Subject: Form Designer
Posted by gedumer on Thu, 04 Feb 2010 19:37:25 GMT
View Forum Message <> Reply to Message

I've used other form designers, but I just can't figure this one out. How do I create a menu? How do I add an image to a button? Where do I get images from? Can I load them from a file or do I have to create everything in the image editor? Can I import images into the editor? Do you have "resource" files for images? Where do I handle events? Where are all the properties for the controls? For example... Button ony has "text", "tip", and "font". What about the rest of it? I just can't find things.

Help?

Subject: Re: Form Designer
Posted by nlneilson on Thu, 04 Feb 2010 21:43:54 GMT
View Forum Message <> Reply to Message

Download the GUI Tutorial from this thread:
http://www.ultimatepp.org/forum/index.php?t=msg&th=597&a mp;a mp;start=0&
or try:
http://upp.sourceforge.net/srcdoc$CtrlLib$Tutorial$en-us.htm l

Also try the code in the directories:
tutorial, examples and bazaar

You can look at this thread for a menu:
http://www.ultimatepp.org/forum/index.php?t=msg&th=4891& amp;start=0&

Subject: Re: Form Designer
Posted by gedumer on Thu, 04 Feb 2010 22:06:23 GMT
View Forum Message <> Reply to Message

Yes... I looked at all those. They don't answer my questions about the form designer or the image editor. The tutorials show you how to write the code to do a couple of those things, but not how to do them in the form designer.

In the first place, how do you start the form designer. There isn't even a mention of it in any of the menus. The same is true of the image editor. If you open an existing app that contains a .lay file, then you can get to the form designer by opening the .lay file, but that's not intuitive. Likewise for the image editor with .iml files. My question is: how do you invoke those functions if your new package doesn't already contain those files?

The tutorials also fail to answer my question regarding importing images from a file, nor do they address the "resource" file issue. I didn't ask my questions out of total ignorance, I did try to find the answers, but they have yet to be forthcoming. That's why I asked the questions on this forum.

Can someone please help?

Subject: Re: Form Designer
Posted by Didier on Thu, 04 Feb 2010 23:14:55 GMT
View Forum Message <> Reply to Message

Hi !

To open the form designer: add a xxx.lay   layout file in the project.
Select the added file ==> the form designer (==layout editor) opens.

The Layout editor only allows to set the layout of widgets and set (some) properties.
All the rest has to coded but is quite easy though.

To add an image to a button:
myButtonInstance.SetImage(image);

where 'myButtonInstance' comes from the layout designer when you use something like:
class MyClass : public WithMyLayout<TopWindow>

The image can be retreived from internal images by using :
MyImgIml::CLUB()

where 'MyImgIml' is the name of the class associated to an '.iml' file included in the project. But you need to have the associated declarations in you're code

```
//in a header file
#define IMAGECLASS MyImgIml
#define IMAGEFILE <MyProject/MyImlFile.iml>
#include <Draw/iml_header.h>

// in one source file
#define IMAGECLASS MyImgIml
#define IMAGEFILE <MyProject/MyImlFile.iml>
#include <Draw/iml_source.h>
```

Subject: Re: Form Designer
Posted by gedumer on Fri, 05 Feb 2010 02:22:42 GMT
View Forum Message <> Reply to Message

I appreciate the help. Don't you think it would make more sense though, to have a menuitem and/or toolbar item for both the form designer and the image editor? New users have no idea whatsoever that the presence of a .lay file or .iml file is required to invoke the designer and editor

functions. Part of broadening the reach and audience of U++ is its ability to attract and subsequently gain the acceptance of new users when they first try the product. If those new users become discouraged by a perceived apparent complexity or lack of capability of the application as I did, they will simply move on to another product. This tool appears to have merit, but I was about to turn away and move on until I discovered a few hidden secrets about the product through dogged persistence and now from the help of someone on this forum. I'm suddenly encouraged to at least give it a further evaluation period before making a decision on its worthiness.

Thanks.

Subject: Re: Form Designer
Posted by alendar on Fri, 05 Feb 2010 02:57:06 GMT
View Forum Message <> Reply to Message

Took me a while to figure it out, but its pretty cool once you do. Take a look at the examples and its straight forward.

1) insert a lay file into your project. You do that by right clicking in the bottom-left frame and selecting "Insert Package Directory File(s)". That was not intuitive for a while. I tried U++ a few months ago and that pissed me off so much I dumped it and went back to struggling with VC. Now I've been more tolerant of the product and its just "Not Microsoft". Plus its free so I can't expect it to behave like a $3,000 product. In fact IMO its better because I've been able to create some fun code, once I got past the little things.

2) Select the file, which opens it. You may have to press Ctrl+T if you see text. Then right-click in the grid and select an object to add. Again, the right-click to add frustrated me for a few seconds, but I got over it. The rubber band lines annoyed me too, until I figured out they actually made life VERY easy! Simply click on/off to lock it to that side of the window. Very slick.

U++ focuses on the things developers want to generally do, instead of trying to force you to use some new hair-brained framework, like ATL or WTL or MFC or WinForms or Managed C++ and .NET VBX then OCX then COM and OLE2 then ActiveX then I don't remember what.

3) name each item. These automagically become members in the class you include the layout in.

4) Create a window class (use an example). All it is is class W : public WithXXXLayout<TopWindow>.

5) Drop this in your constructor:

CtrlLayout(*this, "Title");

or CtrlLayoutOk or OkCancel.

You do have to be careful about window sizing code being after or before the Layout macro.

6) Then you .Run() the window from your Main.

Total Cake.

Now in your constructor, you type one of your objects and poof - you get a context-sensitive popup of all the bases of your window.

To do a menu:

1) Something like this in your class def:
MenuBar menuFrame; // Horizontal stack

```
void constructMenuFrame() {
 AddFrame(menuFrame);
 menuFrame.Set(THISBACK(menuFrameCallback));
}
void menuFrameCallback(Bar& bar) {
bar.Add("File", THISBACK(FileMenu));
bar.Add("Edit", THISBACK(EditMenu));
bar.Add("View", THISBACK(ViewMenu));
bar.Add("Help", THISBACK(HelpMenu));
void HelpMenu(Bar& bar) {
 bar.Add("About Jammit", THISBACK(About));
}
void About() {
 aboutWindow.Open();      }
```

2) Generate it in your constructor.
constructMenuFrame();

Can't get any easier.
Oh, to get callbacks you'll need this in your window class:
typedef MainWin CLASSNAME;

This is a simple thing that allows TopWindow or some such to generate the proper function call.

Other things.
1) Drop a file with extension ".rc" in there and U++ picks it up.
2) Drop an IML file in there and all your images are available.

I put this in my project share header:

```
#define IMAGECLASS MyImages
#define IMAGEFILE "project.iml"
#include <Draw/iml.h>
```

Then to set the icon I just go "this->LargeIcon(MyImages::bigmainicon);" in my window class.

That sets the Taskbar icon. The regular app icon I set in the resource file.

I threw a little Icon on a push button with:

playOrPause = MyImages::play();

On clicking I do "playOrPause = MyImages::pause()"

One learning curve issue is that the writers overrode more operators than a new rooster in a henhouse. The "~" was a surprise to me, but its fine if you just Alt+O to it and view the implementation.

Callbacks are so easy I'm very impressed. Qt's method seems much more complex.

One thing obviously better than MSVC: You can easily extend anything. I extended TopWindow without blinking. A few fanagles with default constructors, the close method, and such.

Now I have my own little custom Window wrapper. Try doin that in VC. Subclassing is a major effort.

Subject: Re: Form Designer
Posted by gedumer on Fri, 05 Feb 2010 03:43:15 GMT
View Forum Message <> Reply to Message

Thanks... that's a lot to look at. I guess I was just expecting too much out of the forms designer. It's probably never going to be like Delphi or Lazarus, et.al.

Subject: Re: Form Designer
Posted by cbpporter on Fri, 05 Feb 2010 08:13:01 GMT
View Forum Message <> Reply to Message

Hi gedumer,

You are right about some of these tools being counterintuitive, but this is not just the sign of less then perfect GUI design of the tools, but a reflection of different kind of objects and work flow.

Both the image designer and layout designer don't save some binary files, edit your source code or otherwise magically do the stuff you want. They edit the ".lay" and ".iml" files, which are C++ code. So opening up the editor without an associated file doesn't make that much sense. I guess you could open a new empty document and use "Save As" to save it to disk to be more like other tools, but I don't know if it is worth the effort.

Menus are a different subject. This is not like Delphi, where every menu item has its own variable. You only have access to the Menubar object, and you create a menu with callbacks. also, you do not add the main menu in the designer, you add it as a frame in code. Frames are hard to understand initially, but they offer huge benefits. Frames reduce the active surface of any widget

and insert there another widget. This means that you can add a menu, a border or pretty much anything you desire to any widget, and the widget doesn't have to know about the frame, or take measures to update its size and behavior.

So the layout editor should not be used for menus and toolbars.

Also, the layout editor is a little bit poor on features and doesn't offer all the options that you would like. I think this is intentional, but I don't agree with this. Right now you set between 60% and 90% of properties in the editor, depending on widget, and the rest in code. This may or may not be improved in the future.

Also important, you must right click on the layout editor to insert a widget. Again, a miss feature, but I don't have time to improve this, and also once you learn it, it won't bother you anymore.

While the layout editor might not be as user friendly, it is very strong. You can apply your layouts to anything, including a Button or EditBox. While this is not that useful for Buttons, it is great with your custom widgets. Great feature that one learns to appreciate and rely upon.

Subject: Re: Form Designer
Posted by mrjt on Fri, 05 Feb 2010 11:52:52 GMT
View Forum Message <> Reply to Message

I believe that the missing properties on ctrls in the layout designer are a result of 2 factors:
- Nobody actually doing the work adding the features
- Because properties are chained on in the .lay file (ie ctrl.SetLabel("Label").LeftPosZ(60, 20)) only properties that return object references can be used. In addition if you don't get the order correct with sub-class properties before base class properties you will get errors.

Both of the are fixable but require someone to put in the time, and I don't think it will be Mirek

---

Subject: Re: Form Designer
Posted by mirek on Fri, 05 Feb 2010 12:34:37 GMT
View Forum Message <> Reply to Message

alendar wrote on Thu, 04 February 2010 21:57Took me a while to figure it out, but its pretty cool once you do. Take a look at the examples and its straight forward.

1) insert a lay file into your project. You do that by right clicking in the bottom-left frame and selecting "Insert Package Directory File(s)". That was not intuitive for a while. I tried U++ a few months ago and that pissed me off so much I dumped it and went back to struggling with VC. Now I've been more tolerant of the product and its just "Not Microsoft". Plus its free so I can't expect it to behave like a $3,000 product. In fact IMO its better because I've been able to create some fun code, once I got past the little things.

Well, I can just imagine reasonable way how to do that otherwise. I mean, I can add "Add layout file " menu item to package local menu, but the only difference would be the file extension... Not a big advantage, considering that the menu is quite long now anyway...

Quote:
I put this in my project share header:
#define IMAGECLASS MyImages
#define IMAGEFILE "project.iml"
#include <Draw/iml.h>

Note: TheIde can generate these for you - right click the editor and choose Insert..

Also note that right-clicking iml image has option to put the image name onto clipboard.

Mirek

Subject: Re: Form Designer
Posted by mirek on Fri, 05 Feb 2010 12:39:27 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Fri, 05 February 2010 03:13
also, you do not add the main menu in the designer, you add it as a frame in code.

Actually, you can. MenuBar can work both as Frame or as normal child widget.

I use it as child in almost all of my commercial apps - if your main window is dialog based, it is usually easier to add menu as child.

Quote:
Also important, you must right click on the layout editor to insert a widget. Again, a miss feature, but I don't have time to improve this, and also once you learn it, it won't bother you anymore.

What do you suggest? Left click? Note that I wanted to avoid "palette" of widgets - too much space on the screen.

(I am serious in asking here. I agree right-click is confusing for newbies).

Subject: Re: Form Designer
Posted by cbpporter on Fri, 05 Feb 2010 13:09:33 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 05 February 2010 14:39

Actually, you can. MenuBar can work both as Frame or as normal child widget.

I use it as child in almost all of my commercial apps - if your main window is dialog based, it is usually easier to add menu as child.

I know that you can add it technically as a child, but you really can't add in in practice. Too much work. Without frame you would need to add resizing in Layout to handle main window resizing. Non-resizable dialogs are dead and a sign of very poor modern GUI design, and so are menus without some kind of frame separating them from the rest of the GUI. If I don't add menus in a frame, how can I add a TopSeparatorFrame without resizing code?

Quote:
What do you suggest? Left click? Note that I wanted to avoid "palette" of widgets - too much space on the screen.

(I am serious in asking here. I agree right-click is confusing for newbies).
Yes, "pallete" would be the only non confusing one. Tabbed pallete like Delpi is two lines wide an can contain hundreds of widgets. But I'm used with right clicking, even if after so much time I need to search for my widgets. But I think all widgets should be added, Option should have a a checked value in the properties editor, etc. Actually copying the Delphi/Visual C# properties editor would be IMO best.

Subject: Re: Form Designer
Posted by mirek on Fri, 05 Feb 2010 15:02:56 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Fri, 05 February 2010 08:09luzr wrote on Fri, 05 February 2010 14:39
Actually, you can. MenuBar can work both as Frame or as normal child widget.

I use it as child in almost all of my commercial apps - if your main window is dialog based, it is usually easier to add menu as child.

I know that you can add it technically as a child, but you really can't add in in practice. Too much work. Without frame you would need to add resizing in Layout to handle main window resizing.

Not sure I understand that point... Why should top-level menu resize any differently than any other widget?

Mirek

Subject: Re: Form Designer
Posted by mrjt on Fri, 05 Feb 2010 16:37:21 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 05 February 2010 12:39What do you suggest? Left click? Note that I wanted to avoid "palette" of widgets - too much space on the screen.

(I am serious in asking here. I agree right-click is confusing for newbies).
Just have a highly-visible button called 'Add Widget' that opens the exact same menu that you get from right-cliking.

cbporter
I know that you can add it technically as a child, but you really can't add in in practice. Too much work. Without frame you would need to add resizing in Layout to handle main window resizing. Non-resizable dialogs are dead and a sign of very poor modern GUI design, and so are menus without some kind of frame separating them from the rest of the GUI. If I don't add menus in a frame, how can I add a TopSeparatorFrame without resizing code?
You can get the same effect by adding the MenuBar as a child ctrl and adding a bottomseparator to it. Layout isn't an issue, just set it to TopPosZ(0, 20).HSizePosZ(0, 0).

Subject: Re: Form Designer
Posted by mirek on Fri, 05 Feb 2010 21:49:39 GMT
View Forum Message <> Reply to Message

mrjt wrote on Fri, 05 February 2010 11:37luzr wrote on Fri, 05 February 2010 12:39What do you suggest? Left click? Note that I wanted to avoid "palette" of widgets - too much space on the screen.

(I am serious in asking here. I agree right-click is confusing for newbies).
Just have a highly-visible button called 'Add Widget' that opens the exact same menu that you get from right-cliking.

At what position?

P.S.: Sorry to be off-topic in newbie foum...
