
Subject: Re: An OpenGL ctrl for Linux / X11
Posted by [mirek](#) on Tue, 06 Jun 2006 20:46:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

cyrion wrote on Tue, 06 June 2006 13:48luzr wrote on Tue, 06 June 2006 17:59
BTW, IMO "UseGLXVisual" could be replaces by existing "BackPaint(EXCLUDEPAINT)".

I have just tested BackPaint() but it doesn't work because DoPaint() is still called with a bad X visual

I get things like that :

X Error: BadDrawable (invalid Pixmap or Window parameter), request: X_CreateGC, resource id: 136899272 = 828EAC8

X Error: BadGC (invalid GC parameter), request: X_SetClipRectangles, resource id: 48234558 = 2E0003E

X Error: BadGC (invalid GC parameter), request: X_SetClipRectangles, resource id: 48234558 = 2E0003E

X Error: BadGC (invalid GC parameter), request: X_SetClipRectangles, resource id: 48234558 = 2E0003E

X Error: BadGC (invalid GC parameter), request: X_FreeGC, resource id: 48234558 = 2E0003E

IMHO, instead of using a variable like my dirty UseGLXVisual it would probably be best to take into account the fact that some controls can require special visuals that does not require a Draw object to be created. But I don't know how ! (a kind of BackPaint(3DPAINT) ?)

In the end, I think that Ctrl::Paint() should still be called, but with a 'null/disabled' Draw object in parameter...

Well, I have not studied your version in detail and in fact, I never really understood concept of X11 visuals, however...

Normal U++ widgets do not have corresponding objects (handles) in neither X11 or Win32 - U++ normally uses just top-level windows. That in turn means that there is only single Draw for each top-level Ctrl.

That of course is sometimes trouble - e.g. for OpenGL which requires separate GUI object. In order to solve that, there is special DHCtrl class, so far implemented in Win32 only, which has corresponing object (HWND) in Win32 and handles WM_PAINT completely separately. BackPaint(EXCLUDEPAINT) just clips the content of DHCtrl out of normal painting procedure (which is always called for entire top-level Ctrl). I believe that Linux should, if possible, follow the path. GLCtrl is derived from DHCtrl.

Mirek
