

---

Subject: UppGL

Posted by [unodgs](#) on Tue, 17 May 2011 09:42:07 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi!

Yesterday I created a new branch called uppgl that is ultimate++ that uses opengl as a main painting engine. This is a part of my current project in which I needed animated interface with some cool effects. First I wanted to create completely separate ui engine but over the years I got used to u++ so much I didn't want to resign from it. Besides positioning system in upp is very powerful and easy to use at the same time. So I added another system drawer.

You can try it by creating assembly that points to "branches\uppgl;branches\uppgl\uppsrc" and running GuiTest. If everything will go fine you should see gui controls rendered by opengl. This demo can be for now run only in windows, but my plan is to make it working under linux as well, but later

Right now I'm using OpenGL in version 1.0 in immediate mode but to my surprise it is quite efficient if it comes to ui rendering. My goal is to switch to OpenGL 2.0 ES so it could work almost everywhere.

Not everything is working:

- popup windows are not automatically closed (I had problems with calling Deactivate method)
- there is only one font implemented for now - tahoma 14 (normal and bold)
- all themes work except host one. Unfortunately it is impossible to cache painting elements that are dynamically rendered by a system
- child windows are not implemented yet

Anyway this is only a start. I wanted to share in hope that someone else than me could make it better and would help to develop missing parts.

Of course there there are many places to improve:

- fonts could be dynamically generated (there is wglUseFontBitmaps but it works only under windows)
- all ui textures could be gathered into one texture. For now many small textures are generated.
- intermediate mode should be replaced with vbo's and shaders

In the project you can see plugins/glew package which is automatically initialized on application startup so you can use every extension possible.

During implementation I came to the conclusion that CtrlCore should be reorganized. It's quite hard to add another system draw kind of class. Right now depending on compiler flag SystemDraw is a win32 draw or x11 draw. It would be better if there was SystemDraw an interface (like Draw is now) and X11SystemDraw, Win32SystemDraw so I could easily add OpenGLSystemDraw. That could also let to have Win32SystemDraw and OpenGLSystemDraw at the same time and use them interchangeably. OpenGL for gui rendering, Win32 one for pdf generating or printing or anything else that OpenGL draw is not good for. Right now I have BaseDraw that is old system draw and SystemDraw that uses OpenGL. Akward but had no time to implement it better and I wanted to hear your (especialy Mirek's) opinion on that first.

That's all for now. Please tell what you think about it.

PS: Of course it would be better to have hardware accelerated Painter and to render everything through it, but current solution seemed easier to me.

---