

# Texturas. Mapping.

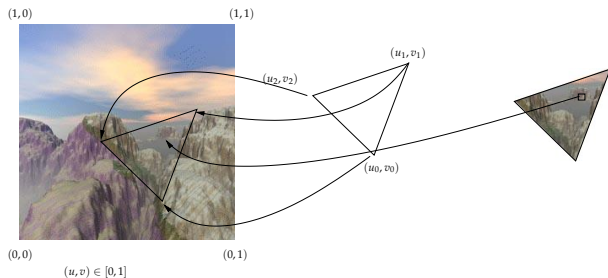
`<a.soroya@ehu.eus>`

EHU

# ¿Qué es una textura?

- Una imagen de tipo *bitmap*.
- Es decir, un array `[altura][anchura][3]`
- Los elementos de textura se denominan *texels*.
  - OpenGL llama  $(s, t)$  a estas coordenadas.
  - otros muchos las denominan  $(u, v)$ .

# Mapeo de texturas



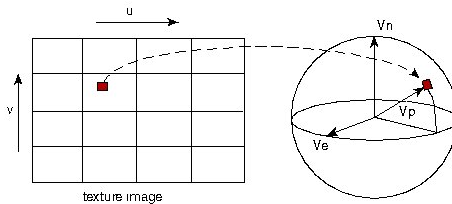
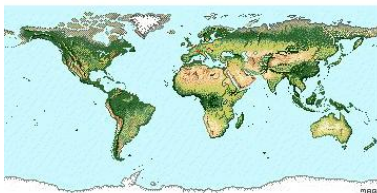
- Cada vértice tiene una coordenada de textura  $(u, v)$ 
  - Las coordenadas de texturas estan en el intervalo  $[0, 1]$
- Los *pixel* interiores se interpolan.

# Texture mapping

- Para asignar una imagen 2D en un objeto se distinguen dos fases:
  - función de *mapping*, que asigna una coordenada  $(u, v)$  a cada vértice:
    - $f(x, y, z) \rightarrow (u, v)$
  - función de *sampling*, que dado una coordenada  $(u, v)$  devuelve un color:
    - $g(u, v) \rightarrow (r, g, b, a)$

# Función mapping

- Dado un punto de la superficie, cómo saber qué textura asignarle?



# Objetos comunes

- Asignar coordenadas a los vértices de objetos comunes:
  - esfera, cubo, etc.
- De forma procedural.

```
// Las siguientes geometrias tienen coord. (u,v) asignadas  
var cube_geom = new THREE.CubeGeometry( 1 );  
var sphere_geom = THREE.SphereGeometry(1,64,64);
```

# Objetos poligonales. Superficie envolvente.

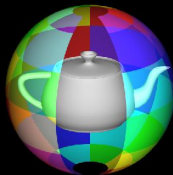
- Se considera que el objeto esta dentro de una superficie.
- Se calcula la coordenada en la superficie.
- Se proyecta al objeto

# Objetos poligonales. Superficie envolvente.

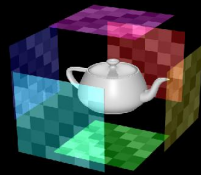
Cilindro



Esfera



Cubo





# Objetos poligonales. UV mapping.

- Se asignan las coordenadas de texturas de forma manual.
- Utilizando modeladores 3D.