Universitat de Girona

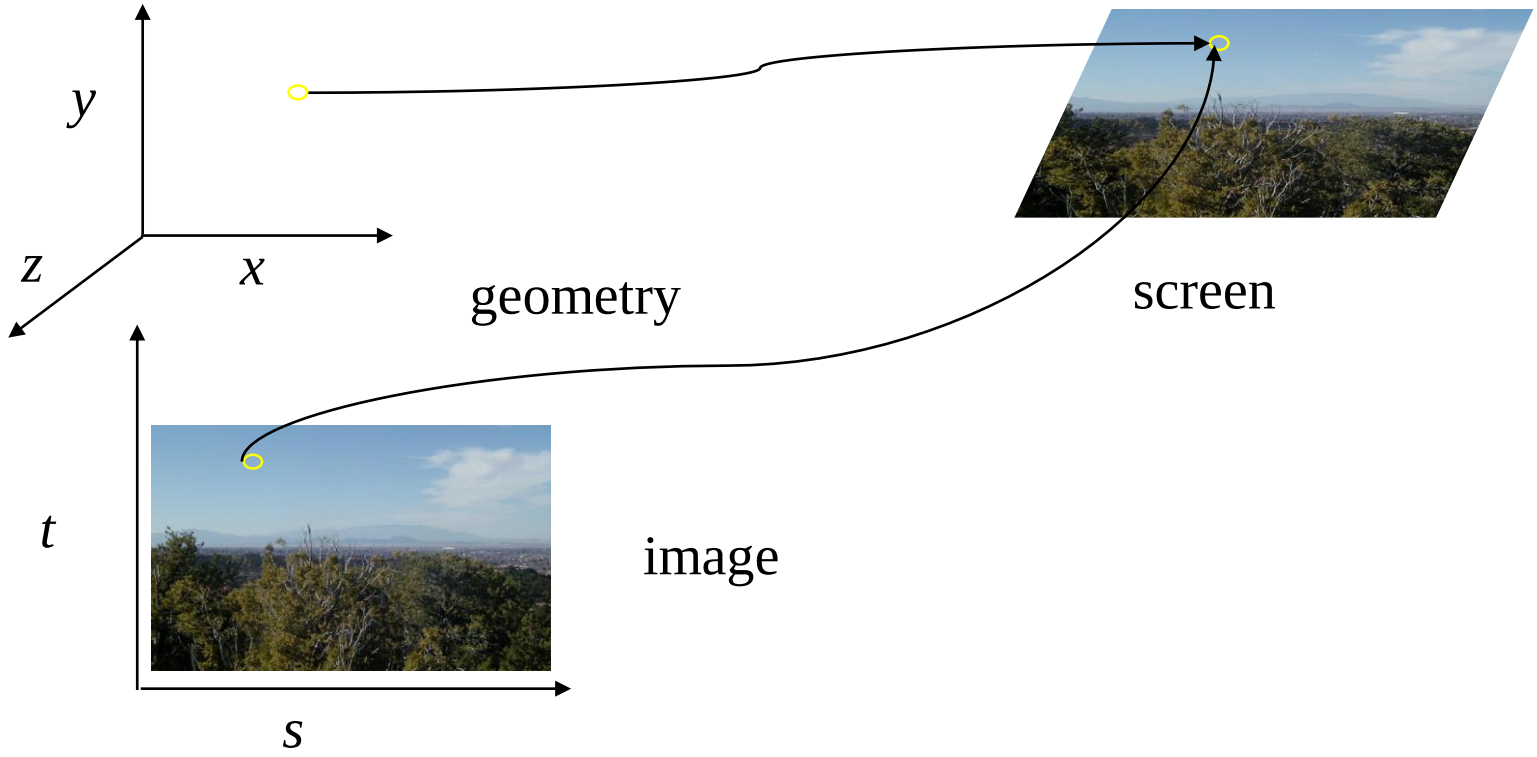
Textures in WebGL

Gonzalo Besuievsky

IMAE - UdG

Universitat de Girona

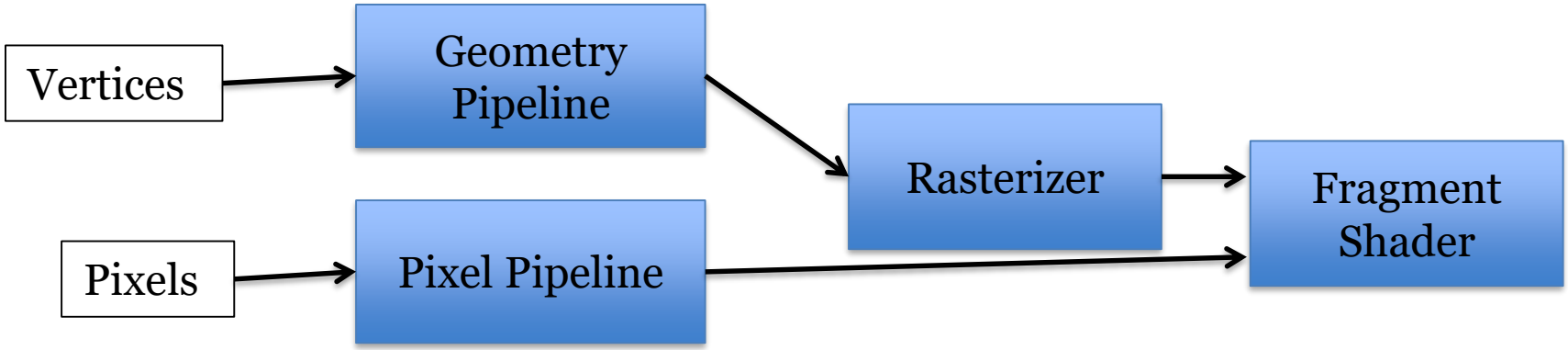
Texture Mapping



Universitat de Girona

Texture Mapping and the OpenGL Pipeline

- Images and geometry flow through separate pipelines that join at the rasterizer
  - "complex" textures do not affect geometric complexity



Universitat de Girona

Applying Textures procedure

- Three basic steps for applying a texture
  1. specify the texture
     - »read (or generate) the image
     - »assign to texture
     - »enable texturing
  2. assign texture coordinates to vertices
  3. specify texture parameters
     - »wrapping, filtering

Universitat de Girona

Specifying a Texture Image

- Define a texture image from an array of *texels* (texture elements) in CPU memory
- Use an image in a standard format (such as JPEG)
  - Scanned image
  - Generate by application code
- WebGL 1.0 supports only 2 dimensional texture maps
  - no need to enable as in desktop OpenGL
- WebGL 2 supports 3 dimensional texture maps
- Desktop OpenGL supports 1-4 dimensional texture maps

Universitat de Girona

Define Image as a Texture

```
gl.texImage2D( target, level, components,
    w, h, border, format, type, texels );
```

`target:` type of texture, e.g. `gl.TEXTURE_2D`
`level:` used for mipmapping
`components:` elements per texel
`w, h:` width and height of `texels` in pixels
`border:` used for smoothing
`format and type:` describe texels
`texels:` pointer to texel array

Example:

```
gl.texImage2D(gl.TEXTURE_2D, 0, 3, 512, 512, 0, gl.RGB, gl.UNSIGNED_BYTE,
 my_texels);
```

Universitat de Girona

Texture Objects

- Have WebGL store your images
  - one image per texture object
- Create an empty texture object

```
var  texture = gl.createTexture();
```

- Bind textures before using

```
gl.bindTexture( gl.TEXTURE_2D, texture );
```

Universitat de Girona

Specifying a Texture Image

- Define a texture image from an array of *texels* in CPU memory

```
gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, texSize,
        texSize, 0,  gl.RGBA, gl.UNSIGNED_BYTE, image);
```

- Define a texture image from an image in a standard format memory specified
  - –For example, with the <image> tag in the HTML file

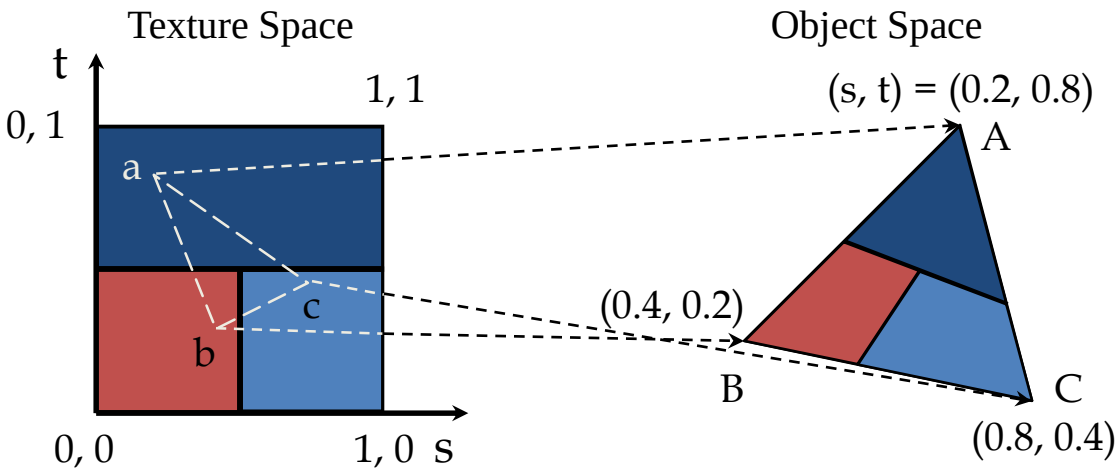
```
var image = document.getElementById("texImage");

gl.texImage2D( gl.TEXTURE_2D, 0, gl.RGB, gl.RGB, gl.UNSIGNED_BYTE, image );
```

Universitat de Girona

Mapping a Texture

- Based on parametric texture coordinates
- Specify as a 2D vertex attribute



Universitat de Girona

Cube Example

```
var exampleCube = {  // 24 vértices, 12 triángulos

  "vertices" : [-0.5, -0.5,  0.5,   0.0,  0.0,  1.0,   0.0, 0.0,
                 0.5, -0.5,  0.5,   0.0,  0.0,  1.0,   1.0, 0.0,
                 0.5,  0.5,  0.5,   0.0,  0.0,  1.0,   1.0, 1.0,
                -0.5,  0.5,  0.5,   0.0,  0.0,  1.0,   0.0, 1.0,
                 0.5, -0.5,  0.5,   1.0,  0.0,  0.0,   0.0, 0.0,
                 0.5, -0.5, -0.5,   1.0,  0.0,  0.0,   1.0, 0.0,
                 0.5,  0.5, -0.5,   1.0,  0.0,  0.0,   1.0, 1.0,
                 0.5,  0.5,  0.5,   1.0,  0.0,  0.0,   0.0, 1.0,
                 0.5, -0.5, -0.5,   0.0,  0.0, -1.0,   0.0, 0.0,
                -0.5, -0.5, -0.5,   0.0,  0.0, -1.0,   1.0, 0.0,
                -0.5,  0.5, -0.5,   0.0,  0.0, -1.0,   1.0, 1.0,
                 0.5,  0.5, -0.5,   0.0,  0.0, -1.0,   0.0, 1.0,
                -0.5, -0.5, -0.5,  -1.0,  0.0,  0.0,   0.0, 0.0,
                -0.5, -0.5,  0.5,  -1.0,  0.0,  0.0,   1.0, 0.0,
                -0.5,  0.5,  0.5,  -1.0,  0.0,  0.0,   1.0, 1.0
                -0.5,  0.5, -0.5,  -1.0,  0.0,  0.0,   0.0, 1.0,
                -0.5,  0.5,  0.5,   0.0,  1.0,  0.0,   0.0, 0.0,
                 0.5,  0.5,  0.5,   0.0,  1.0,  0.0,   1.0, 0.0,
                 0.5,  0.5, -0.5,   0.0,  1.0,  0.0,   1.0, 1.0,
                -0.5,  0.5, -0.5,   0.0,  1.0,  0.0,   0.0, 1.0,
                -0.5, -0.5, -0.5,   0.0, -1.0,  0.0,   0.0, 0.0,
                 0.5, -0.5, -0.5,   0.0, -1.0,  0.0,   1.0, 0.0,
                 0.5, -0.5,  0.5,   0.0, -1.0,  0.0,   1.0, 1.0
                -0.5, -0.5,  0.5,   0.0, -1.0,  0.0,   0.0, 1.0],

  "indices" :  [  0,  1,  2,  0,  2,  3,
                  4,  5,  6,  4,  6,  7,
                  8,  9, 10,  8, 10, 11,
                 12, 13, 14, 12, 14, 15,
                 16, 17, 18, 16, 18, 19,
                 20, 21, 22, 20, 22, 23]

};
```

Texture coordinates

Universitat de Girona

Basic Shaders for 2D Texture

```
// Vertex shader
...
in  vec2 VertexTexcoords;  // nuevo atributo
out vec2 texCoords;

void main()  {
   ...
   // se asignan las coordenades de textura del vertice a la variable texCoords
   texCoords    = VertexTexcoords;
}

// Fragment shader
...
uniform sampler2D myTexture;     // la textura
in vec2 texCoords;               // coordenadas de textura interpoladas

void main()
{
   ...
   // acceso a la textura para obtener un valor de color RGBA
   fragmentColor = texture(myTexture, texCoords);
}
```

type

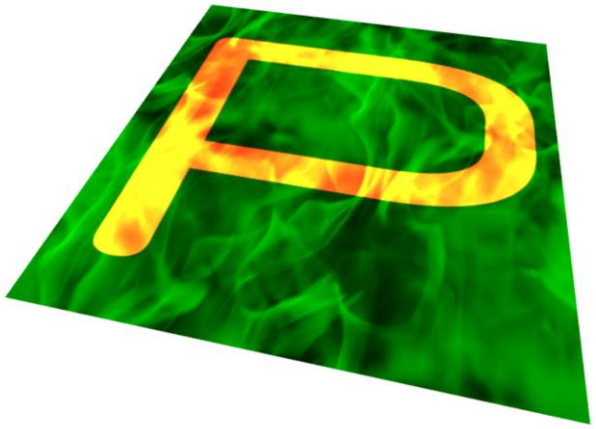
GLSL function

Universitat de Girona

Multiple textures

```
// Fragment shader
...
uniform sampler2D myTexture1, myTexture2;     // las texturas
in vec2 texCoords;     // coordenadas de textura interpoladas

void main()
{
   ...
   // acceso a la textura para obtener un valor de color RGBA
   fragmentColor = texture(myTexture1, texCoords) * texture(myTexture2, texCoords);
}
```



Universitat de Girona

Texture Parameters

- WebGL has a variety of parameters that determine how texture is applied
  - Wrapping parameters determine what happens if s and t are outside the (0,1) range
  - Filter modes allow us to use area averaging instead of point samples
  - Mipmapping allows us to use textures at multiple resolutions
  - Environment parameters determine how texture mapping interacts with shading

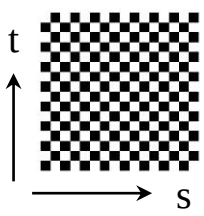
Universitat de Girona

Wrapping Mode

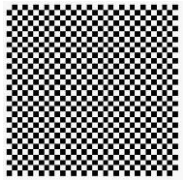
Clamping: if s,t > 1 use 1, if s,t <0 use 0

Wrapping: use s,t modulo 1

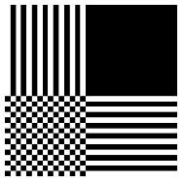
```
gl.texParameteri(gl.TEXTURE_2D,
     gl.TEXTURE_WRAP_S, gl.CLAMP )
gl.texParameteri( gl.TEXTURE_2D,
     gl.TEXTURE_WRAP_T, gl.REPEAT )
```



texture



gl.REPEAT wrapping



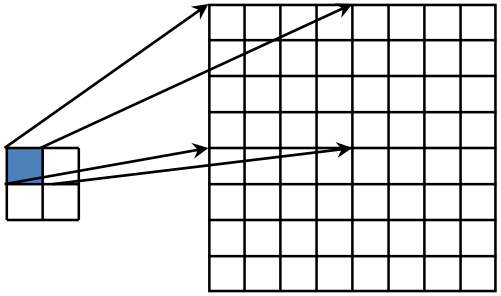
gl.CLAMP wrapping

Universitat de Girona

Magnification and Minification

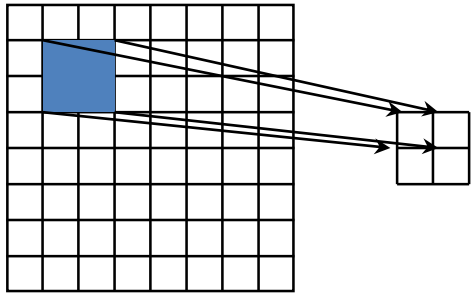
More than one texel can cover a pixel (*minification*) or more than one pixel can cover a texel (*magnification*)

Can use point sampling (nearest texel) or linear filtering (2 x 2 filter) to obtain texture values



Texture Polygon

Magnification



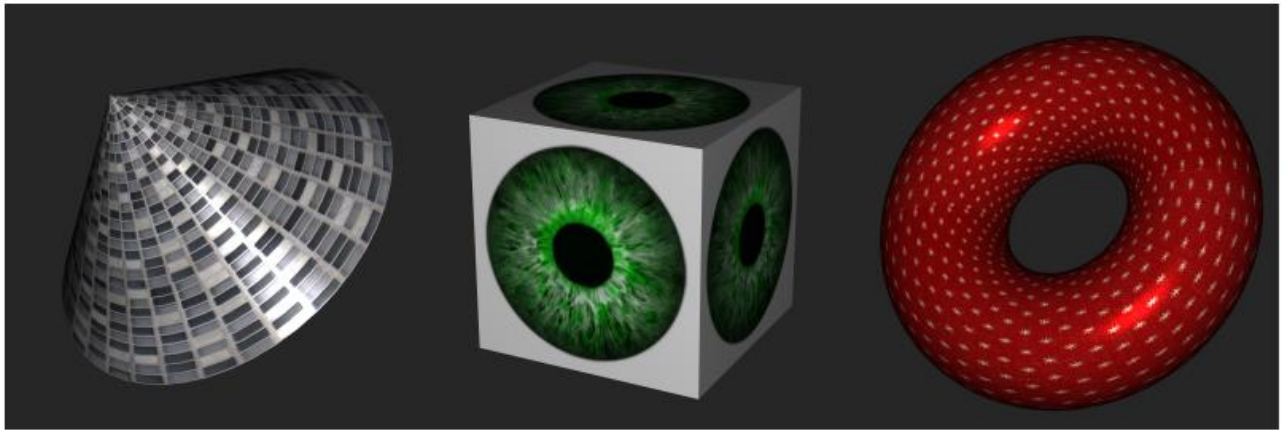
Texture Polygon

Minification

Universitat de Girona

Applying Textures

1. Create a texture object
2. Assign the unit texture
3. Specify vertex coordinate and texture coordinates



Universitat de Girona

Create texture

```
// Crea un objeto textura
texture = gl.createTexture();
gl.bindTexture(gl.TEXTURE_2D, texture);

// Especifica la textura RGB
gl.texImage2D (gl.TEXTURE_2D, 0, gl.RGB, image.ancho, image.alto, 0, gl.RGB, gl.UNSIGNED_BYTE,
      image);

// Repite la textura tanto en s como en t
gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_WRAP_S, gl.REPEAT);
gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_WRAP_T, gl.REPEAT);

// Filtrado
gl.texParameteri (gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER, gl.LINEAR_MIPMAP_LINEAR);
gl.texParameteri (gl.TEXTURE_2D, gl.TEXTURE_MAG_FILTER, gl.LINEAR);
gl.generateMipmap(gl.TEXTURE_2D);
```

Assign texture: each object can have different textures

```
// Selecciona la unidad de textura 0
gl.activeTexture(gl.TEXTURE0);

// Asigna el objeto textura a la unidad de textura seleccionada
gl.bindTexture  (gl.TEXTURE_2D, texture);
```

Universitat de Girona

Link the shader variable with the texture unit

```
// Obtiene el indice de la variable del Shader de tipo sampler2D
program.textureIndex = gl.getUniformLocation(program, 'myTexture');

// Indica que myTexture del Shader use la unidad de textura 0
gl.uniform1i(program.textureIndex, 0);
```

Enable texture coordinate attribute

```
// se obtiene la referencia al atributo
program.vertexTexcoordsAttribute =
  gl.getAttribLocation ( program, "VertexTexcoords");

// se habilita el atributo
gl.enableVertexAttribArray(program.vertexTexcoordsAttribute);
```

Universitat de Girona

Example

- Download and execute "*aplicaTextura.html*"
- Edit and Change
  - –the geometric object
    - Change primitive
  - –the texture files
    - Change image
  - –The texture parameters
    - Change repeat

Universitat de Girona

Example 2

- Download and execute
  - "*dames.html*"
  - "*rayado.html*"
  - *"enrejado.html"*
- Identify the function generation
- Try to change the parameter generation

Universitat de Girona

Assignment 4

- How to manage different shaders
  - Manage several shaders: (vs1, fs1), (vs2, fs2) …
  - Manage opcions with conditional and flags:

```
If (modo==1){ // use procedural texture A}
else if {modo==2) ){ // use procedural texture B}
else if {modo==3) ){ // use texture images}
else …
```

- How to manage many lights
  - The total reflection at p is the sum of all contributed intensities from all sources
  - WebGL allows us to define several light sources (as uniforms)
    - For each light … add the contribution

Universitat de Girona

Use textured objects

See aplicaTexturasUVMap_OBJ.html
Note: open the image "B5-cottage_obj\cottage_diffuse.png" with the file manager

Universitat de Girona

Use textured objects

1. Get a textured object (Blender, hand made or internet)
2. Export to OBJ (with triangles and text coord)
3. Convert to JSON: geometry and image atlas
4. Manage the texture in webgl