Subject: Re: Rainbow, first iteration
Posted by harmac on Wed, 15 Jun 2011 14:21:35 GMT
View Forum Message <> Reply to Message

mirek wrote on Mon, 13 June 2011 15:03I am now plannig for actually developing generic framebuffer backend to tune this thing.

Lacking knowledge about U++ implementation and style and graphics programming in general so far, I don't know if the following resources are helpful in the endeavour, as at present I'm too much of a C/C++ newbie, but still:

There is portable framebuffer library pxCore that seems to have been ported already to Windows, PocketPC (ARM4), Linux(X11), OSX. Maybe you can recycle some code of it for U++ if it fits the need or maybe it suits as a generic backend already.

It may not be directly framebuffer related, but when I read about the Fog-Framework, its Fog-UI component seems to use an abstract interface for GUI descriptions that is later mapped to different native targets. It sounds somewhat like what Rainbow seems to be trying. I haven't looked at the implementation but conceptually it sounds cleaner than the current U++ approach, which you name ugly yourself. Coming from different languages, from what I understand about the C/C++ preprocessor, it is an ugly kludge in the first place, and maybe it is worth to consider not using it at all.

If not for Rainbow, looking at Fog might be worthy in its own right, as some of its explicit design notes (like for instance not using STL) seem to be in accord with U++ philosophy, so that it might possibly be interesting to recycle code from there or maybe even join some efforts, as some purposes of both projects seem to overlap.

I don't know in what state Fog currently is, though, as the author seems to be concerned with another project (AsmJit), part of which he but seems to plan to integrate into Fog in the form of BlitJit, and in a post on a blog about the Fog-Framework (which also has a post with a number of vector geometry resources, which are probably a bit over my mind at the moment but which more involved people or those interested in learning might find interesting) he assures that the project is not dead. One of the goals at least seems to be high performance with maximum compatibility and adaptation to the target.

Speaking of which, when somebody recently posted about jslinux, I also read in an article about Fabrice Bellard that he had implemented with TinyGL a small subset of OpenGL that can be used as a fallback for systems that normally don't have hardwaresupport for it. As it is very old and probably incomplete, I don't know how close it is to OpenGL ES, but maybe it might be interesting for implementers of U++ OpenGL backends to take a look at such a limited subset whose GL instruction implementation for some arbitrary target platform can be done with reasonable effort and use that subset as a portable target for translations from U++ GUI descriptions.

Finally, there is Agar, which I hadn't heard about before. It may not follow any U++ philosophy but also has different raw targets and might therefore be interesting for somebody interested in the topic to see how different folks are doing their implementations.

pxCore, Fog-Framework and Agar are BSD-licensed, TinyGL zlib-like. As also some external links on their respective websites might be interesting for one or another and some of the projects themselves might be interesting for the U++ folks, I thought that pointing to these projects might be worthwhile. Note that I currently lack the knowledge to decide whether or not they actually are useful.
