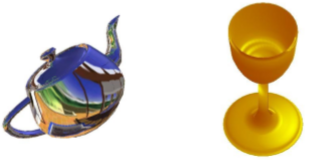
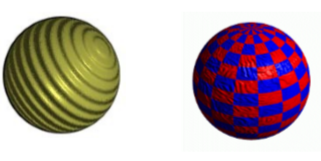
**Texture Mapping**

**Basic Texture Mapping**

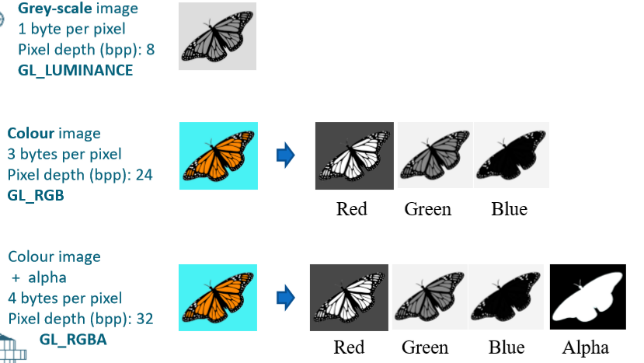
* Per vertex mapping
* Adds colour based surface features to polygons
* More realistic

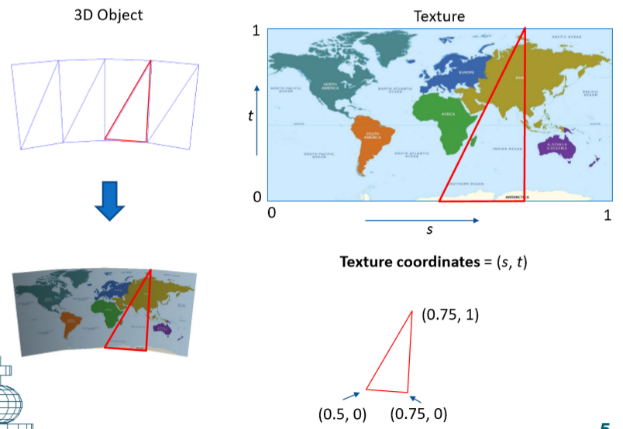
**Advanced Applications**

Environment Mapping = reflections of objects in the world environment on another

Billboarding = View oriented texture mapped polygons commonly used in place of models of trees.

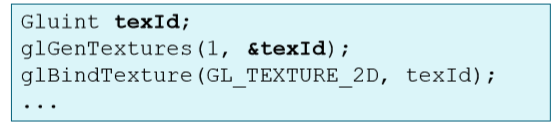
Bump Mapping = surface displacements without modifying the geometry, to create the appearance of bumps and wrinkles.

 **Image Types**

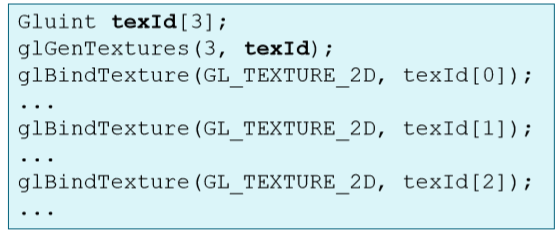
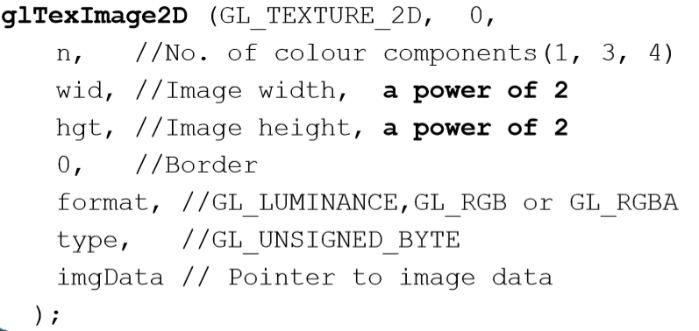
 **Texture Mapping** (5 steps)

**Step 1**

1 Texture

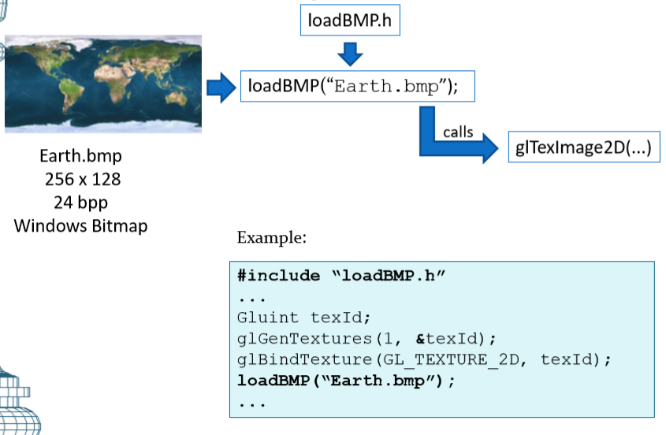
* **Get** the texture ID = by using **glGenTextures**()

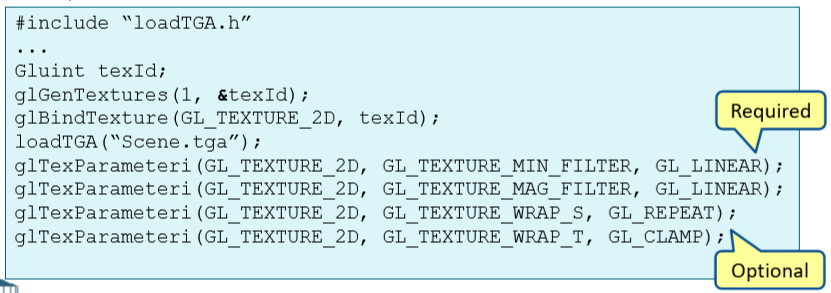
**Step 2**

* **Load** a texture by calling the function:

3 Texture ->

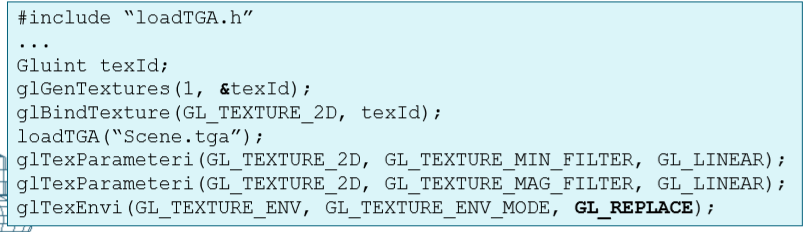
**Loading Textures**



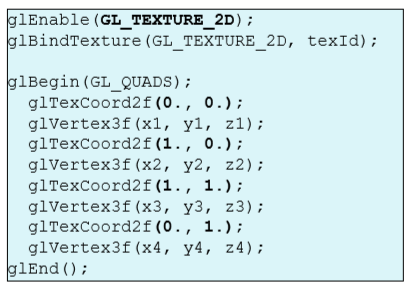
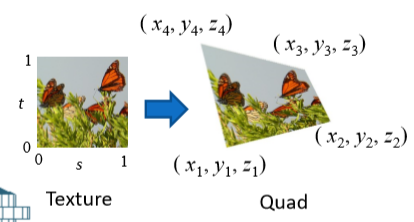
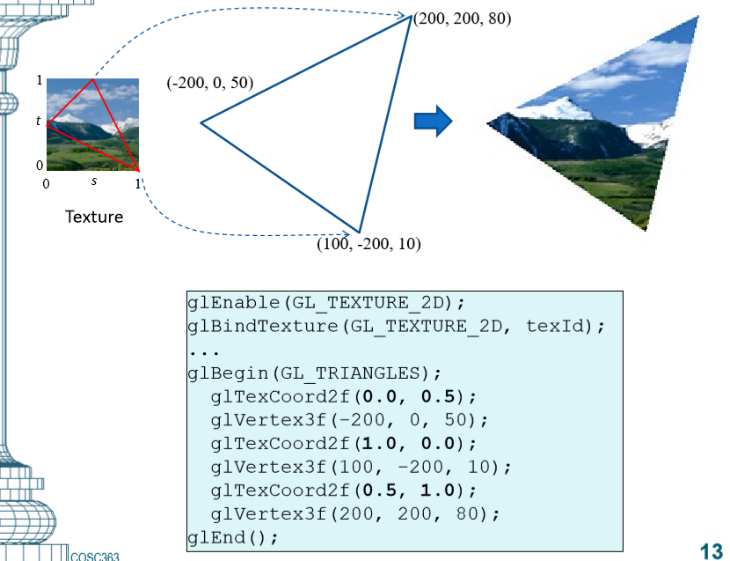
**Step 3**

* **Set** texture sampling parameters:
* GL\_TEXTURE\_MIN\_FILTE = small amount of polygons
* GL\_TEXTURE\_MAG\_FILTE = Large amount of polygons

**Step 4**

* **Set** texture environment parameters:
* GL\_REPLACE = Texture colour replaces the fragment’s colour
* GL\_MODULATE = Texture colour is multiplied by fragment’s colour

**Step 5**

* **Enable** texturing and assign texture coordinates to vertices.

Example

**Texture Image Types**

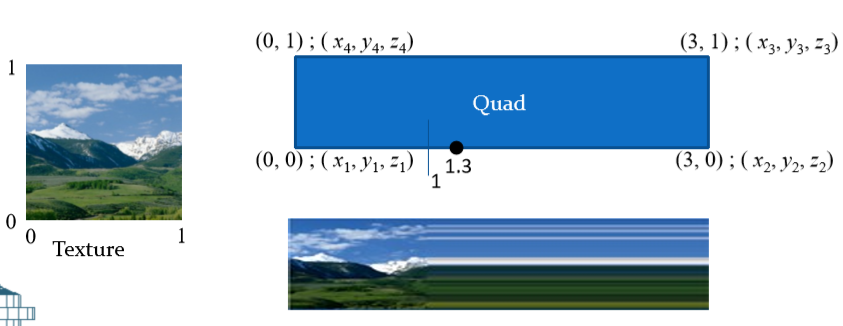
* Bitmap (.bmp) : 24 bits per pixel, Windows bitmap.

Use **loadBMP**() function included in **loadBMP**.h

* Targa (.tga): 24 bits per pixel, Uncompressed.

Use **loadTGA**() function included in **loadTGA**.h

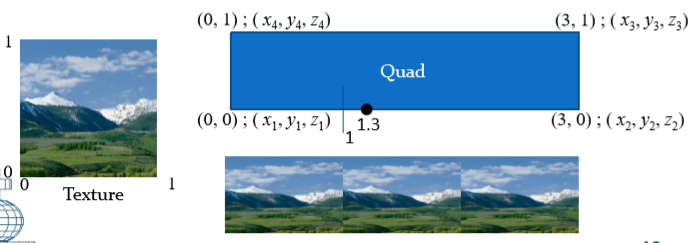
* <https://www.textures.com/> is a good texture website
* Use resolution (1024x1024 or lower)



**Texture Tiling**

* If texture axis is set to **GL\_CLAMP**, then the coordinate value is clamped to the range [0, 1]

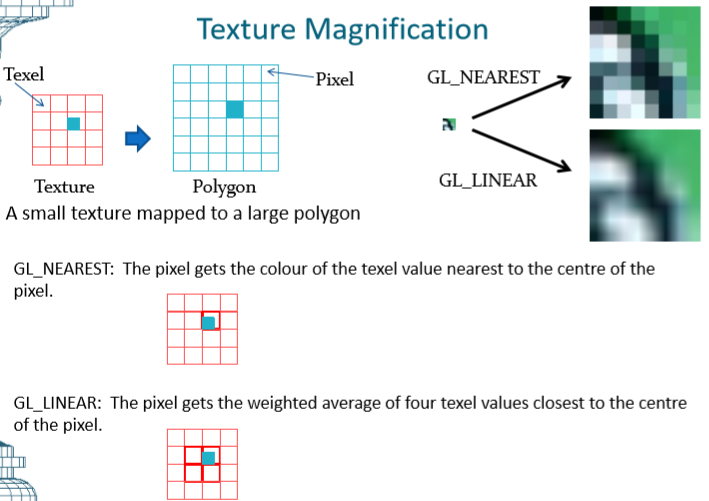
**glTexParameteri**(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_S, GL\_CLAMP);

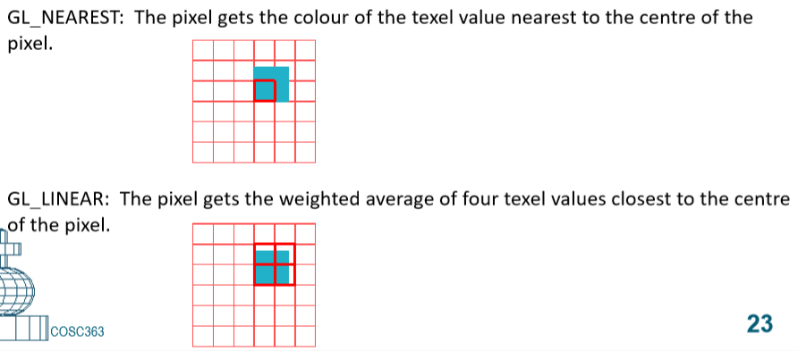
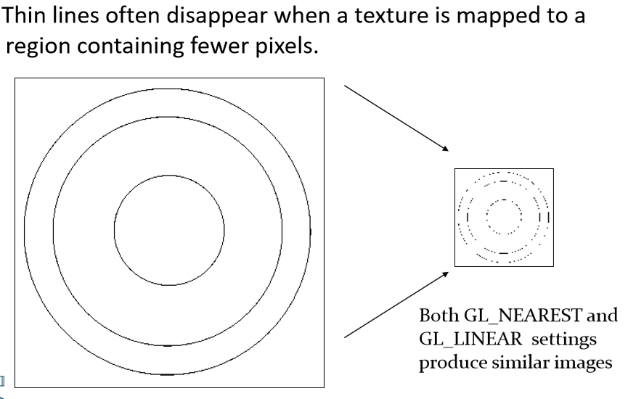
* If the wrap parameter for a texture axis is set to GL\_REPEAT, then the integer part of the texture coordinate along that axis is ignored. (eg. A value 1.3 is treated as 0.3). This results in the tiling of the image along that axis.

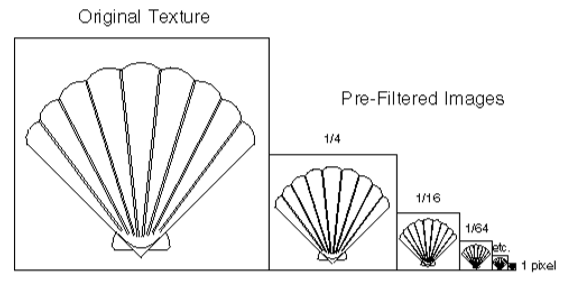
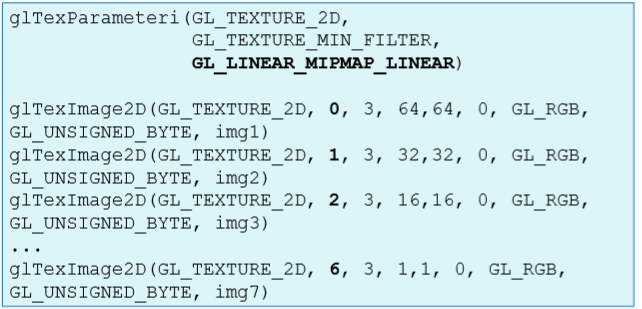
**glTexParameteri**(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_S, GL\_REPEAT);

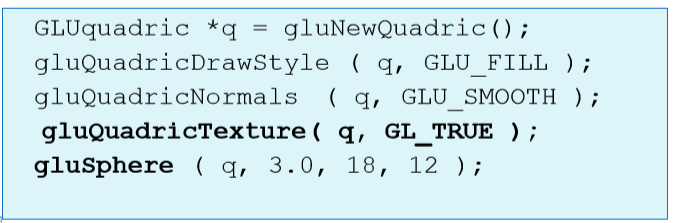
**Texture Sampling**

* 2 samples, GL\_NEAREST, GL\_LINEAR
* …

**Texture Magnification**

* GL\_NEAREST = The pixel gets the colour of texel closest to the centre of the pixel.
* GL\_LINEAR = The pixel gets weighted of four texel values closest to centre.
* problem !!!
* This can be fixed by using Mipmaps
* MIP = Multum In Parvo
* Mipmaps scales the image
* Mipmapping requires additional processing, and 33% extra texture storage space.
* Mipmaps can only be used if **GL\_TEXTURE\_MIN\_FILTER**  is specified in **glTexParameteri(…)**





**Texturing a Quadric Surface**

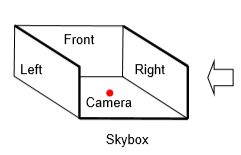
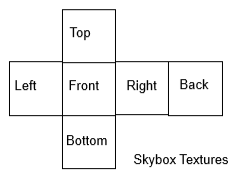
* Functions in the **GLU** library
* Quadric surfaces have a two-dimensional parametric representation which can be used to get a mapping to the texture coordinate space (s, t).

**Texturing and Lighting**

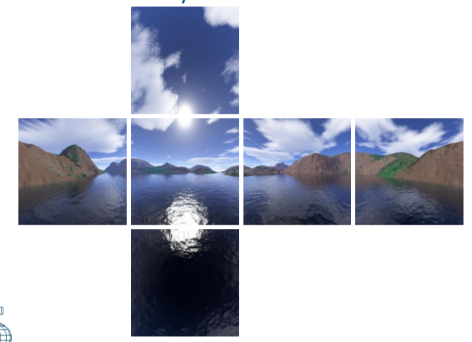
1. Lighting computation = per-vertex operation.
2. Texturing = fragment processing stage.

If **GL\_REPLACE** is used as the texturing environment, the colour values got from lighting computation would be replaced with texture colours.

* In order to see the variation of diffuse reflections from the surface, the texture values must be modulated with the already computed fragment colour (GL\_MODULATE)
* Modulation will reduce the effect of specular highlights. To get a strong specular highlight on a textured surface, select the following light model: **glLightModeli**(GL\_LIGHT\_MODEL\_COLOR\_CONTROL, GL\_SEPARATE\_SPECULAR\_COLOR);

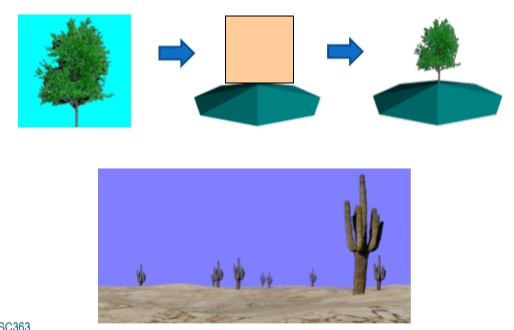
**Sky Boxes**

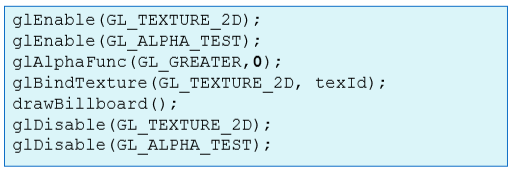
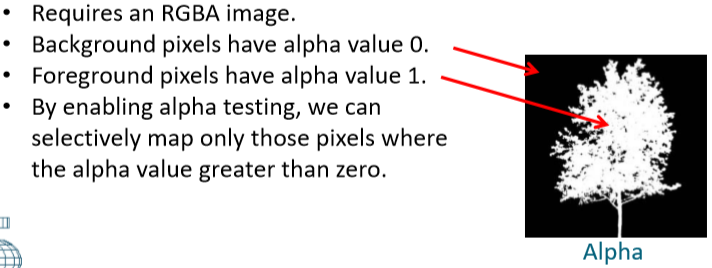
* Environment = textures on the faces of a large cube, and the cube is rendered centred around the view point.

Minimise perspective distortions:

* Adjusting the focal length (“**near**” value in **gluPerspective**) and the field of view (“**fov**” value in **gluPerspective**)
* Adjusting the size of the cube used for texture mapping .
* Not moving the camera very close to the four sides of the cube.

**Billboarding**

* Technique that changes the orientation of texture mapped quads in a 3D environment based on view direction.
* When a texture is mapped to a quad, only those pixels belonging to the object are rendered. The background of the texture is removed using alpha texturing.

******Alpha Texturing**

**Decals**

…

…