

---

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

#ifdef PLATFORM_POSIX
#ifdef 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);
```

```
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
    glTexParameterf(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 ).