

---

# Gráficos 3D

## X3DOM - WEBGL

---

*Programación Multimedia.*

*G.I.M.*

*Francisco Grimaldo, Inmaculada Coma*



---

# Índice

- Información 3D en la Web
- Tecnologías para gráficos 3D Web
- X3D
- X3DOM
- WebGL: Web Graphics Library
- Referencias

---

# Información 3D en la Web

- Interés creciente en aplicaciones web para:
  - Presentación de productos.
  - Visualización de información abstracta (p.ej. líneas temporales de datos).
  - Navegación “enriquecida” (p.ej. museos).
  
- Visualizaciones sofisticadas:
  - Ayer (¿sólo?): Uso de Adobe Flash y vídeos.
  - Mañana: Inmersión 3D a través de navegadores.

# Tecnologías para gráficos 3D Web (1/2)

- Basadas en lenguajes de marcas:

- VRML: Virtual Reality Modeling Language.
- X3D: Evolución de VRML.
- X3DOM: X3D en HTML.
- Desarrollado por Web 3D Consortium:
  - <http://www.web3d.org>

- Basadas en script:

- WebGL: OpenGL + GLSL mediante JavaScript.
- Desarrollado por Khronos Group
  - <http://www.khronos.org>

# Tecnologías para gráficos 3D Web (2/2)

	2D (Final HTML5 spec)	3D (No W3C spec yet)
<b>Declarative</b> Scenegraph Part of HTML-document DOM Integration CSS/ Events		
<b>Imperative</b> Procedural API Drawing context		

---

# X3D: Introducción

- X3D es un formato de fichero estándar abierto basado en XML que permite la comunicación en tiempo real de datos 3D entre aplicaciones y aplicaciones de red.
- Ofrece una amplia gama de características para el uso en ingeniería y visualización científica, CAD y arquitectura, visualización médica, entrenamiento y simulación, multimedia, entretenimiento, educación,...
- Heredero de VRML97.

---

# X3D: Características

- Integración XML:
  - Servicios Web, multiplataforma, transferencia de datos entre aplicaciones.
- Compartimentado:
  - Distribuciones, perfiles y extensiones.
- Visualización en múltiples dispositivos.
- Gráficos, audio y vídeo interactivos en tiempo real.
- Estándar ISO bien especificado

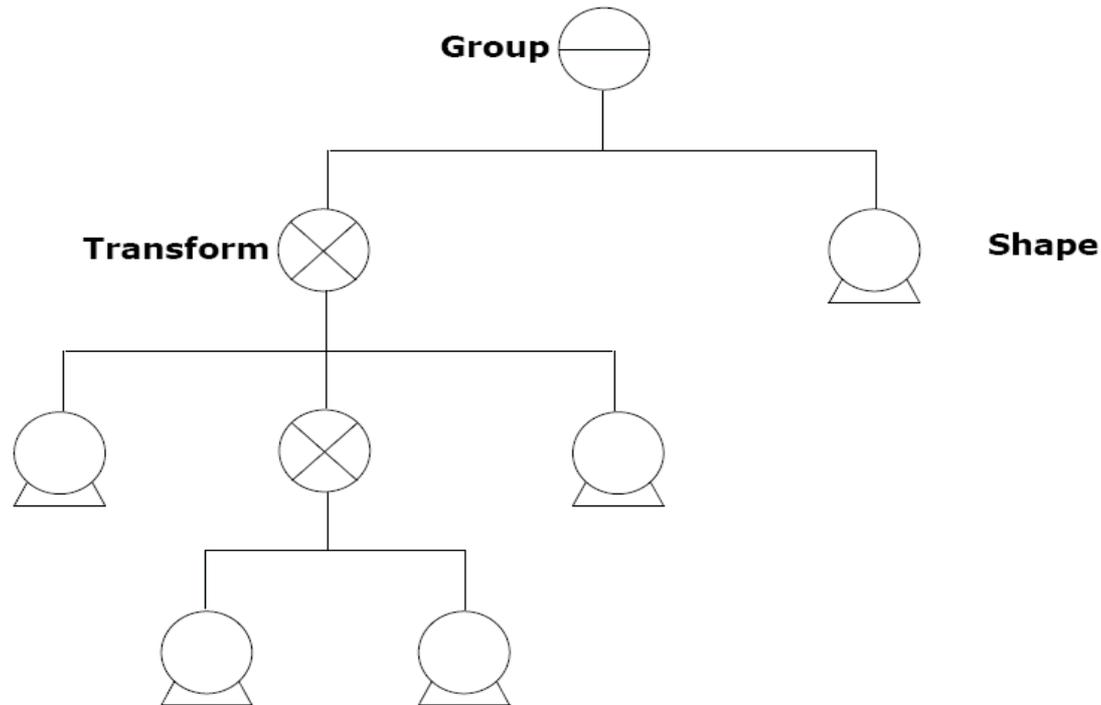
# X3D: Funcionalidades

- **Objetos definidos por el usuario:** Posibilidad de extender la funcionalidad del navegador para definir tipos de datos definidos por el usuario.
- **Scripting:** Capacidad de cambio dinámico de la escena utilizando lenguajes de programación y de script.
- **Red:** Capacidad de composición de una escena X3D a partir de “assets” ubicados en una red; hiperenlace de objetos a otras escenas o “assets” alojados en cualquier lugar de la WWW.
- **Simulación física:** Animación humanoide, conjuntos de datos geoespaciales, integración con protocolos DIS (Distributive Interactive Simulation).

# X3D: Grafo de escena (1/2)

- Grafo acíclico que describe el mundo 3D creado (objetos 3D, interactividad,...).
- Cada nodo del grafo es una instancia de uno de los tipos de nodos disponibles:
  - Un nodo Shape define una forma; uno Transform posiciona, orienta y escala sus nodos hijos; y un nodo DirectionalLight define una fuente de luz.
- Respecto a VRML97, en X3D aparecen nuevos nodos que incorporan avances tridimensionales aprovechando los nuevos recursos hardware y software.

# X3D: Grafo de escena (2/2)

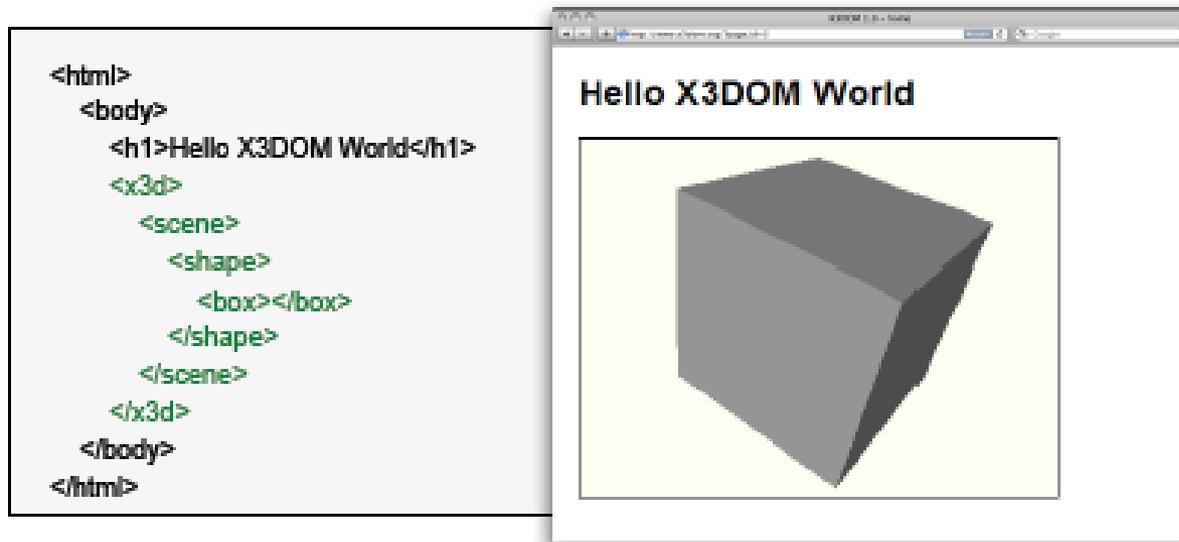


## ■ Ejemplos básicos:

- <http://www.web3d.org/x3d/content/examples/Basic/>

# X3DOM: X3D en HTML5

- X3DOM = X3D + DOM
  - Integra los gráficos 3D definidos de forma declarativa en HTML5.
  - Permite el uso de JavaScript y el acceso al DOM para contenido 3D



# X3DOM: Ventajas (1/2)

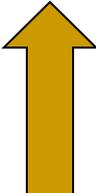
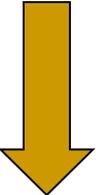
- Integración nativa con el navegador:
  - No necesita la instalación de *plugins* o aplicaciones.
  - Evita problemas de instalación, permisos y seguridad
  - Independiente del sistema operativo, especialmente en dispositivos móviles
- Facilita el desarrollo:
  - Entorno de ejecución completo en el navegador.
  - Ubicuidad de navegadores incrementa accesibilidad.
  - Carece de un API especial y avanzada.
- Integración con otras técnicas Web estándar:
  - CSS, Ajax

# X3DOM: Ventajas (2/2)

- Descripción de contenido declarativo y legible:
  - Encapsula detalles gráficos de bajo nivel.
  - Interacción con el usuario mediante APIs Web estándar (p.ej. JavaScript y CSS).
  - Interoperabilidad: El mismo código se ejecuta en escritorio, red y dispositivo móvil.
- Integración HTML vs Sistema cerrado:
  - Metadatos. Facilita la indexación y búsqueda.
  - Permite la combinación de contenidos de diversas fuentes (*mashups*).
  - Conexión de datos existentes (p.ej. geo-información) con contenido 3D.

# X3DOM: Plantilla básica (1/2)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Plantilla X3DOM</title>
    <link rel="stylesheet" type="text/css"
          href="http://www.x3dom.org/download/dev/x3dom.css">
    <script type="text/javascript"
            src="http://www.x3dom.org/download/dev/x3dom.js">
    </script>
  </head>
  ...
```



# X3DOM: Plantilla básica (2/2)

...

```
<body>
```

```
<h1>Plantilla X3DOM</h1>
```

```
<x3d width="400" height="300">
```

```
<scene>
```

```
<shape>
```

```
<appearance>
```

```
<material
```

```
diffuseColor='red'>
```

```
</material>
```

```
</appearance>
```

```
<box></box>
```

```
</shape>
```

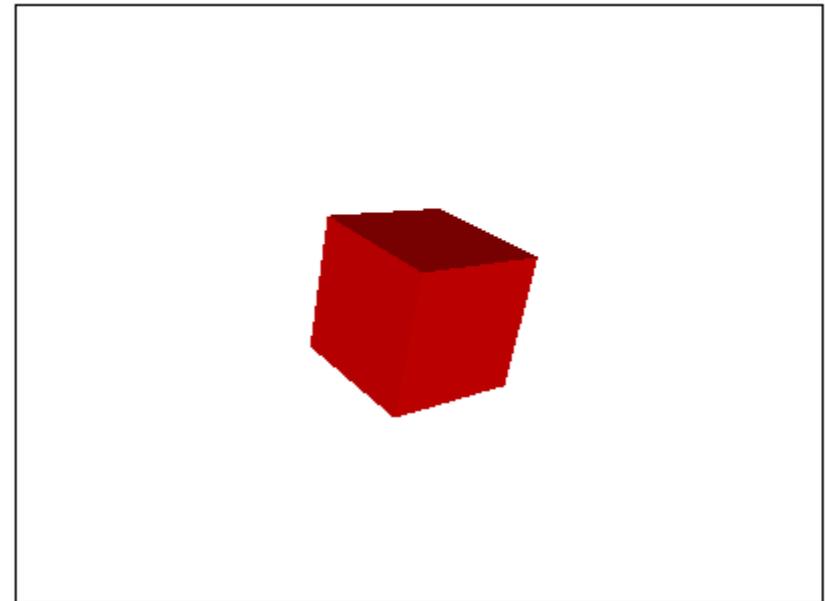
```
</scene>
```

```
</x3d>
```

```
</body>
```

```
</html>
```

Plantilla X3DOM



# X3DOM: Background

## ■ Nodo X3D:

```
<scene>
```

```
  <shape>...</shape>
```

```
  <background skyColor="0 0 0">
```

```
  </background>
```

```
</scene>
```

## ■ Estilo CSS:

```
<x3d style="background-color:#00F">
```

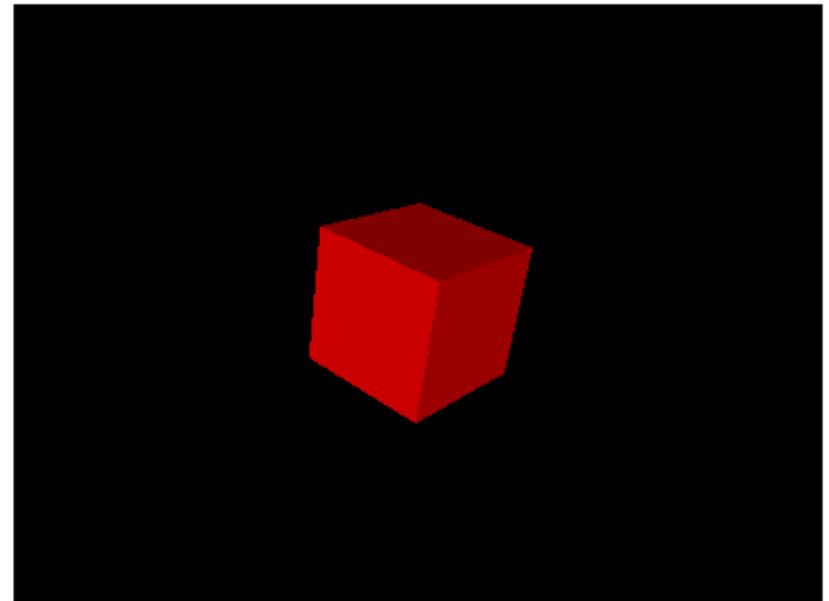
```
  <scene>
```

```
    ...
```

```
  </scene>
```

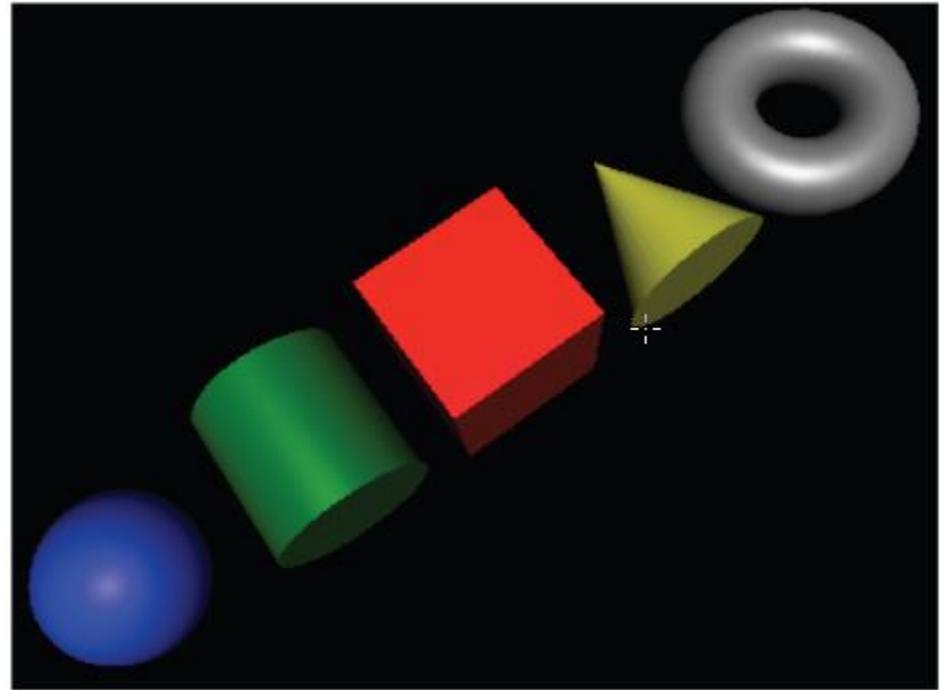
```
</x3d>
```

Plantilla X3DOM



# X3DOM: Formas geométricas

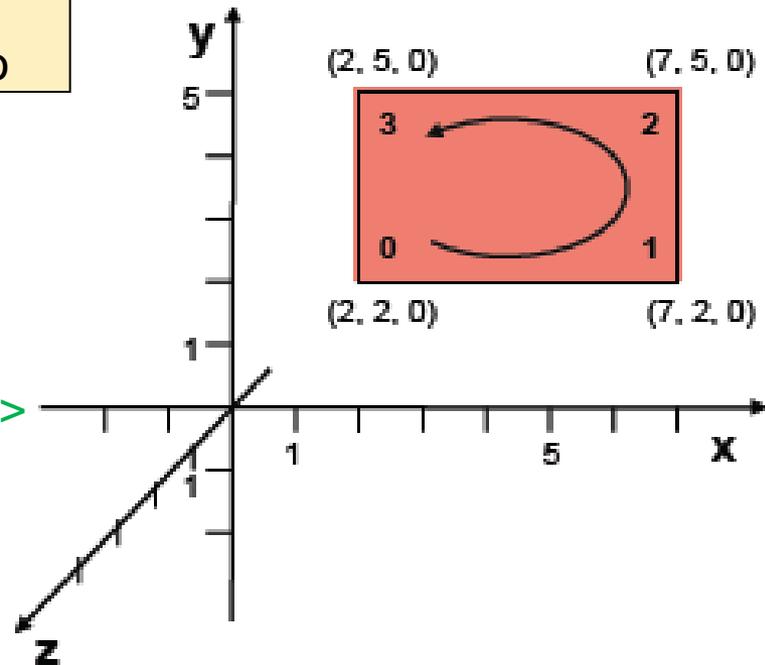
- `<sphere radius="1.0">`
- `<cylinder radius="1.0" height="2.0">`
- `<box size="2.0 2.0 2.0">`
- `<cone bottomRadius="1.0" height="2.0">`
- `<torus innerRadius="0.5" outerRadius="1.0">`



# X3DOM: Creación de formas

```
<scene>
  <shape>
    <appearance>
      <material diffuseColor="salmon">
      </material>
    </appearance>
    <indexedFaceSet coordIndex="0 1 2 3 -1">
      <coordinate point="2 2 0, 7 2 0, 7 5 0, 2 5 0">
      </coordinate>
    </indexedFaceSet>
  </shape>
  <viewpoint position="0 0 15"></viewpoint>
</scene>
```

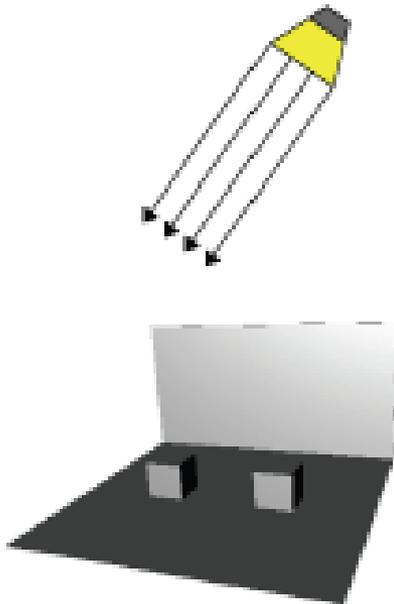
Marcador de final del polígono



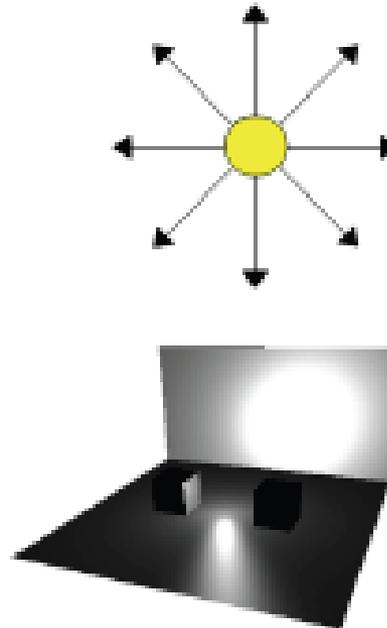
Sobre el eje Z y mirando hacia abajo

# X3DOM: Fuentes de luz

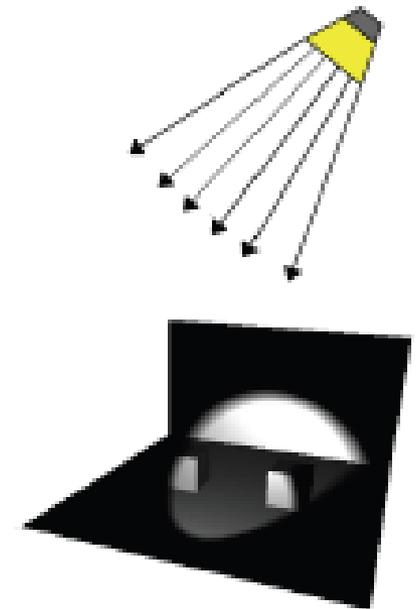
Directional light



Point light



Spot light



```
<directionalLight direction='0 0 -1' intensity='1'> </directionalLight >
```

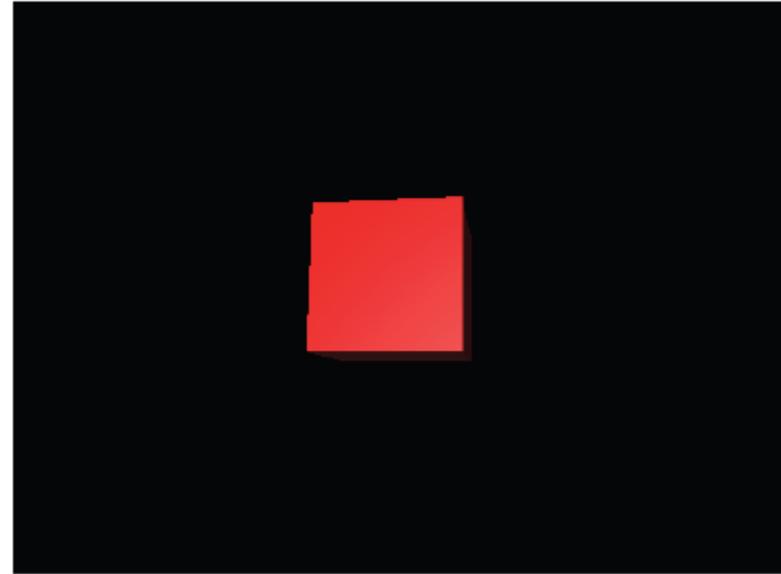
```
<pointLight location='0 0 0' intensity='1'> </pointLight >
```

```
<spotLight direction='0 0 -1' location='0 0 0' intensity='1'> </spotLight >
```

# X3DOM: Iluminación de los objetos

...

```
<body>  
  <x3d width="400" height="300">  
    <scene>  
      <shape>  
        <appearance>  
          <material  
            diffuseColor='red'  
            specularColor="#808080">  
          </material>  
        </appearance>  
        <box></box>  
      </shape>  
    </scene>  
  </x3d>  
</body>
```



# X3DOM: Texturas

## ■ Imágenes

```
<x3d width="500px" height="400px">  
  <scene>  
    <shape>  
      <appearance>  
        <imageTexture url="logo.png"></imageTexture>  
      </appearance>  
    <box></box>  
  </shape>  
</scene>  
</x3d>
```

## ■ Vídeo:

```
<movieTexture url="foo.mp4", "foo.ogv"></movieTexture>
```

# X3DOM: Traslaciones

```
<transform translation="-2 0 0">
```

```
<shape>
```

```
<appearance>
```

```
<material diffuseColor="red"></material>
```

```
</appearance>
```

```
<box></box>
```

```
</shape>
```

```
</transform>
```

```
<transform translation="2 0 0">
```

```
<shape>
```

```
<appearance>
```

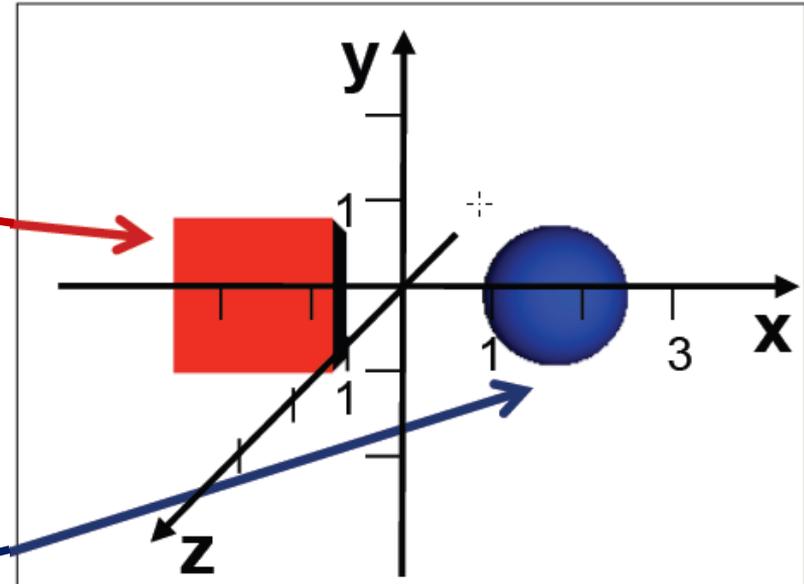
```
<material diffuseColor="blue"></material>
```

```
</appearance>
```

```
<sphere></sphere>
```

```
</shape>
```

```
</transform>
```



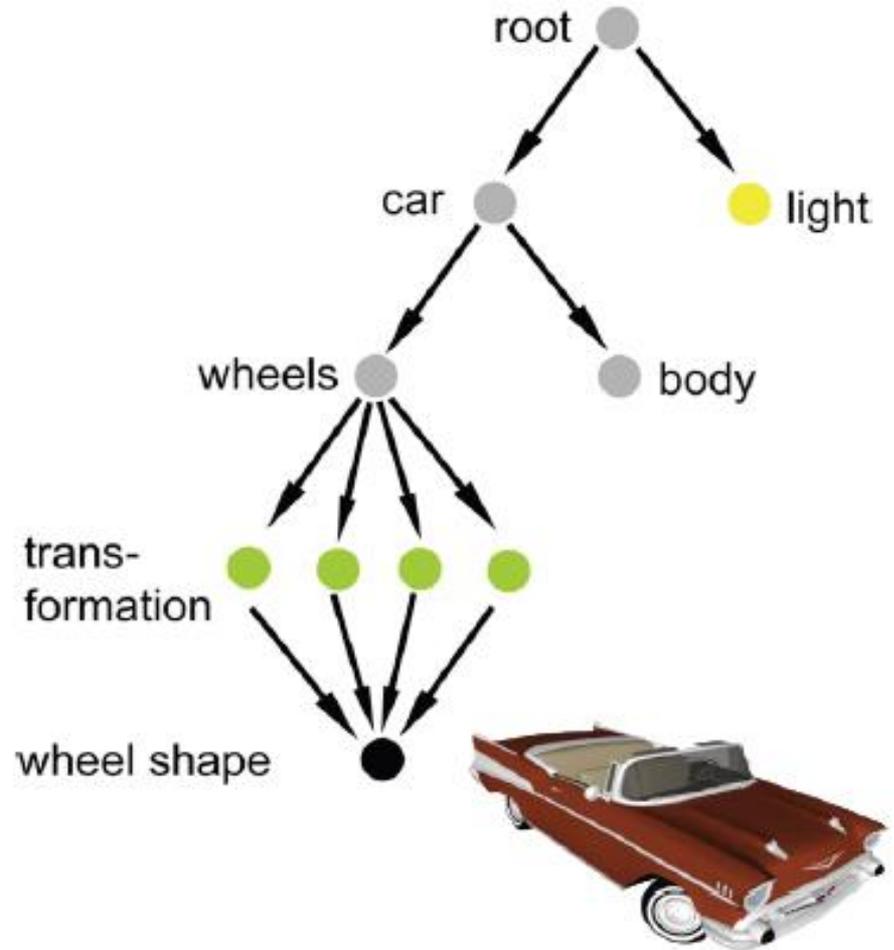
# X3DOM: Grupos y transformaciones

- Los nodos `<group>` y `<transform>`, respectivamente, agrupan y posicionan uno o más objetos del grafo de escena:

```
<transform translation="0 0 0"  
              rotation="0 1 0 0"  
              scale="1 1 1">
```

...

```
</transform>
```



# X3DOM: Acceso al DOM

## ■ **Nombrar nodos:**

- ❑ `<group id='root'></group>`
- ❑ También se usa mucho el atributo *ref*.

## ■ **Añadir nodos:**

- ❑ `root = document.getElementById('root');`
- ❑ `trans = document.createElement('Transform');`
- ❑ `trans.setAttribute('translation', '1 2 3');`
- ❑ `root.appendChild(trans);`

## ■ **Eliminar nodos:**

- ❑ `root.removeChild(trans);`

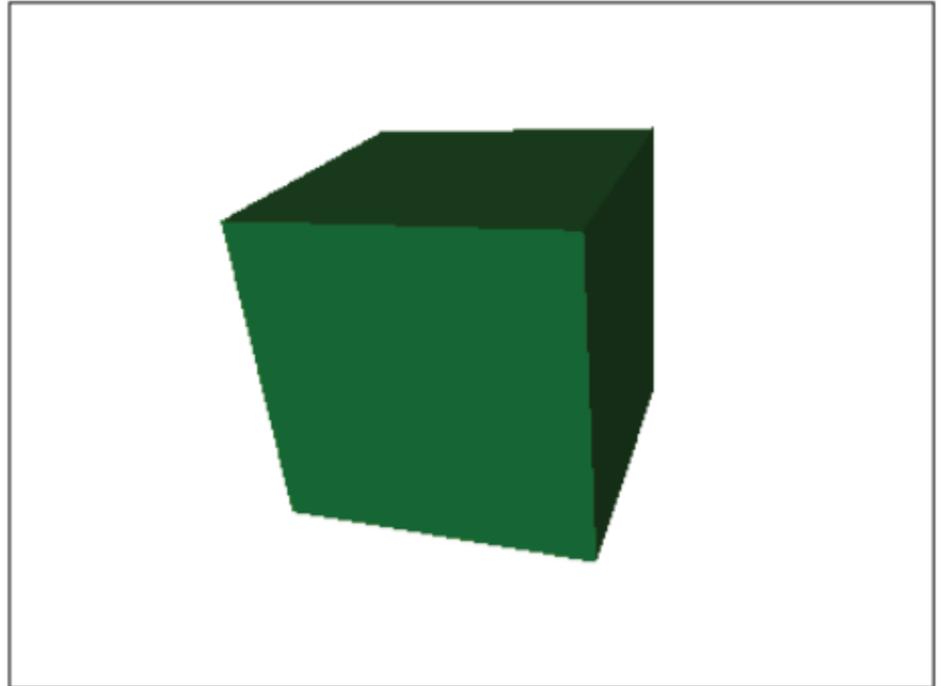
## ■ **Modificar atributos:**

- ❑ `document.getElementById('mat').setAttribute('diffuseColor', 'red');`

# X3DOM: Interacción con usuario (1/2)

## ■ Alternativa A:

```
...  
<shape>  
  <appearance>  
    <material id="mat"  
              diffuseColor="red">  
    </material>  
  </appearance>  
  <box onclick="  
        document.getElementById('mat').setAttribute('diffuseColor', 'green');" >  
  </box>  
</shape>  
...
```



# X3DOM: Interacción con usuario (2/2)

## ■ Alternativa B:

```
<shape>
```

```
  <appearance>
```

```
    <material id="mat" diffuseColor="red"></material>
```

```
  </appearance>
```

```
  <box id="box"></box>
```

```
</shape>
```

```
<script type="text/javascript">
```

```
  document.onload = function() {
```

```
    document.getElementById('box').addEventListener('click', function() {
```

```
      document.getElementById('mat').setAttribute('diffuseColor', 'olive');
```

```
    }, false) };
```

```
</script>
```

# X3DOM: Animaciones

```
<scene>
  <transform id="trafo"
    rotation="0 1 0 0">
    <shape>
      <appearance>
        <material diffuseColor="red">
          </material>
        </appearance>
      <box></box>
    </shape>
  </transform>
```

```
<timeSensor id="ts" loop="true" cycleInterval="2">
</timeSensor>
```

```
<orientationInterpolator id="oi" key="0.0 0.5 1.0"
  keyValue="0 1 0 0, 0 1 0 3.14, 0 1 0 6.28">
</orientationInterpolator>
```

```
<ROUTE fromNode='ts' fromField='fraction_changed'
  toNode='oi' toField='set_fraction'></ROUTE>
<ROUTE fromNode='oi' fromField='value_changed'
  toNode='trafo' toField='set_rotation'></ROUTE>
</scene>
```

- El primer <route> hace que el nodo <timeSensor> “ts” active al nodo <orientationInterpolator> “oi”, que proporciona los ángulos de rotación alrededor del eje Y (0 1 0).
- El segundo <route> hace que el valor resultante se use para el atributo *rotation* del nodo <transform> “trafo”.

---

# Referencias X3DOM

- **Web 3D Consortium:** Información general X3D.
  - <http://www.web3d.org>
- **X3DOM:** Sitio oficial, documentación y ejemplos.
  - <http://www.x3dom.org>
  - <http://x3dom.org/docs/dev/index.html>
  - [http://www.x3dom.org/?page\\_id=5](http://www.x3dom.org/?page_id=5)
  - <http://x3dom.org/x3dom/test/functional/>

---

# WebGL: Web Graphics Library

- Interfaz de programación de gráficos 3D dentro de páginas web.
- Origen: 2006, desarrollo de su propio Canvas3D por Mozilla.
- Actualmente desarrollado por Khronos Group del cual forman parte tanto Mozilla, Opera como Apple y Google.
- Especificación WebGL 1.0 (Feb. 2011)
  - <https://www.khronos.org/registry/webgl/specs/1.0/>

# WebGL: Introducción

- Herramienta multiplataforma para dibujar contenido 3D en HTML que se basa en:
  - OpenGL ES 2.0.
  - OpenGL Shading Language (GLSL ES).
- JavaScript para acceder al DOM a bajo nivel.
- Utiliza el elemento *canvas* de HTML5:
  - `gl = canvas.getContext('webgl');`
- Soportado: Firefox, Chrome, Opera y Safari.
  - <http://www.doesmybrowsersupportwebgl.com>
- Con IE Explorer necesita un plug-in: IEWebGL.

---

# WebGL: Modelos

- Se pueden importar modelos 3D creados con programas con Blender y exportados o descargar modelos creados:
  - <https://sketchfab.com/>
- Se puede trabajar a bajo nivel con el API de programación de WebGL o utilizar algunas librerías de alto nivel que encapsulan esta API.

---

# WebGL: Librerías de programación

- **WebGLU:** la primera librería WebGL disponible públicamente.
- **C3DL:** Canvas 3D JavaScript Library.
- Librerías alto nivel: **Three.js, GLGE, SceneJS, Oak3D**
- **Processing.js:** utiliza WebGL para renderizar un objeto 2D o 3D, programado en Processing, dentro de un canvas de HTML.
- **GLOW:** Envoltente (wrapper) de WebGL desarrollado por la BBC.
- **SpiderGL:** librería gráfica JavaScript 3D basada en WebGL.
- **gwt-g3d:** librería 3D para GWT (Google Web Toolkit).

# Referencias WebGL

- **WebGL en Wikipedia:** Introducción al lenguaje. Breve reseña histórica
  - <http://es.wikipedia.org/wiki/WebGL>
  - <http://en.wikipedia.org/wiki/WebGL>
- **.Lenguaje WebGL:** Sitio oficial. Especificaciones completas, documentación del API, recursos de programación, foros y demás.
  - <http://www.khronos.org/webgl/>
- **Tutorial WebGL de Mozilla Developer Network:** Tutorial de Mozilla sobre WebGL, centrado en el navegador Firefox.
  - <https://developer.mozilla.org/en/WebGL>
- **Learning WebGL:** Blog con lecciones incrementales, ejercicios y ejemplos (original en inglés y traducción al español).
  - <http://learningwebgl.com/blog/>
  - <http://www.jlabstudio.com/webgl/tutoriales-webgl/>
- **Tutorial de WebGL. Universidad de La Coruña:** Ejemplos de uso de librerías de desarrollo para WebGL (WebGLU y C3DL).
  - <http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Tutoriales/TutorialWebGL/index.htm>