
Subject: opinions about FOX-TOOLKIT

Posted by [fudadmin](#) on Wed, 07 Dec 2005 19:01:31 GMT

[View Forum Message](#) <> [Reply to Message](#)

have used FOX-TOOLKIT?

<http://www.fox-toolkit.org/>

Subject: Re: opinions about FOX-TOOLKIT

Posted by [fudadmin](#) on Wed, 07 Dec 2005 19:39:30 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quote:

I would like to spend some words about FOX to make an example on why new and light GUI

Personally, FOX is a toolkit good enough to create professional projects, but its structure is too similar to both wxWidgets and Qt ones. I would prefer making a software with wxWidgets rather than using FOX, and another reason is the language bindings which are not updated to the latest releases or are much instable.

Nowadays, new GUI toolkits (everything in the world) must have something new and particular to be used instead of already existing projects, and IMHO, FOX is not a revolutionary project.

More <http://www.lethalman.net/?p=5>

Subject: Re: opinions about FOX-TOOLKIT

Posted by [fudadmin](#) on Thu, 08 Dec 2005 10:32:41 GMT

[View Forum Message](#) <> [Reply to Message](#)

from <http://www.newplanetsoftware.com/jx/compare.php>

Quote:FOX appears to be a widget set rather than an application framework.

FOX messages are not the same as JX messages. FOX defines the messages in the target, which strongly couples each source to it, since each source has to know the appropriate message for each of its targets. This is not as serious as it seems, however, because FOX messages are designed to simulate Objective C's ability to redirect work to other objects. In this case, the source needs to know how to keep the target happy. Note that there are much cleaner ways to achieve this in C++. One method is to define a base class and then instantiate one of several derived classes at run-time to get the desired behavior.

JX, on the other hand, defines messages in the source, which promotes loose coupling since the source simply broadcasts, and each target does whatever is appropriate. This solves a different problem from that of FOX messages. JX messages help implement Model-View-Controller designs.

In addition, FOX messages consist of an integer and a void*, so one needs to cast from a void* to get the associated state. This breaks the type safety provided by C++. JX messages are objects,

so they can be downcast safely.

Subject: Re: opinions about FOX-TOOLKIT
Posted by [mirek](#) on Thu, 08 Dec 2005 12:41:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

BTW, reading through both Jx and FOX, I have seen two "advantages" listed that in fact are disadvantages (and unfortunately, U++ has both :):

- FOX lists as advantage that apps look the same on any platform. Anyway, users require quite opposite (native look&feel)

- Jx (also wxWidgets, etc...) lists as advantage that it is "complete framework". However, that is not felt as advantage for many users...

There is not much we can do about second one (yes, U++ complete platform as well - ok, at least it is a good platform , anyway we can try to fight first problem via chameleon...

Subject: Re: opinions about FOX-TOOLKIT
Posted by [fudadmin](#) on Thu, 08 Dec 2005 12:47:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Thu, 08 December 2005 07:41BTW, reading through both Jx and FOX, I have seen two "advantages" listed that in fact are disadvantages (and unfortunately, U++ has both :):

- FOX lists as advantage that apps look the same on any platform. Anyway, users require quite opposite (native look&feel)

- Jx (also wxWidgets, etc...) lists as advantage that it is "complete framework". However, that is not felt as advantage for many users...

There is not much we can do about second one (yes, U++ complete platform as well - ok, at least it is a good platform , anyway we can try to fight first problem via chameleon...

WideStudio has skinning...

Subject: Re: opinions about FOX-TOOLKIT
Posted by [mirek](#) on Thu, 08 Dec 2005 13:08:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

Chameleon is supposed to be both skinning and adapting technology. The core of it very advanced and simple skinning system, other part will be "detection" system that detects actual look&feel and adjusts U++ skin according to it.

Subject: Re: opinions about FOX-TOOLKIT
Posted by [fudadmin](#) on Thu, 08 Dec 2005 13:58:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Thu, 08 December 2005 08:08Chameleon is supposed to be both skinning and adapting technology. The core of it very advanced and simple skinning system, other part will be "detection" system that detects actual look&feel and adjusts U++ skin according to it.

It will definitely be a hit if it lets the user to choose between different custom skins and system or desktop look.

Subject: Re: opinions about FOX-TOOLKIT
Posted by [jadeite](#) on Sun, 19 Feb 2006 13:45:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

fudadmin wrote on Thu, 08 December 2005 08:58luzr wrote on Thu, 08 December 2005 08:08Chameleon is supposed to be both skinning and adapting technology. The core of it very advanced and simple skinning system, other part will be "detection" system that detects actual look&feel and adjusts U++ skin according to it.

It will definitely be a hit if it lets the user to choose between different custom skins and system or desktop look.

You know, I have been saying that OSX should be the next major dev task that should be completed, but the more I think about, I am changing my mind. I think the chameleon should be the next major dev task that should be finished after the BGAR business is done. I'm guessing there will be some low-level adjustments needed in core code to make this possible. Best to do the chameleon first as it will be a big hit, but while doing chameleon always have in back of your mind x-platform compatibility so you don't have to make major changes while porting to OSX and wince.

Just my thoughts over morning coffee.
Cheers

Subject: Re: opinions about FOX-TOOLKIT
Posted by [mirek](#) on Sun, 19 Feb 2006 14:12:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

In fact, that is exactly the plan:

- new Draw first (so we do not need to reimplement Chameleon for it..)
- then Chameleon
- then, with both available, WinCE and MacOS X

