
Subject: openGL and texture

Posted by [forlano](#) on Mon, 25 Feb 2008 20:32:45 GMT

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

Hello,

openGL is really a great fun!

The next step is to add a texture to the solid as explained here

<http://nehe.gamedev.net/data/lessons/lesson.asp?lesson=06>

Unfortunately they used some unknown (to me) structure `AUX_RGBImageRec *LoadBMP(char *Filename)` to store image and I cannot find an equivalent U++ method to follow the tutorial.

So please let me ask you if there is around a test case that show how to add a texture using our lovely U++ method.

Many thanks in advance.

Luigi

PS: I would like to get the basis of openGL and then add some physics to simulate the motion of charged particles in an electromagnetic field. What I saw around are only weird bidimensional examples. Adding the third dimension should be more appealing for students.

Subject: Re: openGL and texture

Posted by [masu](#) on Tue, 26 Feb 2008 13:19:18 GMT

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

Theoretically, you can use Image classes provided with U++, because the definition for `AUX_RGBImageRec` (in `glaux.h`) is:

```
typedef struct _AUX_RGBImageRec {
    GLint sizeX, sizeY;
    unsigned char *data;
} AUX_RGBImageRec;
```

So it only contains size and raw data that can also be extracted using class methods from within U++.

What you basically have to do is to load an RGB image using U++ facilities and extract image size and raw data from it.

And of course adapt the example code provided at NeHe.

Matthias

Subject: Re: openGL and texture

Posted by [mrjt](#) on Tue, 26 Feb 2008 14:29:38 GMT

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

Unfortunately it's not quite that easy, Upp Image formats are not standard RGBA. Depending on the platform they could be BGRA (Win32) or ARGB (see Core/Color.h). Unfortunately there is no GL unpacking format that copes with this directly, so it requires an Image copy with some byte swapping.

This code should work for all platforms (only tested on Win32), though you may wish to change some of the GL parameters:

```
#ifdef PLATFORM_WIN32
#define RGBA_FIX(q) Swap(q->r, q->b);
#endif

#ifndef PLATFORM_POSIX
#ifndef CPU_BE
#define RGBA_FIX(q) { Swap(q->a, q->b); Swap(q->b, q->g); Swap(q->r, q->g); }
#else
#define RGBA_FIX(q) Swap(q->r, q->b);
#endif
#endif

Image RGBAFormat(Image img)
{
    ImageBuffer ib(img);
    RGBA *eoi = ~ib + ib.GetLength();
    for (RGBA *q = ~ib; q < eoi; q++)
        RGBA_FIX(q);
    return ib;
}

GLuint GLTexture(Image img)
{
    GLuint texnum = 0;
    Size sz = img.GetSize();
    Image copy = RGBAFormat(img);

    glGenTextures(1, &texnum);
    glBindTexture(GL_TEXTURE_2D, texnum);

    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER,
                    GL_LINEAR_MIPMAP_NEAREST);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE);

    glTexImage2D(GL_TEXTURE_2D, 0, 4, sz.cx, sz.cy, 0, GL_RGBA, GL_UNSIGNED_BYTE,
                ~copy);
    gluBuild2DMipmaps(GL_TEXTURE_2D, 4, sz.cx, sz.cy, GL_RGBA, GL_UNSIGNED_BYTE,
```

```

~copy);
return texnum;
}

Example usage:
void Init()
{
wglMakeCurrent(GLCtrl::GetDC(), GLCtrl::GetHGLRC());

texture = GLTexture(StreamRaster::LoadFileAny("C:\\texture.png"));

wglMakeCurrent(NULL, NULL);
}

virtual void GLPaint()
{
    ....Set up viewports/molelview matrix as necessary
glEnable(GL_TEXTURE_2D);
glBindTexture(GL_TEXTURE_2D, texture);
glPushMatrix();
glTranslatef(0, 0, -6);
glBegin(GL_QUADS);
glColor3f(1.0f, 1.0f, 1.0f);

glTexCoord2f(0.0f, 1.0f);
glVertex3f(0.0f, 0.0f, 1.0f);

glTexCoord2f(1.0f, 1.0f);
glVertex3f(1.0f, 0.0f, 1.0f);

glTexCoord2f(1.0f, 0.0f);
glVertex3f(1.0f, 1.0f, 1.0f);

glTexCoord2f(0.0f, 0.0f);
glVertex3f(0.0f, 1.0f, 1.0f);
glEnd();
glPopMatrix();
}

```

Obviously the Image copy will create a certain amount of overhead for very large/lots of textures, but it shouldn't be too bad unless intending to write a AAA game (in which case you've got other problems).

Subject: Re: openGL and texture
 Posted by [forlano](#) on Tue, 26 Feb 2008 17:29:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

mrjt wrote on Tue, 26 February 2008 15:29Unfortunately it's not quite that easy, Upp Image formats are not standard RGBA. Depending on the platform they could be BGRA (Win32) or ARGB (see Core/Color.h). Unfortunately there is no GL unpacking format that copes with this directly, so it requires an Image copy with some byte swapping.

...

Thank you James,

I tried the above code that simplified the code of the tutorial. But unfortunately it doesn't work I got only a beautiful white rotating cube with no trace of texture. I tried several things without success. I do not know if the problem resides in the swapping procedure or somewhere else. I attach the code in case somebody has time to check it.

Thanks,
Luigi

File Attachments

1) [openGL5.rar](#), downloaded 416 times

Subject: Re: openGL and texture

Posted by [mrjt](#) on Tue, 26 Feb 2008 17:43:56 GMT

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

AGH! The dreaded white cube!

I assume you are using Linux. After modifying it for the Windows version of GLCtrl (why aren't they the same?) it works fine, so I can't really help.

My best guess would be that it's something to do with the Linux GLCtrl though, if the byte swapping wasn't working properly you'd still see expect to see a texture, just with the wrong colors.

Subject: Re: openGL and texture

Posted by [forlano](#) on Tue, 26 Feb 2008 18:09:20 GMT

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

mrjt wrote on Tue, 26 February 2008 18:43AGH! The dreaded white cube!

I assume you are using Linux.

No, I'm using windows with mingw.

Luigi

Subject: Re: openGL and texture

Posted by [mrjt](#) on Tue, 26 Feb 2008 18:44:01 GMT

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

In that case the problem is that in the Win32 version GLInit doesn't get called by GLCtrl for some reason.

You need to add the following function to the GLExample class:

virtual void State(int reason)

```
{  
    GLCtrl::State(reason);  
    if (reason == Ctrl::OPEN) {  
        wglMakeCurrent(GLCtrl::GetDC(), GLCtrl::GetHGLRC());  
        GLInit();  
        wglMakeCurrent(NULL, NULL);  
    }  
}
```

You can remove the wglMake current calls from GLInit. Also double check that the file is being loaded. If GLTexture example is successful then texture should be non-zero.

Subject: Re: openGL and texture

Posted by [forlano](#) on Tue, 26 Feb 2008 19:06:22 GMT

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

mrjt wrote on Tue, 26 February 2008 19:44In that case the problem is that in the Win32 version GLInit doesn't get called by GLCtrl for some reason.

You need to add the following function to the GLExample class:

virtual void State(int reason)

```
{  
    GLCtrl::State(reason);  
    if (reason == Ctrl::OPEN) {  
        wglMakeCurrent(GLCtrl::GetDC(), GLCtrl::GetHGLRC());  
        GLInit();  
        wglMakeCurrent(NULL, NULL);  
    }  
}
```

You can remove the wglMake current calls from GLInit. Also double check that the file is being loaded. If GLTexture example is successful then texture should be non-zero.

It works!

Thanks a lot!

Now I can proceed to the next tutorial hoping do not disturb again...

Luigi

PS:

GLInit(); should be InitGL(); in the code I posted above

Subject: Re: openGL and texture

Posted by [mrjt](#) on Tue, 26 Feb 2008 19:35:57 GMT

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

You're more than welcome, I did a lot of OpenGL at university so I'm more than happy to help if you have problems.

In fact, the idea of using OpenGL with Upp has me quite excited. I'm trying to think up a project to justify playing with it myself

James
