Posted by mirek on Mon, 13 Jun 2011 13:03:25 GMT
View Forum Message <> Reply to Message

I took time, especially considering it is mostly preprocessor compile time hack, but I believe that I have reached important milestone in Rainbow development, I believe it should be now possible to develop custom GUI for U++ without the need dir directly changing CtrlCore/CtrlLib (like MacOS or Android or OpenGL).

How is it supposed to work, at least, starting point:

The CtrlCore.h now begins with:

#include <guiplatform.h>

#ifndef GUIPLATFORM_INCLUDE

#ifdef PLATFORM_WIN32
#define GUIPLATFORM_INCLUDE "Win32Gui.h"
#endif

#ifdef PLATFORM_X11
#define GUIPLATFORM_INCLUDE "X11Gui.h"
#endif

#endif

guiplatform.h is in uppsrc root (not in package) and it is empty, so for normal operations "defaults" kick in.

If you are about to develop custom GUI backend, create a new nest, place guiplatform.h into it and using #define GUIPLATFORM_INCLUDE 'redirect' it to your own GUI specification header.

As for development itself, I believe that two 'default' backends give a good hint... I am using quite ugly combination of macros, includes and platform defined functions/methods; for now it seems to be the most cost efficient way. We will say how that works in practice...

Important notice: Whereas in the past we were using PLATFORM_WIN32 and PLATFORM_X11 in CtrlLib and other GUI code for #ifdefs to tell apart the GUI backend, this should be now replaced by 'GUI_WIN' and 'GUI_X11' (those are defined in GUIPLATFORM_INCLUDE), because it is no longer true that Windows backend has to used on Windows platform...

OK, this should work as first introduction. I am now plannig for actually developing generic framebuffer backend to tune this thing.

Perhaps Daniel could now also try to 'port' OpenGL U++ to rainbow too.

Mirek