# Texturing in GLSL


**Mike Bailey**
mjb@cs.oregonstate.edu



Oregon State University
Computer Graphics





## 2D Texturing

Vertex shader:

```
#version 330 compatibility
out  vec2  vST;

void
main( )
{
        vST = gl_MultiTexCoord0.st;
        gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
}
```


Fragment shader:

```
#version 330 compatibility
in  vec2  vST;
uniform sampler2D  uTexUnit;

void
main( )
{
        vec3 newcolor = texture( uTexUnit, vST ).rgb;
        gl_FragColor = vec4( newcolor, 1. );
}
```

Oregon State Univers
Computer





## 2D Texturing in the OpenGL API

Fragment shader:

```
#version 330 compatibility
in  vec2  vST;
uniform sampler2D  uTexUnit;

void
main( )
{
        vec3 newcolor = texture( uTexUnit, vST ).rgb;
        gl_FragColor = vec4( newcolor, 1. );
}
```

```
glGenTextures( 1, &TexName );
int nums, numt;
Texture = BmpToTexture( "filename.bmp", &nums, &numt );
glBindTexture( GL_TEXTURE_2D, TexName );
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 );
glTexImage2D( GL_TEXTURE_2D, 0, 3, nums, numt, 0, GL_RGB, GL_UNSIGNED_BYTE, Texture );
. . .
Pattern->Use( );
glActiveTexture( GL_TEXTURE6 );              // use texture unit 6
glBindTexture( GL_TEXTURE_2D, TexName );
Pattern->SetUniformVariable( “uTexUnit”, 6 );
```





## 2D Texturing in glman

Fragment shader:

```
#version 330 compatibility
in  vec2  vST;
uniform sampler2D  uTexUnit;

void
main( )
{
        vec3 newcolor = texture( uTexUnit, vST ).rgb;
        gl_FragColor = vec4( newcolor, 1. );
}
```

```
Texture2D 6 worldtex.bmp
. . .
Program World     uTexUnit  6
```

Oregon State University
Computer Graphics

## The OsuSphere Has Sensible s,t Coordinates Assigned

5




## The GLUT Teapot Doesn't

6



## What if You Want to Use Two Textures in One Shader?

7

**C++ Program:**

```
// In Display( ):
Pattern->Use( );
glActiveTexture( GL_TEXTURE5 );
glBindTexture( GL_TEXTURE_2D, TexName0 );

glActiveTexture( GL_TEXTURE6 );
glBindTexture( GL_TEXTURE_2D, TexName1 );

Pattern->SetUniformVariable( "uTexUnit0", 5 );
Pattern->SetUniformVariable( "uTexUnit1", 6 );

<< draw something >>
Pattern->UnUse( );
```

**Fragment shader:**

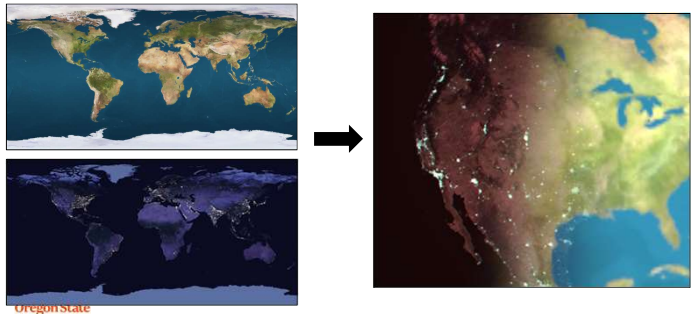
```
#version 330 compatibility
in  vec2  vST;
uniform sampler2D  uTexUnit0;
uniform sampler2D  uTexUnit1;

void
main( )
{
        vec3 newColor0 = texture( uTexUnit0, vST );
        vec3 newColor1 = texture( uTexUnit1, vST );
        gl_FragColor = …
}
```



## Why Might You Want to Use Two Textures in One Shader?

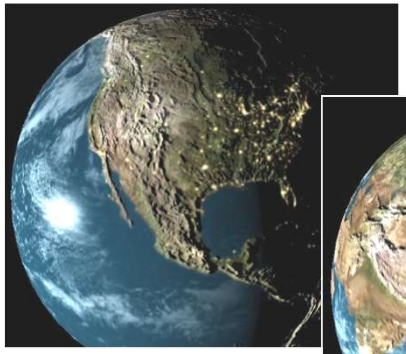
8

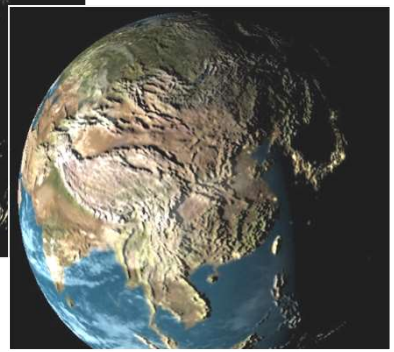
Once the RGBs have been read from a texture, they are just numbers. You can do any arithmetic you want with the texture RGBs, other colors, lighting, etc. Here is an example of blending two textures at once:



## Shaders Can Combine More than Two Textures







Just assign each texture a different texture unit

Textures used here:
- Day
- Night
- Heights (bump-mapping)
- Clouds
- Specular highlights

Visualization by Nick Gebbie