

# Textures

## Lecture 26

Robb T. Koether

Hampden-Sydney College

Fri, Nov 8, 2019

# Outline

- 1 Textures
- 2 Creating Textures
- 3 Assignment

# Outline

- 1 Textures
- 2 Creating Textures
- 3 Assignment

## Definition (Texture)

A **texture** is an image that can be “pasted” onto a surface.

- The image may be a simple one that was generated within the program, such as a checkerboard pattern, or it may be a photograph that was read from a file.
- Either way, the image consists of “texture elements” called **texels**.
- These are analogous to **pixels** (which are “picture elements.”)

# Texture Format

- There are many internal formats for textures.
- See pages 280-281 of the Red Book.
- Among them are the following.
  - `GL_RGB` - Each texel (texture element) stored in 3 bytes (red, green, blue)
  - `GL_RGBA` - Each texel stored in 4 bytes (red, green, blue, alpha)
  - `GL_R3_G3_B2` - Each texel stored in 1 byte (3 red bits, 3 green bits, 2 blue bits), etc.
- We will use the `GL_RGB` format.

# Outline

1 Textures

**2 Creating Textures**

3 Assignment

# Creating Textures

- The `GL_RGB` texture image is created as a 3-dimensional array.

```
GLubyte image[rows][cols][3];
```

- Each vector contains the red, green, and blue values as unsigned bytes from 0 to 255.

# Creating Textures

- Or it could be a 3-dimensional array of **floats**.

```
GLfloat image[rows][cols][3];
```

- Each vector contains the red, green, and blue values as **floats** from 0.0 to 1.0.



# Creating Textures

- Or it could be a 2-dimensional array of `vec3`s.

```
vec3 image[rows][cols];
```

- Each vector contains the red, green, and blue values as `floats` from 0.0 to 1.0.

# Creating Textures

- Or it could be a 2-dimensional array of `vec3`s.

```
vec3 image[rows][cols];
```

- Each vector contains the red, green, and blue values as `floats` from 0.0 to 1.0.
- Or in many other forms.

# Creating Textures

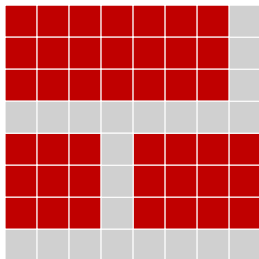
- There are also 1-dimensional textures, which are pasted onto lines and 3-dimensional textures, which are pasted into solids.
- And there are 1-dimensional arrays of 1-dimensional textures, and 1-dimensional arrays of 2-dimensional textures.
- We will do only 2-dimensional textures.

# Creating Textures

- The dimensions of a texture image must be powers of 2, but they need not be the same power of 2.
- Typical dimensions
  - $256 \times 256$
  - $256 \times 512$
  - $32 \times 32$

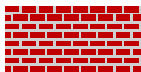
# Example

- A brick wall texture may be designed as an  $8 \times 8$  texture.



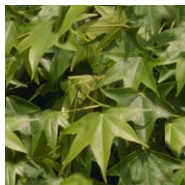
# Tiling Textures

- Often textures are “tiled” across a surface.
- When they are tiled, it is desirable that the images match left-to-right and top-to-bottom.
- The brick wall texture:



# Tiling Textures

- Tiling textures works well if the texture is supposed to have a regular pattern, such as in the case of the brick wall.
- Other textures look better if there is no discernable pattern, such as leaves.

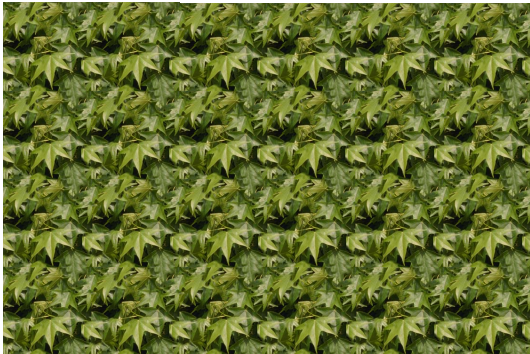


# Tiling Textures





# Tiling Textures



# Window Texture

- Other textures are not meant to be tiled.



# Outline

1 Textures

2 Creating Textures

**3 Assignment**

# Homework

## Homework

- Read pages 255 - 260: Introduction to Texturing & Basic Texture Types
- Read pages 279 - 283: Texture Formats