Subject: Development ToDo finalized
Posted by mirek on Wed, 05 Oct 2011 20:58:14 GMT
View Forum Message <> Reply to Message

It looks like there are no new suggestion, so here is the complete list of possible future development directions:

- Google NaCl support in theide(IMO, this one would be big for U++) [m]
- Google NaCl U++ backend [q]
- Web development framework (something like Jango in U++ should be piece of cake..) [q]
- Web development support in theide (html/js highlighting, some way how to export these to output directory) [w]
- Android support in theide [m]
- Android U++ backend [q]
- MacOS X support in theide [w]
- Gtk backend (more closely coupled, using Gtk eventloop) [m]
- PdfPainter [w]
- theide: Windows 64-bit debugging support [w]
- theide look&feel improvements (docking, new icons...) [m]

- "U++ embedded" - improving FB rainbow backend and maybe theide too to make U++ a good embeded platform [m]

- configurable MenuBar [w]

- some production quality complex examples for various different topics (MT events handling, client-server or P2P, etc...) (these would differ from simple examples in the way that they would solve everything including handling all error states, etc..) [w]

- JavaScript V8 in U++ [w]
- TheIDE extensions using JavaScript V8 [w]
- event loop/event for windows (libuv) and for unix (libev). [?]

- Painter: ability to render to preexisting buffer [d]
- Painter: multithreaded rendering [d]
- Painter: maybe ability to use different final backend, or some other optimization here to improve performance on 16-bit displayes - needed for embded [w]

- further and proper extension of upp for python. [w]

- codereview of Firebird SQL adapter and moving it to uppsrc... [w]
- on demand loading of some packages (U++ installation grew from 10MB to 30MB because it contains the whole bazaar). [w]

- choose between ScatterCtrl and PlotLib, adapt for uppsrc. [w]

- ipv6 support [d]

- finish video player, move to uppsrc [?]

Now, I would like to do 4 tasks to plan the next release (which is the one coming after 2011-10 release, say february-april 2012):

- vote what features are the most nice to have in U++
- estimate how hard would be to implement of each feature
- find volunteers to implement them (surely, I am ready to implement a couple of them...)
- based on above info, plan which features will get implemented for the future release

Now I have to admit I am now influenced by SCRUM metodology I became used to lately in my work; it is sort of stupid in the beginning, but it is productive in the long term... So I am trying to adapt what I think might work well for U++ here...

Write your opinions here, or perhaps we should have some nice IRC conference once again....

Mirek

Subject: Re: Development ToDo finalized
Posted by koldo on Sat, 08 Oct 2011 10:07:44 GMT
View Forum Message <> Reply to Message

Hello Mirek

This is a good initiative. The subjects are important and require good skills and many hours.

As U++ only deserves the best , I propose you a focus good for U++ but fair with the community:

- You propose the development directions

- People (for example, me) choose one or more subjects and propose:
-- a conceptual design with classes structure and visual samples if needed
-- a plan, for example, a proof of concept in 1 month, a beta in 3 months and a release version in 4 months.

- All people put their proposals during a period of time

- When that period is over, the best proposals are chosen

- If the programmer complies with the planning and goals, everything is ok

- If the programmer is too much delayed (for example a 50% of delay), the subject is again opened for anybody to follow with it.

I think an approach like this is fair with the people and is also good for U++.

Subject: Re: Development ToDo finalized
Posted by forlano on Sat, 08 Oct 2011 14:08:19 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 05 October 2011 22:58
- vote what features are the most nice to have in U++
- estimate how hard would be to implement of each feature
- find volunteers to implement them (surely, I am ready to implement a couple of them...)
- based on above info, plan which features will get implemented for the future release
...
Write your opinions here, or perhaps we should have some nice IRC conference once again....

Mirek

Hello Mirek,

because you ask to vote and I have not read about any particular programming skill the elector should have, here I am with my opinion. I vote for

1) Web development framework (something like Jango in U++ should be piece of cake..)
2) Web development support in theide (html/js highlighting, some way how to export these to output directory

I am not aware how difficult these tasks will be as it is beyond my possibility. I can offer only support for testing. I am ready to rewrite what I have done in PHP + MySql (two application for my School but very useful) in U++.
I suppose you want to mantain the easy/U++ way to create GUI with sql capability and deploy on a server. Today the market offers what they call "easy" and productive (sic!) framework. But when you want to obtain a stupid table with a minimum of CRUD operations then one discover their terrible limit and lack of flexibility. The solution is then to use in their "easy" framework thirdth part component that performs quite natural operation simply unforeseen.
From other side I have seen some really good framework but they are not free or with price not affordable by not professional programmer. I believe that in this scenario U++ would be very innovative and the web version could become very popular.

Luigi

Subject: Re: Development ToDo finalized
Posted by mirek on Tue, 18 Oct 2011 18:35:53 GMT
View Forum Message <> Reply to Message

I have updated the list with "complexity estimates", where

[h] - takes hours to develop
[d] - days
[w] - weeks
[m] - months

[q] - quartals

Mirek

Subject: Re: Development ToDo finalized
Posted by unodgs on Tue, 18 Oct 2011 20:04:45 GMT
View Forum Message <> Reply to Message

I vote for:
1. Nacl
2. Web framerwork
3. Javascript v8
4. 64bit debugger

Frankly I would like to see all of your list implemented

Subject: Re: Development ToDo finalized
Posted by 281264 on Sun, 23 Oct 2011 18:44:21 GMT
View Forum Message <> Reply to Message

I congratulate you for this initiative.

I vote for:

1.- theide: Windows 64-bit debugging support. Remark: I think the debugger still contains some minor bugs;

2.- theide look&feel improvements (docking, new icons...);

3.- configurable menu bar: a great improvement, certainly;

4.- Multi threading for Painter: would it be possible to implement something similar fot GLCltr?.

Best wishes.

Thanks,

remark: somehow, my last post didn't work, that's why is empty.

Javier

Subject: Re: Development ToDo finalized
Posted by harmac on Fri, 18 Nov 2011 03:33:32 GMT
View Forum Message <> Reply to Message

Not many votes so far. How come?

From what you've listed I'd like to see:

1. U++ embedded [m]
Rather than creating the n-th web development framework, exploring a more neglected field might be more interesting. Especially interesting would be to run applications without an underlying operating system in a frame buffer on some development board. In another thread, the Raspberry Pi has been identified as potentially promising.

2. some production quality complex examples [w]
If I understand the description correctly, this will provide well-commented and explained example U++ applications and will as such serve documentation. This would be very welcome. Additionally, further in the documentation department, there should really be a PDF manual on the web site or alternatively a printer friendly single-page HTML manual to get people started. Documentation is actually as important as functionality and seems to be somewhat poor.

3. further and proper extension of U++ for python [w]
What exactly does this mean? Extending U++ for Python development? If so, that should be done in a general way to allow U++ to support further languages. Support for Python development in U++ would also address the wish to do web development in U++, since there already are some Python web frameworks. I also don't see of what use C++ would be in web development. There's also haXe that can be used to generate C++ code, and with a general way for U++ to support development in different languages, it could be used in U++ for web development.

4. IPv6 support [d]
Can't be wrong, no?

Subject: Re: Development ToDo finalized
Posted by mirek on Fri, 18 Nov 2011 13:42:53 GMT
View Forum Message <> Reply to Message

harmac wrote on Thu, 17 November 2011 22:33Not many votes so far. How come?

From what you've listed I'd like to see:

1. U++ embedded [m]
Rather than creating the n-th web development framework, exploring a more neglected field might be more interesting. Especially interesting would be to run applications without an underlying operating system in a frame buffer on some development board.

Actually, we are 95% there already...

Quote:
In another thread, the Raspberry Pi has been identified as potentially promising.

I guess Pi is too powerful... you can run normal X11 desktop on it.

Quote:
2. some production quality complex examples [w]
If I understand the description correctly, this will provide well-commented and explained example U++ applications and will as such serve documentation. This would be very welcome.

We have now 117 reference examples and 35 more complex demo examples.... More comments/docs are always fonr and more examples are always coming, but I would not say we are particulary lacking in examples.

Quote:
Additionally, further in the documentation department, there should really be a PDF manual on the web site or alternatively a printer friendly single-page HTML manual to get people started. Documentation is actually as important as functionality and seems to be somewhat poor.

Even PDF we had at some moment, right now we have some problems as doumentation book got way too long to generate in single pass. Will be fixed..

Quote:
3. further and proper extension of U++ for python [w]
What exactly does this mean?

Python as plugin, so that U++ app can run Python scripts...

Quote:
Python web frameworks. I also don't see of what use C++ would be in web development.

Well, we already have most of technology ready. We have much stronger DB iface that would ever be possible in Python.

The endpoint for U++/web is that we are able to create more compact, more reliable and more easily deployable web apps, running about 20x faster than with Python. I believe it is worth the try...

Subject: Re: Development ToDo finalized

Posted by forlano on Fri, 18 Nov 2011 14:06:04 GMT
View Forum Message <> Reply to Message

mirek wrote on Fri, 18 November 2011 14:42harmac wrote on Thu, 17 November 2011 22:33
Python web frameworks. I also don't see of what use C++ would be in web development.

Well, we already have most of technology ready. We have much stronger DB iface that would ever be possible in Python.

The endpoint for U++/web is that we are able to create more compact, more reliable and more easily deployable web apps, running about 20x faster than with Python. I believe it is worth the try...

BANZAI!

I have stopped to program in PHP and waiting the day I can deploy the app in U++.

Luigi

Subject: Re: Development ToDo finalized
Posted by kohait00 on Wed, 23 Nov 2011 10:09:04 GMT
View Forum Message <> Reply to Message

Quote:
3. further and proper extension of U++ for python [w]
What exactly does this mean?

actually, python scripts are already runable and useable from upp. see bazaar/PyConsoleCtrlTest, this is straight forward.

the proper interaction / bridge Python <-> upp

* calling upp from python scripts,
* invoking python from upp code
* exporting the upp infrastructure, namely Value, Callbacks, CtrlCore, CtrlLib to a cetrain degree

is still a starter and needs some continuation. for web dev. POV, im not into it

Subject: Re: Development ToDo finalized
Posted by mirek on Fri, 25 Nov 2011 17:21:13 GMT
View Forum Message <> Reply to Message

OK, in the next 6 months or so I am going to create U++ based web framework.

I would also like to see:

- theide: Windows 64-bit debugging support [w]
- theide look&feel improvements (docking, new icons...) [m]
- Google NaCl support in theide(IMO, this one would be big for U++) [m]
- ipv6 support [d]

but let us consider them 'optional', volunteers needed....

With web I am pretty sure now we can deliver...

Subject: Re: Development ToDo finalized
Posted by lectus on Mon, 12 Dec 2011 15:22:04 GMT
View Forum Message <> Reply to Message

My votes:

- Web development framework (something like Jango in U++ should be piece of cake..) [q]
- MacOS X support in theide [w]
- Android support in theide [m]
- Android U++ backend [q]

Of course, all of them are great! But I had to choose the most urgent.
Here's why I chose those:
The more platforms U++ runs on the bigger the possibilities of attracting new users, attracting donations and contributions. This will set up U++ for growing faster in the future.

Subject: Re: Development ToDo finalized
Posted by mirek on Mon, 12 Dec 2011 18:58:45 GMT
View Forum Message <> Reply to Message

lectus wrote on Mon, 12 December 2011 10:22My votes:

Well, store them for the next round of voting

Mirek
