20:41:47 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Thu, 06 July 2017 20:25

As far as I can see, you are calling a pure virtual method (or storing its address -which is undefined at that moment) from base constructor.
Calling a pure virtual method from base constructor is undefined behaviour. (Since at that point the vtable entry for the said method is not yet assigned. It is basically "0" (undefined)). That's why you dont't even get an explicit error message.

But why is it working for the OnClickButton callbacks in the FnWindow?

Oblivion wrote on Thu, 06 July 2017 20:25

Also I see that you use some obscure internal macro(s) explicitly. You dont need them at all.

What macros do you mean and how can I avoid them?

Thanks,
Daniel

Subject: Re: THISBACK and inheritance
Posted by Oblivion on Thu, 06 Jul 2017 21:17:03 GMT
View Forum Message <> Reply to Message

Hello Daniel,

Quote:
But why is it working for the OnClickButton callbacks in the FnWindow?

It is because those button events aren't called immediately. They are called when there is some user action (They are called WHEN there is action (click/push, etc.).
Had they been called immediately you would have the same undefined behaviour.
On the other hand, MenuBar::Set() method calls the provided callback, and sets the menubar immediately, which leads to crash.

Quote:

What macros do you mean and how can I avoid them?

Ah, that's my fault. Nevermind. (For some reason, the below macros reminded me of U++ internal

macros used in the TheIDE layout editor.)

```
#define _FN_LABEL(button, label)    button.SetLabel(label).Enable(true)
#define FN_LABEL(number, label)     _FN_LABEL(b_fn##number, label)

#define _FN_UNLABEL(button) button.SetLabel("").Enable(false)
#define FN_UNLABEL(number)  _FN_UNLABEL(b_fn##number)
```

Best regards,
Oblivion.

Subject: Re: THISBACK and inheritance
Posted by rafiwui on Fri, 07 Jul 2017 06:40:23 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Thu, 06 July 2017 23:17Hello Daniel,

It is because those button events aren't called immediately. They are called when there is some user action (They are called WHEN there is action (click/push, etc.).
Had they been called immediately you would have the same undefined behaviour.
On the other hand, MenuBar::Set() method calls the provided callback, and sets the menubar immediately, which leads to crash.

Ah okay. Thanks for your answers.

Oblivion wrote on Thu, 06 July 2017 23:17Hello Daniel,

Ah, that's my fault. Nevermind. (For some reason, the below macros reminded me of U++ internal macros used in the TheIDE layout editor.)

Oh yeah they are pretty similar :lol:
