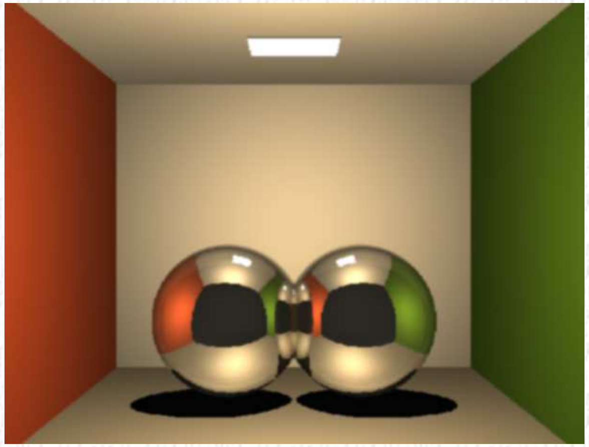
Lab 6

Kostas Vardis

http://graphics.cs.aueb.gr/graphics/people.html

Athens University of Economics and Business

Computer Graphics BSc



Introduction

- Texture mapping
- Texture wrapping
- Texture filtering
- Texture sampling

Textures

Create Textures:

```
// Generate the OpenGL texture id
glGenTextures(1, &texture_id);
// Bind this texture to its id
glBindTexture(GL_TEXTURE_2D, texture_id);
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, 512, 512,
    0, GL_RGBA, GL_UNSIGNED_BYTE, NULL);
```

Texture Mapping

C++ (in Render):

```
// check if diffuse texture is present
if (texture_exists) {
// activate texture unit 0 for the diffuse texture
glActiveTexture(GL_TEXTURE0);
// bind the diffuse texture to the active texture unit
glBindTexture(GL_TEXTURE_2D, texture_id);
}
// pass the sampler to GLSL
glUniform1i(uniform_sampler, 0);
```

Texture Mapping

In Vertex Shader:

```
layout(location = 2) in vec2 texcoord0;
out vec2 texcoord;

void main(void) {
// pass the texture coordinates
texcoord = texcoord0;
……
}
```

Texture Mapping

In Fragment Shader:

```
// the incoming texture coordinates from the vertex shader
in vec2 texcoord;
// samplers
uniform sampler2D sampler_diffuse;
 void main(void) {
vec4 final_color = uniform_material_color;
final_color *= texture(sampler_diffuse, texcoord.xy);
out_color = final_color;
}
```

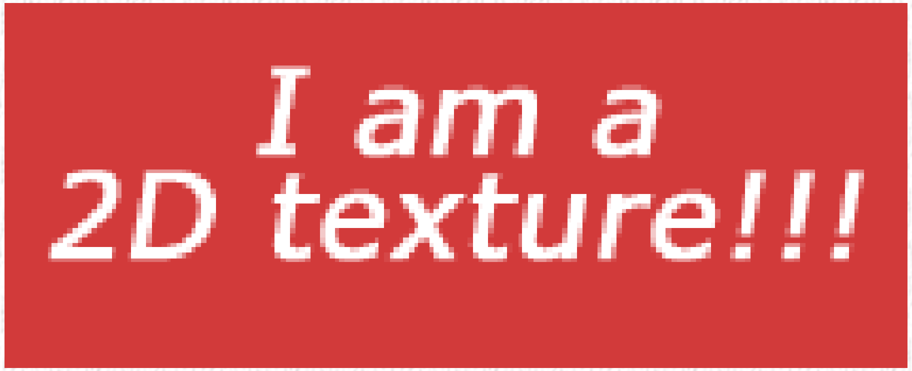
Filtering

- Filtering based on the texel:pixel mapping ratio
- Texel covers more than one pixel -> magnification filter
- Texel covers less than one pixel-> minification filter

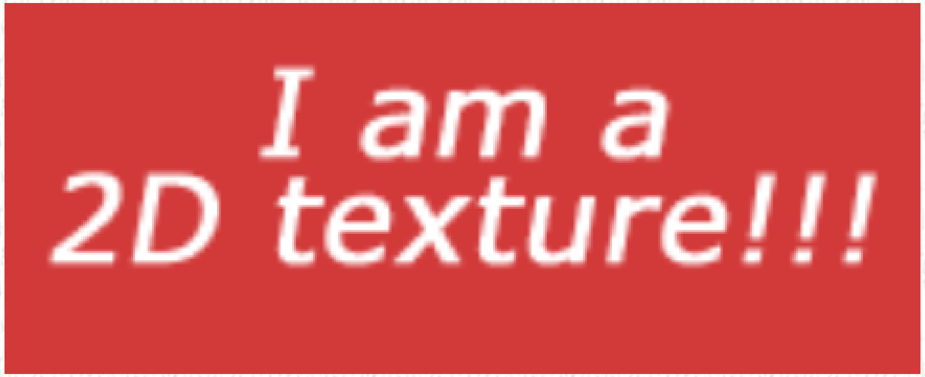
Filtering

Two main modes

Gl_NEAREST



GL_LINEAR



Mipmaps

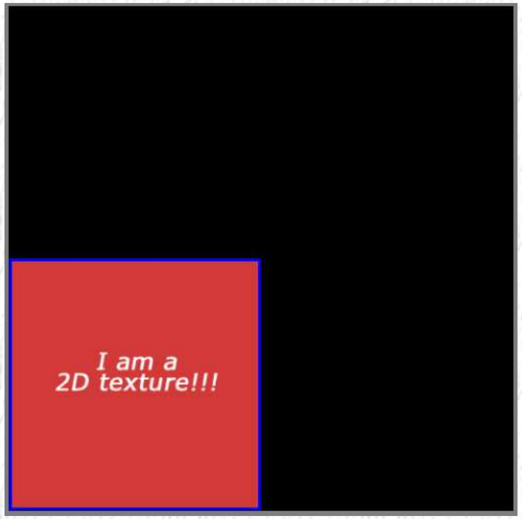
- When more than one texels cover a pixel (minification filter), nearest and linear filtering are not enough
- Mipmapping generates a sequence of lower-resolution images

Wrapping

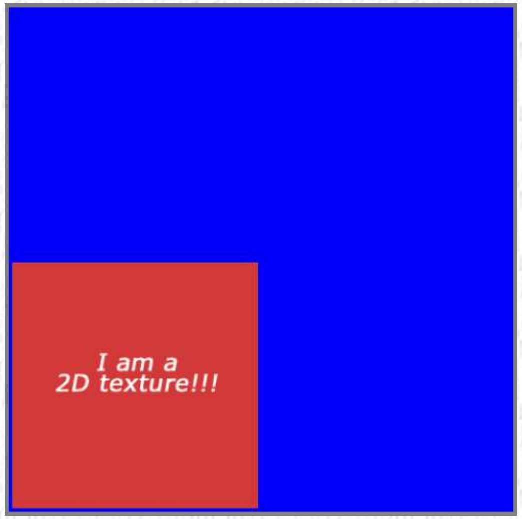
- Controls what happens beyond [0, 1) uv range
- GL_Clamp_to_border sets a border color
- GL_Clamp_to_edge sets the edge color
- GL_Repeat causes the coordinates to repeat (no integer part)

Wrapping

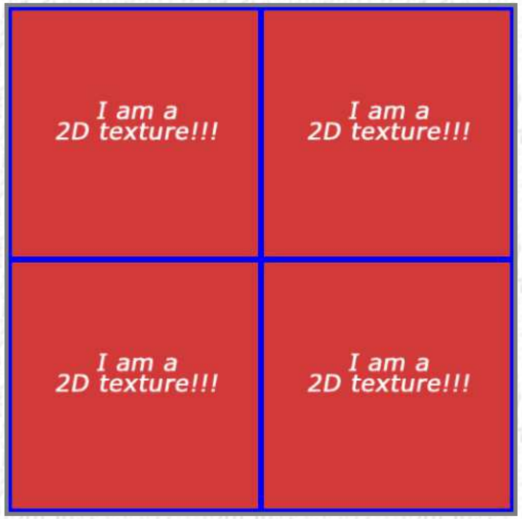
GL_CLAMP_TO_BORDER



GL_CLAMP_TOO_EDGE



GL_REPEAT



Filtering-Wrapping OpenGL

```
// wrapping
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S,
GL_REPEAT/GL_CLAMP_TO_BORDER/GL_CLAMP_TO_EDGE/)
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T,
GL_REPEAT/GL_CLAMP_TO_BORDER/GL_CLAMP_TO_EDGE/)
// mag filtering
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER,
GL_NEAREST or GL_LINEAR);
// min filtering (no mipmaps)
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER,
GL_NEAREST or GL_LINEAR);
```

Filtering-Wrapping OpenGL

```
// min filtering (mipmaps)
glTexParameteri(GL_TEXTURE_2D,
GL_TEXTURE_MIN_FILTER, …);
```

- GL_NEAREST_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_LINEAR

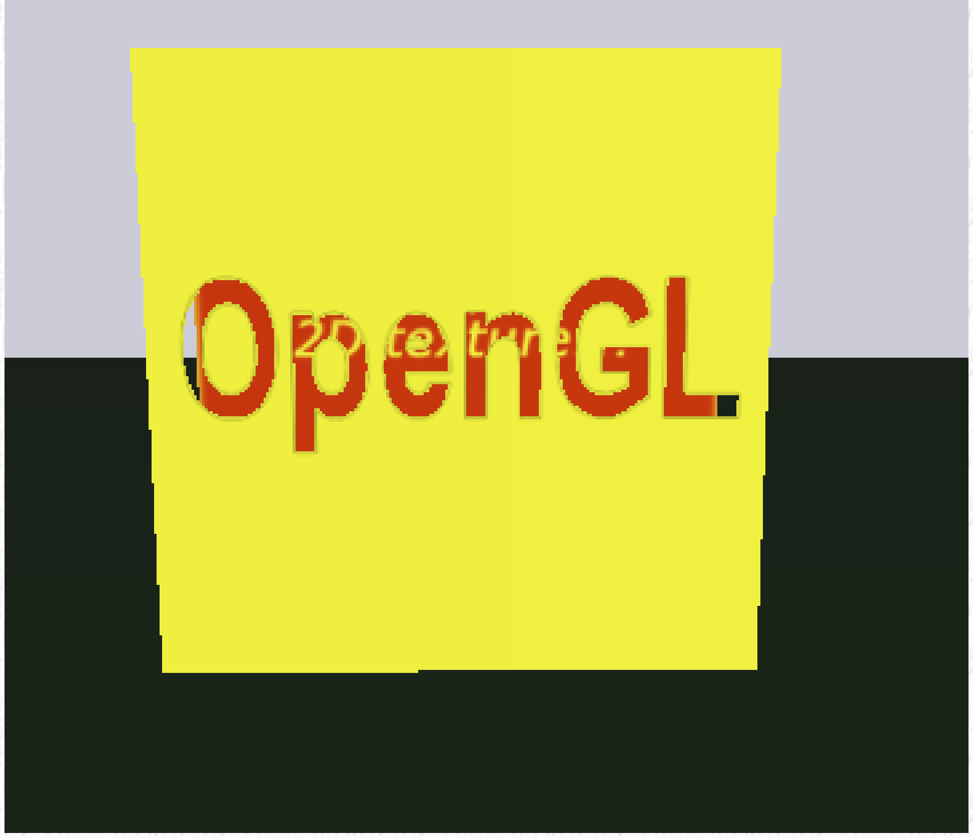
Alpha testing

- Can discard a fragment based on alpha value

Example:

```
vec4 tex_value = texture(uniform_sampler_diffuse, texcoord.xy);
if (uniform_has_sampler_diffuse > 0) {
diffuse_tex *= tex_value;
if (diffuse_tex.a < 1.0) discard;
}
```

Alpha testing



Done!

Check lab6 project