OpenGL 3 texture example

posted on: 2013-11-11 10:30:20

A bare bones example of using opengl 3 and apply a texture. The whole program is in one file and uses GLFW3.

>Here is the program, minimum_texture2d.cpp. It creates a square composed of two triangles and puts a texture on the square. The texture is a blue and red grid.

The program uses standard OpenGL 3 techniques: creates a context and gl window using GLFW3, creates and compiles a set of shaders into a program, buffers the vertex positions and the texture into the gpu memory; finally starts a loop where the square is drawn to the screen.

I am using a mac to run this program, I'm sure if I built this on linux I would have to add a few ifdef's to get it to work. This is what I use to compile on mac though.

```
GLFWDIR="/path/to/glfw/installation"
g++ -I"$GLFWDIR/include" -L"$GLFWDIR/lib" -framework iokit -framework cocoa -framework opengl -lglfw3 minimum_texture2
```

Here is how the texture is buffered.

```
GLuint texBufferdObject;
glGenTextures(1, &texBufferdObject);
glBindTexture(GL_TEXTURE_2D, texBufferdObject);
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, 8, 8, 0, GL_RGBA, GL_UNSIGNED_BYTE, &texture[0]);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_BASE_LEVEL, 0);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAX_LEVEL, 0);
glBindTexture(GL_TEXTURE_2D, 0);
```

The first parts look just like a normal buffering operation. I do not know the purpose of the glTexParameteri exactly but it was nescessary to make the textures draw.

The next step is to setup the sampler. Generating a sampler is not necessary, it lets us store configurations for the sampler.

```
GLuint sampler=0;
glGenSamplers(1, &sampler);
glSamplerParameteri(sampler, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
const int texUnit=0;
GLuint samplerUniform = glGetUniformLocation(program, "texSampler");
glUniform1i(samplerUniform, texUnit);
glUseProgram(0);
```

The line

glSamplerParameteri(sampler, GL_TEXTURE_MAG_FILTER, GL_NEAREST);

could be replaced with a call to set the TexParameter.

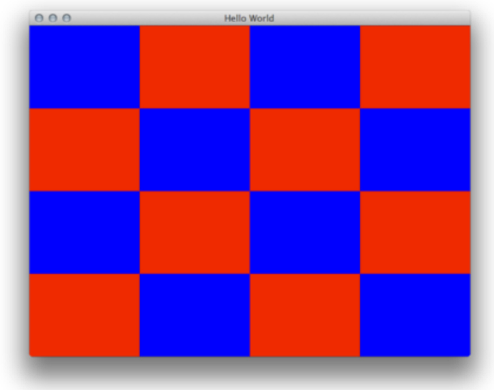
The texUnit is a number for the active texture. When we set the uniform of our samplerUniform to texUnit, it is saying that we will use the uniform sampler2D texSampler; on the active texture texUnit.

This comes up in the drawing code.

```
   glActiveTexture(GL_TEXTURE0+texUnit);
   glBindTexture(GL_TEXTURE_2D, texBufferdObject);
   glBindSampler(texUnit, sampler);
```

We set the glActiveTexture, which is represented by GL_TEXTURE0 plus a texUnit index. We bind the texture we want to use at the active as the active texture position. Then we bind our generated sampler to the same position. So the sampler at texUnit will use the bound texture with the supplied sampler.

Here is the output.



This is the simplest example I could come up with. I built my example from the arcsynthesis tutorials. I should have left out the sampler code but I didn't.

Comments

create comment?