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
