
Subject: Re: Ideas and priorities for the next development cycle

Posted by [mirek](#) on Tue, 04 Sep 2012 11:19:58 GMT

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

kohait00 wrote on Tue, 04 September 2012 04:28@android support: we should know where we want to go with android?

a) blank canvas with upp typical look and feel (from w32 i.e.), single touch support... should be possible over the NDK

b) android typical appearance in terms of control widgets (way more complicated, implies to properly integrate moultitouch in upp Ctrl.

c) limiting usage of upp to Core stuff, using NDK bridging gap of java to c++. native widgets from android.. not a 'real' integration.. that said: is a 'real' integration possible anyway?

apreciate your ideas..
cheers

Well, a and (then) b are "standard approach".

Anyway, I have to say that recently I have started thinking about c) option, which basically is "wxWidget way". Interesting part then is what makes U++ compelling to me and what of that could be preserved when going "native GUI".

IMO, Core is simple and clear, esp. on android.

Anything that can be compiled without CtrlCore is preservable as well - that includes Draw. It should also be relatively easy to integrate Draw with any native GUI.

Of course, CtrlCore would have to go, CtrlLib would have to be replaced by something. The main issue at that point is how to make CtrlLib cross-platform. Another issue is what are important "U++ style" features of CtrlLib and whether they could be preserved. I believe that those factors are important:

- widgets are independent from GUI. I guess this part might be quite hard to achive, but perhaps not impossible. (Just to make this point clear, it for example means that I can take ArrayCtrl, fill in values, THEN add it to dialog, and THEN open the dialog in GUI. Or in fact, I can use ArrayCtrl and never use it in GUI, just to store and retrieve values). Related issue is that widgets are represented by object variables, not pointers to objects.

- frame system; we have no view class, but we have powerfull ScrollBar that can be added as Frame to any Ctrl. Hopefully could be preserved in some form, but most likely not for native views...

- customization using Display/Convert

- layouts, but those are not as important...

Hard to say, perhaps we might try something like that... Maybe even use something like wxWidgets, encapsulate U++ way and let THEM to deal with platform independency
