Subject: GLCtrl questions
Posted by cbpporter on Mon, 13 Sep 2010 09:52:27 GMT
View Forum Message <> Reply to Message

I have a few questions about GlCtrl. I don't know too much about OpenGl because I'm more on the side of DirectX. But for some pretty basic stuff it should suffice and setting up DirectX is not easy.

1. As far as I can tell, except for the control itself standard OpenGl API is used. This means that one can do anything with GLCtrl that one can do with straight forwards OpenGl? Also, these function seem to not take any destination. Can I have more OpenGl views at the same time?

2. Can you have other controls inside the OpenGl control? And does normal (HDC based for Windows) draw operation work on that control? Can I render a 3D scene and then draw a line on top of it without OpenGL?

3. How can I make my application full screen and supply a resolution? Also, is it possible for OpenGL in fullscreen mode to use full/real hardware accelerated page swapping (not just hardware accelerated bliting)?

Subject: Re: GLCtrl questions
Posted by 281264 on Mon, 13 Sep 2010 18:01:20 GMT
View Forum Message <> Reply to Message

Hi cbpporter,

Not being an OpenGL expert (I am currently using a limited portion of it and not for games, btw) these are my answers:

1.- GLCTrl is a way to create a substratum for the usage of OpenGL in Windows: i.e. specification of pixelformatdescriptor, creation of an OpenGL rendering context (by using wglCreateContext(), and so on. These features are easily recognizable in the code. I suggest you to have a look to the

http://fly.cc.fer.hr/~unreal/theredbook/). Also have a lookt at the example included in the reference assembly of TheIde.

Yes, you can create anything in OpenGL by using GLCTrl. For example, the function in GLCTrl.cpp:void GLCtrl::StdView()

```
{
 glShadeModel(GL_SMOOTH);
 glClearColor(0.0f, 0.0f, 0.0f, 0.5f);
 glClearDepth(1.0f);
 glEnable(GL_DEPTH_TEST);
 glDepthFunc(GL_LEQUAL);
 glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST);

 Size sz = GetSize();
 glViewport(0, 0, (GLsizei)sz.cx, (GLsizei)sz.cy);
 glMatrixMode(GL_PROJECTION);
 glLoadIdentity();
 gluPerspective(45.0f, (GLfloat)(sz.cx)/(GLfloat)(sz.cy), 1.0f, 100.0f);
 glMatrixMode(GL_MODELVIEW);
 glLoadIdentity();
}
```

sets up the basic features you need: viewport, type of perspective projection (gluPerspective in this case, although I am currently using glOrtho), etc..

Yes, you can have more than one OpenGL views at the same time. Just generate a second

2.- I have never tried that for I never needed it.

3.- As long as you are able to create a window without borders, capable to occupy the whole screen (which is not related with OpenGL), then OpenGL will occupy the full screen.

BTW I am interested in working with a TopWindow in U++ capable to occupy the full screen, i.e. to use the screen space used by the lower Windows toolbar; any hint how to do it?

Subject: Re: GLCtrl questions
Posted by cbpporter on Mon, 13 Sep 2010 18:53:46 GMT
View Forum Message <> Reply to Message

TopWindow::FullScreen is used for that.

Anyway, my dislike for OpenGL has proven on good base again and it managed not to work alright on any hardware I have. For cutting edge 3D DirectX is more advanced and more supported (I only need a few polys), and on cheap integrated cards I always get graphical errors. And in a period at least, ATI drivers did not have OpenGL hardware acceleration at all. I had hello world OpenGL apps running at 6 FPS. From my personal experience I can say that the chance of OpenGL working is very low when compared to DirectX. I tried two computers today and both were unsatisfactory even with basic stuff.

I am investigating Irrlicht right now. It has support for multiple backends so if OpenGL does not work, DirectX does. Software rendering is also fast enough. I managed to use it with U++, but not to limit it it to a single DHCtrl yet. Maybe I can get something that only falls back under OpenGL for Linux.

Subject: Re: GLCtrl questions
Posted by dolik.rce on Mon, 13 Sep 2010 19:01:53 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Mon, 13 September 2010 11:523. How can I make my application full screen and supply a resolution? Also, is it possible for OpenGL in fullscreen mode to use full/real hardware accelerated page swapping (not just hardware accelerated bliting)?

281264 wrote on Mon, 13 September 2010 20:013.- As long as you are able to create a window without borders, capable to occupy the whole screen (which is not related with OpenGL), then OpenGL will occupy the full screen.

BTW I am interested in working with a TopWindow in U++ capable to occupy the full screen, i.e. to use the screen space used by the lower Windows toolbar; any hint how to do it?

To get a full-screen window, there is method TopWindow::FullScreen() (surprisingly). Just make your GlCtrl filling the whole window and you get what you want, in the resolution currently set on the monitor.

To change the resolution, I'm afraid you'd have to go low-level. Not sure if there is some way to do it using OpenGL. I remember switching VGA modes using inline assembler (very long ago, in dos programs written in pascal, but it still works sometimes, at least on win XP). I'm not sure what is the current situation in modern OSes.

Honza

Honza

Subject: Re: GLCtrl questions
Posted by cbpporter on Mon, 13 Sep 2010 19:12:21 GMT

View Forum Message <> Reply to Message

dolik.rce wrote on Mon, 13 September 2010 22:01cbpporter wrote on Mon, 13 September 2010 11:523. How can I make my application full screen and supply a resolution? Also, is it possible for OpenGL in fullscreen mode to use full/real hardware accelerated page swapping (not just hardware accelerated bliting)?

281264 wrote on Mon, 13 September 2010 20:013.- As long as you are able to create a window without borders, capable to occupy the whole screen (which is not related with OpenGL), then OpenGL will occupy the full screen.

BTW I am interested in working with a TopWindow in U++ capable to occupy the full screen, i.e. to use the screen space used by the lower Windows toolbar; any hint how to do it?

To get a full-screen window, there is method TopWindow::FullScreen() (surprisingly). Just make your GlCtrl filling the whole window and you get what you want, in the resolution currently set on the monitor.

To change the resolution, I'm afraid you'd have to go low-level. Not sure if there is some way to do it using OpenGL. I remember switching VGA modes using inline assembler (very long ago, in dos programs written in pascal, but it still works sometimes, at least on win XP). I'm not sure what is the current situation in modern OSes.

Honza

Honza

I was trying to achieve the DirectX kind of fullscreen. You can have a small view, like 300x200 dedicated to DirectX and the rest normal Windows widgets. If you go fullscreen with a given resolution, your 3D widget will remain small and the rest of the widgets will still work.

VGA is dead . Anyway, you can't change the resolution that way for Windows applications. There may be a way with Windows API.

Subject: Re: GLCtrl questions
Posted by koldo on Mon, 13 Sep 2010 20:30:18 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Mon, 13 September 2010 20:53TopWindow::FullScreen is used for that.

Anyway, my dislike for OpenGL has proven on good base again and it managed not to work alright on any hardware I have. For cutting edge 3D DirectX is more advanced and more supported (I only need a few polys), and on cheap integrated cards I always get graphical errors. And in a period at least, ATI drivers did not have OpenGL hardware acceleration at all. I had hello world OpenGL apps running at 6 FPS. From my personal experience I can say that the chance of OpenGL working is very low when compared to DirectX. I tried two computers today and both were unsatisfactory even with basic stuff.

I am investigating Irrlicht right now. It has support for multiple backends so if OpenGL does not work, DirectX does. Software rendering is also fast enough. I managed to use it with U++, but not to limit it it to a single DHCtrl yet. Maybe I can get something that only falls back under OpenGL for Linux.
Hello cbpporter

Good for testing Irrlicht.

Have you also considered other libraries like Ogre3D or other more centered in engineering?

Subject: Re: GLCtrl questions
Posted by mrjt on Tue, 14 Sep 2010 14:35:06 GMT
View Forum Message <> Reply to Message

I've attached a ctrl that can display an Ogre3D rendering context. I can't guarantee it will work with latest version since I haven't used it in over a year but it should at least point you in the right direction. It can toggle between DirectX and OpenGL rendering.

If you want full screen rendering you're best off creating a seperate non-Upp context.

I'm reasonably sure you can place ctrls over a GL/Ogre Ctrl without problems. I never tried GDI but it should work in windowed mode.

I'm not really sure why you've had so many problems with OpenGL support. I released a program last year in pure OpenGL (a 3D visualiser for a game called Dwarf Fortress) that was used by a lot of people on all sorts of shoddy hardware. That used vertex and pixel shaders, shadow maps and some other tricky stuff and ran on everything from a GeForce2 MX 400 (I have that in my PC right now) to the most recent cards, includingg ATi ones.

File Attachments

1) OgreCtrl.zip, downloaded 395 times

Subject: Re: GLCtrl questions
Posted by cbpporter on Tue, 14 Sep 2010 14:53:14 GMT
View Forum Message <> Reply to Message

mrjt wrote on Tue, 14 September 2010 17:35(a 3D visualiser for a game called Dwarf Fortress)
No way!!!!

Link please!

Subject: Re: GLCtrl questions
Posted by cbpporter on Thu, 16 Sep 2010 12:28:13 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Tue, 14 September 2010 17:53mrjt wrote on Tue, 14 September 2010 17:35(a 3D visualiser for a game called Dwarf Fortress)
No way!!!!

Link please!

Is it 3D Dwarf?

I'm interested because I'm working on my own Dwarf Fortress inspired project. Are you involved with DF community/development?

Subject: Re: GLCtrl questions
Posted by mrjt on Thu, 16 Sep 2010 16:23:06 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Thu, 16 September 2010 13:28cbpporter wrote on Tue, 14 September 2010 17:53mrjt wrote on Tue, 14 September 2010 17:35(a 3D visualiser for a game called Dwarf Fortress)
No way!!!!

Link please!

Is it 3D Dwarf?

I'm interested because I'm working on my own Dwarf Fortress inspired project. Are you involved with DF community/development?
Mine was much better than that

I'm the author of VisualFortress. SourceForge page.

The last version it officially worked with was DF 0.28.181.40d and I've stopped developing it. I'm not really playing DF any more, though I will again someday, and have moved onto other projects. Development of MediaPortal 2 is my current time-sink.

I just went back to the forum for the first time in a long time and my original thread is now 89 pages long and still on the second page! There are some nice screenshots on this page and there are hundreds more in the thread.

James

Subject: Re: GLCtrl questions
Posted by mrjt on Thu, 16 Sep 2010 16:35:39 GMT
View Forum Message <> Reply to Message

What's your project by the way?

Subject: Re: GLCtrl questions
Posted by cbpporter on Wed, 22 Sep 2010 11:53:50 GMT
View Forum Message <> Reply to Message

It is a DF inspired game. A hobby project. I'm hopping for Dwarf3D level graphics out of the box somewhere in the future.

My experimentation with Irrlicht has been very successful. I'm still experimenting with having a small Irrlicht window and U++ in rest. Maybe the right solution is to have an Irrlicht singleton and a custom version of SystemDraw that uses that object for all the drawings. Like a frame-buffer port of U++, but to a 3D window. I don't know yet.

My framerate is good, but I found a computer where I have terrible framerate. There is a lot more fuss when doing hardware acceleration than with simple software rendering .

Subject: Re: GLCtrl questions
Posted by cbpporter on Fri, 24 Sep 2010 10:22:25 GMT
View Forum Message <> Reply to Message

Well things are no longer smooth. I am having troubles with switching between U++ event loop and Irrlicht event loop. There are strange bugs like a single label not updating and key event getting lost.

And things become even nastier if I start out with a "normal" GUI application and then switch over to a window and setup 3d acceleration.

Subject: Re: GLCtrl questions
Posted by 281264 on Mon, 27 Sep 2010 10:27:12 GMT
View Forum Message <> Reply to Message

In this context, how can GLCTrl be encapsulated in its own thread? What I am looking for is a way

Is this achievable?.

Cheers,

Javier

Subject: Re: GLCtrl questions
Posted by cbpporter on Wed, 29 Sep 2010 11:02:43 GMT
View Forum Message <> Reply to Message

I don't know what to say about multithreading and GL. I am having troubles with mixing U++ windows and the rendering context. I'll continue to investigate but probably I am going to have to drop CtrlCore and CtrlLib.

And another thing I have determined experimentally after a lot of troubles with a few select computers: use textures that have dimensions divisible by 64. Even my 2D API had visual artifacts because after activating hardware acceleration the textures would get enlarged to whatever size the video card uses most comfortably. I only saw this happen on low quality video cards. Just to be safe, I am using now 512x512 size textures.

Subject: Re: GLCtrl questions
Posted by cbpporter on Fri, 01 Oct 2010 15:00:58 GMT

View Forum Message <> Reply to Message

I did something silly: I wrote a very rudimentary bridge/wrapper system that maps U++ like controls to Irrlicht controls. It only works for 3 widgets, but changing Button to DeviceHw::Button and recompiling does the job. Maybe I can get a define going and switch between builds U++ controls to Irrlicht controls. Of course, only a small subset of methods has been implemented and the entire thing is quite hacky and might blow up anytime.

Subject: Re: GLCtrl questions
Posted by mrjt on Mon, 04 Oct 2010 10:00:09 GMT

View Forum Message <> Reply to Message

It's strange you're having problems, I was able to get multiple GLCtrls working in a Upp app quite easily. The main gotcha is to know that there are some cases (such a modal windows / prompts) that change the rendering context so you must always do wglGetContext / wglSetContext before calling GL functions.

I'm not sure what befit you would get from running the rendering in another thread. If your rendering is that slow then you will never be able to assure a responsive application. Just keep the GUI and Rendering in the main thread and move everything else (where possible) into worker threads.