Subject: An OpenGL ctrl for Linux / X11
Posted by cyrion on Fri, 02 Jun 2006 16:40:19 GMT
View Forum Message <> Reply to Message

Hi,
I have just finished (the first version of) an opengl ctrl for X11.
It works well but I had to make modifications to U++ core, and I would like to know if there is a best way to proceed.

The modifications I have made are:

1/ in CtrlCore.h, class Ctrl :

* I put the static ArrayMap<> Xwindow() in protected section.
I had to do so because the Create() method does not allow to use another X visual than the top window one. So I had to put my new X (sub)window "by hand" into the static ArrayMap of windows.

* I added a 'bool UseGLXVisual' (in protected section), defaulting to false except in my new Ctrl.

2/ in X11Wnd.h

* in Ctrl::DoPaint(), I 'return' without creating a GC if UseGLXVisual is true (just before XCreateGC).
I had to do so to prevent the Ctrl::Draw() method to be called with a GC context that is not compatible with my X window.

Finally, all seems to work fine : resizing, moving, hide/show, multiple gl ctrls, etc...

I hope that the source code will be more comprehensible than my english.

Btw, _many_many_ thanks for this wonderful library !

Damien
(Linux 2.6, debian, Xorg, U++ 605)

File Attachments
1) openglctrl.tgz, downloaded 1867 times

Subject: [EDIT] An OpenGL ctrl for Linux / X11
Posted by cyrion on Sat, 03 Jun 2006 10:36:31 GMT
View Forum Message <> Reply to Message

Hi,
Below are the modifications I have done to the CtrlCore package.

In the attached file you will find a working example of the gl ctrl (a spinning cube). I also added the binary (for Linux i386).

Modifications to CtrlCore.h

```
(...)
 static dword KEYtoK(dword);

protected:
 bool UseGLXVisual; // here !

private:
 Ctrl(Ctrl&);
 void operator=(Ctrl&);

(...)
```

and here:

```
(...)
#ifdef PLATFORM_X11
protected:
 struct XWindow {
  Ptr<Ctrl>    ctrl;
  bool         exposed;
  Vector<Rect> invalid;
  Ptr<Ctrl>    owner;
  Ptr<Ctrl>    last_active;
  XIC          xic;
 };
 static ArrayMap<Window, XWindow>& Xwindow(); // here !
private:
 static Ptr<Ctrl> WndCaretCtrl;
(...)
```

Modifications to X11Wnd.cpp

```
(...)
void Ctrl::DoPaint(const Vector<Rect>& invalid)
{
(...)

  if( UseGLXVisual ) // here !
```

```
  return;

 GC gc = XCreateGC(Xdisplay, (Drawable)top->window, 0, 0);
#ifdef PLATFORM_XFT
```

(...)

And finally to Ctrl.cpp

```
(...)
Ctrl::Ctrl() {
 LLOG("Ctrl::Ctrl");
(...)
 unicode = false;
 UseGLXVisual = false; // here !
}
(...)
```

Damien.

File Attachments

1) example.tgz, downloaded 2074 times

Subject: Re: [EDIT] An OpenGL ctrl for Linux / X11
Posted by mirek on Mon, 05 Jun 2006 08:57:29 GMT
View Forum Message <> Reply to Message

Looks excelent, however, there is already existing GlCtrl which so far lacks POSIX implementation. (check reference/OpenGL).

Would not it be wiser to have single interface for Linux / Win32?

Mirek

Subject: Re: An OpenGL ctrl for Linux / X11
Posted by cyrion on Tue, 06 Jun 2006 14:02:47 GMT
View Forum Message <> Reply to Message

Hi,
I am aware of the GL Ctrl working on Windows, that's exactly why I made the posix version

What I wanted to know was if there was a best way to proceed, i.e. while avoiding modifying the core of Upp.
If all seems ok, I could make it portable with a couple of #ifdef in the Upp GLCtr, and if you are interested, you could use it for the next release.
This would be my first contribution to Upp !

Subject: Re: An OpenGL ctrl for Linux / X11
Posted by mirek on Tue, 06 Jun 2006 15:59:24 GMT
View Forum Message <> Reply to Message

cyrion wrote on Tue, 06 June 2006 10:02Hi,
I am aware of the GL Ctrl working on Windows, that's exactly why I made the posix version
What I wanted to know was if there was a best way to proceed, i.e. while avoiding modifying the core of Upp.
If all seems ok, I could make it portable with a couple of #ifdef in the Upp GLCtr, and if you are interested, you could use it for the next release.
This would be my first contribution to Upp !

Perfect!

That would be an important milestone, as by that act, we would finally reach Linux/Win32 parity (because other things like printing and clipboard are being resolved now as well).

BTW, IMO "UseGLXVisual" could be replaces by existing "BackPaint(EXCLUDEPAINT)".

Mirek

Subject: Re: An OpenGL ctrl for Linux / X11
Posted by cyrion on Tue, 06 Jun 2006 17:48:23 GMT
View Forum Message <> Reply to Message

luzr 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...

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

Subject: Re: An OpenGL ctrl for Linux / X11
Posted by cyrion on Thu, 08 Jun 2006 05:00:12 GMT
View Forum Message <> Reply to Message

Hi,

I checked on windows to see how it works. I'm not sure, but I think that BackPaint(EXCLUDEPAINT) does the job for Windows, but not exactly in the same manner for Linux because the GC is always created.
While using BackPaint(EXCLUDEPAINT), my problem disappear only if I don't call the inherited EventProc() of Ctrl in my OpenGLCtrl::EventProc, but in this case Frames are never painted !
I run out of ideas.

In addition, a desirable feature would be that Ctrl objects (at least GL ctrls) receive a 'POSITION' event. Currently I have to watch for a modification of the widget position each time Paint is called, which slows down rendering.
BTW, the windows version of the gl ctrl does not handle displacements of the ctrl. For instance, try to use win.Add(gl.BottomPos(0, 120).RightPos(0, 120));

In the attached file I put a small app using the new version of the Ctrl. It now handles multiples GL widgets on the same app automatically.

Damien.

File Attachments

1) glctrl_app.tar.gz, downloaded 1885 times

Subject: Re: An OpenGL ctrl for Linux / X11
Posted by mirek on Fri, 09 Jun 2006 12:39:27 GMT
View Forum Message <> Reply to Message

cyrion wrote on Thu, 08 June 2006 01:00Hi,

I checked on windows to see how it works. I'm not sure, but I think that BackPaint(EXCLUDEPAINT) does the job for Windows, but not exactly in the same manner for Linux because the GC is always created.

Of course it is - it is used to paint sibling Ctrls...

Quote:
In addition, a desirable feature would be that Ctrl objects (at least GL ctrls) receive a 'POSITION' event.

There is "State" virtual that should get called when position changes - in that case, the parameter is equal to "LAYOUTPOS".

Note that this is overloaded in DHCtrl and should do position syncs...

Mirek

Subject: Re: An OpenGL ctrl for Linux / X11
Posted by cyrion on Fri, 09 Jun 2006 16:01:19 GMT
View Forum Message <> Reply to Message

Mirek, thanks for your quick answers and your patience.

After many hours checking U++ source code as well as my source code, I think there are some bugs into the core:

pb#1: my widget never receive LAYOUTPOS when the position change. However LAYOUTPOS is received when it is resized.

pb#2: when I put my gl ctrl into a TabCtrl, itself contained into another TabCtrl, my function RectInTopWindow
```
Rect OpenGLCtrl::RectInTopWindow() const
{
 return GetScreenView() - GetTopCtrl()->GetScreenRect().TopLeft();
}
```
 does not return a good value. I suspect that GetTopCtrl() does not return the good top ctrl.

pb#3: My new x subwindow with a special visual generates X errors when Paint() is called, except if I use my IsGLXVisual variable to prevent DoPaint to create a graphics context.
If I use BackPaint(EXCLUDEPAINT), I can get rid of X errors if I don't call the inherited Ctrl::EventProc(w, event), void OpenGLCtrl::EventProc(XWindow& w, XEvent *event)
```
{
 // Flush 'Expose' events
```

```
 while( XCheckWindowEvent( Xdisplay, SubWindow, ExposureMask, event )) {};

 if( IsMapped )
  OpenGLPaint();

 Ctrl::EventProc(w, event );
```
} but in this case the frames surrounding my Ctrl (like InsetFrame for instance) are not displayed anymore.

pb#4: When the app window is resized, my ctrl receive a lot of SHOW events, that make my X subwindow swapping to hide/show state. This slows down the resizing process in this case (that is, a X sub window, not a standard ctrl).

If you have time for that, please look at my source code. It would be a great help for me to find if I made a mistake or don't understand something. At this point, I have no more ideas to make my GL Ctrl working without modifying the U++ core.

Sincerely,
Damien.

Subject: Re: An OpenGL ctrl for Linux / X11
Posted by h3l1 on Mon, 02 Oct 2006 16:14:40 GMT
View Forum Message <> Reply to Message

Is there any update on an OpenGL Ctrl for Linux?
Or does it already work?

Bye
Heli

Subject: Re: An OpenGL ctrl for Linux / X11
Posted by mirek on Mon, 02 Oct 2006 17:35:41 GMT
View Forum Message <> Reply to Message

h3l1 wrote on Mon, 02 October 2006 12:14Is there any update on an OpenGL Ctrl for Linux?
Or does it already work?

Bye
Heli

Actually, it would be nice to have it soon for 611 release.

Mirek

Subject: Re: An OpenGL ctrl for Linux / X11
Posted by cyrion on Mon, 02 Oct 2006 18:08:35 GMT
View Forum Message <> Reply to Message

Hi,
I will make a portable version of the widget soon.
Until then, you can use my patched version of the CtrlCore.
In the attached .tgz you will find the patchs for upp 605 and upp 609dev3 (with an example of an opengl application).

Quote:cd upp/
tar zxvf glctrl-upp.tgz

Warning ! it will replace the files :

uppsrc/CtrlCore/CtrlCore.h
uppsrc/CtrlCore/Ctrl.cpp
uppsrc/CtrlCore/X11Wnd.cpp

Damien.

File Attachments

1) glctrl-upp.tgz, downloaded 2051 times

Subject: Re: An OpenGL ctrl for Linux / X11
Posted by h3l1 on Fri, 06 Oct 2006 13:47:52 GMT
View Forum Message <> Reply to Message

Hi Damien,

great work, finally I got it to run. So I tried to get antialiasing to the controls and had to add

```
#ifdef GLX_SAMPLE_BUFFERS_ARB
visual << GLX_SAMPLE_BUFFERS_ARB << 1 << GLX_SAMPLES_ARB << 4;
#endif
```

in the OpenGLCtrl::CreateGLXWindow method. It would be nice if this can be added to the final code. So it is possible to create individual scenes with antialiasing enabled.

Finally enabling antialiasing works with the following code:

```
glEnable(GL_MULTISAMPLE_ARB);
glHint(GL_MULTISAMPLE_FILTER_HINT_NV, GL_NICEST);
```

I added it at the InitGL of the CubeGL control and voila the cube was antialiased. This is really cool!
Now I'm trying to create a control with opengl in high quality.

bye
Heli

Subject: Re: An OpenGL ctrl for Linux / X11
Posted by cyrion on Fri, 17 Nov 2006 00:46:08 GMT
View Forum Message <> Reply to Message

@Mirek & Heli: Sorry for the late reply !
@Heli: I followed your suggestion and added arb multisample antialiasing to the parameters of the GLCtrl.
Note: you still have to glEnable(GL_MULTISAMPLE_ARB) "yourself".

So, the attached archive contains :

1/ the patch against the CtrCore lib
uppsrc/CtrlCore/CtrlCore.h
uppsrc/CtrlCore/Ctrl.cpp
uppsrc/CtrlCore/X11Wnd.cpp

2/ the portable GLCtrl (now including Linux/X11)
uppsrc/GLCtrl/GLCtrl.cpp
uppsrc/GLCtrl/GLCtrl.h
uppsrc/GLCtrl/GLCtrl.upp

3/ the reference example slightly modified to demonstrate antialiasing (optional, of course)
reference/OpenGL/main.cpp

Theses files works with upp611-dev2.

Bye,
Damien.

File Attachments

1) glctrl.tgz, downloaded 1790 times

Subject: Re: An OpenGL ctrl for Linux / X11
Posted by h3l1 on Mon, 20 Nov 2006 13:18:53 GMT

Hi cyrion,

thanks for your work, really great news to have a portable glctrl.
I will try it out in december, since I will have 4 weeks vacation and maybe I get the time to do a small opensource projekt with upp.
I have already an idea to use the opengl control

Bye
Heli

Subject: Re: An OpenGL ctrl for Linux / X11
Posted by cyrion on Mon, 20 Nov 2006 20:56:47 GMT

you will see some small modification to the code: I mostly rename the callbacks to match the Windows version of GlCtrl.

Nice holidays !
Bye,
Damien.