TEXTURE MAPPING
Textures in OpenGL ...

**glEnable(GL_TEXTURE_2D)**

- turn on the 2D texture store

**glTexImage2D**

- declares a texture’s size, color components (RGBA, etc), data type (byte, float...), pixel data

**glBindTexture**

- “bind” the given texture to the active store. Only one texture can be bound at a time. All future configuration and co-ordinates correspond to this texture. For the fixed pipeline, you can only use one texture for rendering. For shaders, you bind textures to uniforms
Textures in OpenGL Continued...

**glTexParameteri**

Used to set texture configuration:

How are the texture values interpolated?
GL_NEAREST vs GL_LINEAR
- GL_NEAREST rounds to nearest texel to get the color
- GL_LINEAR linearly interpolates the colors of the texels

Does the texture repeat itself?
GL_REPEAT vs GL_CLAMP
- Say we have a texture coordinate of (-0.1,1.1)
- GL_CLAMP changes it to (0.0,1.0)
- GL_REPEAT changes it to (0.9,0.1)

More options for Texture Parameters can be found here: [http://www.opengl.org/sdk/docs/man/xhtml/glTexParameter.xml](http://www.opengl.org/sdk/docs/man/xhtml/glTexParameter.xml)
Examples of use can be found in the Texture class in the framework
Textures in CS 4620 Framework ...

Takes the burden of:

- Loading texture files as texture maps (\texttt{glTexImage2D})
- Setting up the texture parameters (\texttt{glTexParameteri})
- Managing the texture units (\texttt{glBindTexture})

Wrapper classes for working with 1D, 2D and 2D Mip-Mapped textures.

Simple interface for using textures with GLSL.
Texturing in GLSL/Pipeline
Texturing in GLSL

New elements:

sampler2D (type)

vec4 texture2D(sampler2D, vec2) (function)
Texturing in GLSL – Vertex Shader

Figure out the coordinate that we want to sample from using an attribute variable

```glsl
attribute vec2 in_TexCoord;
varying vec2 coord;

void main() {
    gl_Position = ...
    coord = vec2(in_TexCoord);
}
```
Texturing in GLSL – Fragment Shader

Take the coordinate data from the vertex shader and sample the appropriate pixel from the desired texture

```glsl
varying vec2 coord;
uniform sampler2D sampler;

void main() {
    gl_FragColor = texture2D(sampler, coord);
}
```
Inside Init()

// Load the 2D texture
texture = new GLTexture(TextureTarget.Texture2D, true);

try {
    texture.setImage2DResource("path/to/image.png", false);
} catch (Exception e) {
    // No image found!
    System.out.println(e.getMessage());
    System.exit(1);
}
Textures in CS 4620 Framework ...

Inside Render()

program.use(); // Activate the shader

// Activate the texture and bind the active texture unit
// to the sampler uniform
texture.use(TextureUnit.Texture0,
    program.getUniform("sampler"));

glDrawElements(..); // Render your scene

// clean up
texture.unuse();
GLProgram.unuse();