Subject: Submenu Clickable
Posted by Qwak on Tue, 12 Mar 2019 09:01:46 GMT
View Forum Message <> Reply to Message

Hi everyone,

I have a tray icon with a list of items and one of them is also a list like this :

I would like to be able to click on the 1st Pause.

```
bar.Sub(true, "Pause", THISBACK(PauseMenu));

void AOO::PauseMenu(Bar& bar) {
 bar <<= THISBACK(BreakOut);
	bar.Add("Pause", myFuncBreak);
	bar.Add("Réunion", myFuncMeeting);
}
```

This isn't working and if i call BreakOut() in PauseMenu() then the event/function is fired when submenu pop-out.

Any suggestion, idea or clues is welcome.

Regards,

Qwak

File Attachments

1) tray.png, downloaded 491 times

Subject: Re: Submenu Clickable
Posted by Oblivion on Tue, 12 Mar 2019 11:50:07 GMT
View Forum Message <> Reply to Message

Hello Qwak, and welcome to the U++ forums!

A simple example might be better than describing it (This works on Linux, at the moment I don't have access to Windows)

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

```
using namespace Upp;

class MenuTest : public TopWindow {
 TrayIcon tray;


public:
 typedef MenuTest CLASSNAME;
 void MainMenu(Bar& bar)
 {
  bar.Sub("Pause", THISFN(PauseMenu));
  bar.Separator();
  bar.Add("Exit", [&]{ tray.Break(); });
 }
 void PauseMenu(Bar& bar)
 {
               bar.Add("Pause", THISFN(Func1));
               bar.Add("Réunion", THISFN(Func2));
 }

 void Func1()
 {
  PromptOK("Pause menu entry invoked!");
 }
 void Func2()
 {
  Exclamation("Meeting menu entry invoked!");
 }
 void Do()
 {
  tray.Icon(CtrlImg::Diskette());
  tray.WhenBar = THISFN(MainMenu);
  tray.Run();
 }
};

GUI_APP_MAIN
{
 MenuTest().Do();
}
```

Note that this menu bar can also be used as context menu, just pass it to the relevant function.

Best regards,
Oblivion

Subject: Re: Submenu Clickable
Posted by Qwak on Tue, 12 Mar 2019 13:21:32 GMT

View Forum Message <> Reply to Message

Hey thanks for replay Oblivion !

I successfully did what you do. What i want to do is :

```
void MainMenu(Bar& bar)
 {
  bar.Sub("Pause", THISFN(PauseMenu)); // There we can't call another function on click here ?
  bar.Separator();
  bar.Add("Exit", [&]{ tray.Break(); });
 }
```

My apologies if i do not explain myself clearly.

Best regards,

Qwak.

Subject: Re: Submenu Clickable
Posted by Oblivion on Tue, 12 Mar 2019 13:53:38 GMT

View Forum Message <> Reply to Message

> Quote: What i want to do is :

I see. AFAIK, there is no direct way to achieve that. It maybe possible with some workaround but I never tried such a thing. :)

But what is the point of it anyway?

On the other hand, if you need to alternate between a menu item and a sub menu, for example a "pause" submenu and a clickable "continue" menu item, you can achieve that easily. (If this is what you want to achieve.)

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

using namespace Upp;

class MenuTest : public TopWindow {
 TrayIcon tray;
 bool     paused;
public:
 typedef MenuTest CLASSNAME;

 MenuTest()
 {
  paused = false;
 }
 void MainMenu(Bar& bar)
 {
  if(paused) bar.Add("Continue", [&] { paused = false; });
  else bar.Sub("Pause", THISFN(PauseMenu));
  bar.Separator();
  bar.Add("Exit", [&]{ tray.Break(); });
 }
 void PauseMenu(Bar& bar)
 {
	bar.Add("Pause", THISFN(Func1));
	bar.Add("Réunion", THISFN(Func2));
 }
 void Func1()
 {
  PromptOK("Pause menu entry invoked!");
  paused = true;
 }
 void Func2()
 {
  Exclamation("Meeting menu entry invoked!");
  paused = true;
 }
 void Do()
 {
  tray.Icon(CtrlImg::Diskette());
  tray.WhenBar = THISFN(MainMenu);
  tray.Run();
 }
};

GUI_APP_MAIN
{
 MenuTest().Do();
}
```

Best regards,
Oblivion

Subject: Re: Submenu Clickable
Posted by Qwak on Tue, 12 Mar 2019 14:08:32 GMT

View Forum Message <> Reply to Message

Hey Oblivion thanks again, i had what you proposed.

i finnaly came up with :

```
void TrayMenu(Bar& bar) {
 bar.Add(connect ? "Deconnexion":"Connexion automatique",
THISBACK(SwitchConnect)).Enable(!error);
 if (connect)
  bar.Add("Pause", THISBACK(PauseMenu)).Enable(connect);
 else
  bar.Add("Pause", THISBACK(BreakOut)).Enable(!error);
 bar.Add("About", THISBACK(About));
 bar.Add("Quitter", THISBACK(Exit));
}

void PauseMenu(Bar& bar) {
 for(Argument &b : breaks) {
        bar.Add(b.ArgValue, [=] {
  currentBreak != std::stoi(b.ArgName.ToStd()) ? BreakIn(std::stoi(b.ArgName.ToStd())) :
BreakOut();
        }).Check(currentBreak == std::stoi(b.ArgName.ToStd()));
    }
}
```

it actually works fine but i feel like it would be better to be able to open and click on the same itemMenu.

Thanks Oblivion for brainstorming with me.

Qwak.