

Programación de Dispositivos Móviles



Sesión 13: Gráficos 3D

Índice



- **Introducción**
- **Modo inmediato**
- **Modo retained**

Gráficos 3D



- **Introducción**
- **Modo inmediato**
- **Modo retained**

Mobile 3D Graphics



- **La API *Mobile 3D Graphics* nos permite crear gráficos 3D en los dispositivos móviles**
- **Soporta dos modos:**
 - **Modo inmediato**
 - Se crean gráficos a bajo nivel
 - Se especifica los vértices, caras y apariencia de los objetos
 - Adecuado para representar datos en 3D
 - **Modo *retained***
 - Se crea un grafo con los distintos objetos de la escena 3D
 - Los objetos 3D se cargan de un fichero M3G
 - Adecuado para juegos



Gráficos 3D



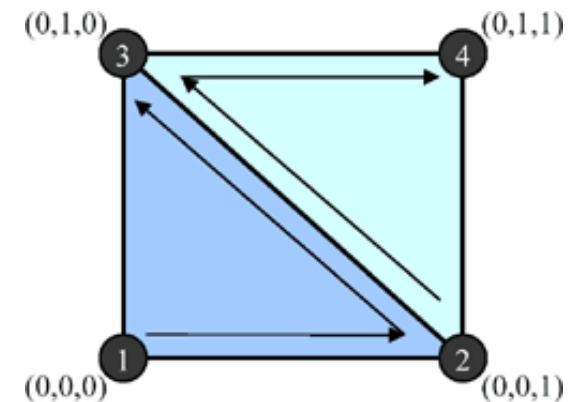
- **Introducción**
- **Modo inmediato**
- **Modo retained**

Modo inmediato



■ Definimos vértices y caras de los objetos

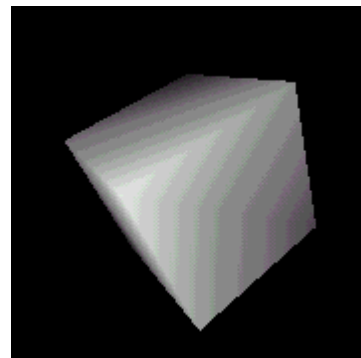
```
short [] vertexValues = {  
    0, 0, 0, // 0  
    0, 0, 1, // 1  
    0, 1, 0, // 2  
    0, 1, 1, // 3  
    1, 0, 0, // 4  
    1, 0, 1, // 5  
    1, 1, 0, // 6  
    1, 1, 1 // 7 };  
  
int [] faceIndices = {  
    0, 1, 2, 3,  
    7, 5, 6, 4,  
    4, 5, 0, 1,  
    3, 7, 2, 6,  
    0, 2, 4, 6,  
    1, 5, 3, 7 };
```



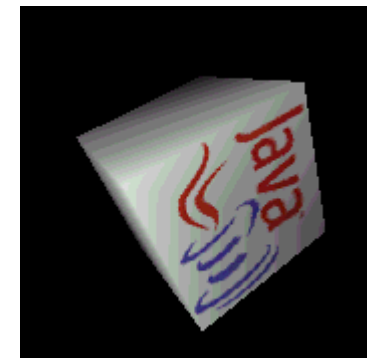
Cara { 0, 1, 2, 3 }



Sin material



Con material



Con textura

Gráficos 3D

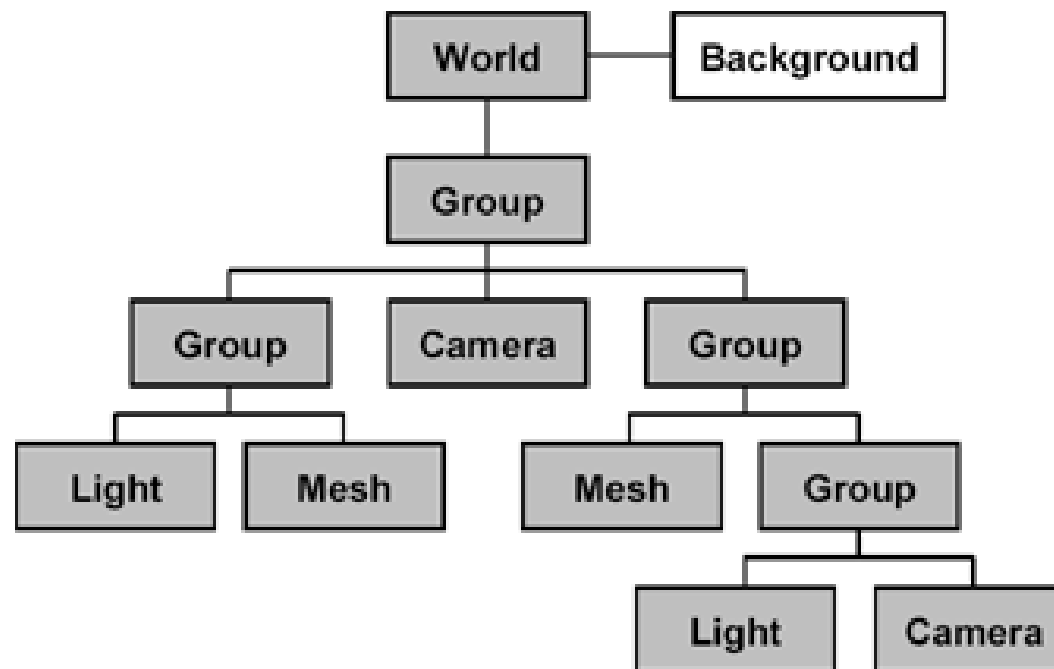


- **Introducción**
- **Modo inmediato**
- **Modo retained**

Modo retained



- Se construye un grafo de la escena
 - Contiene todos los objetos en distintos grupos

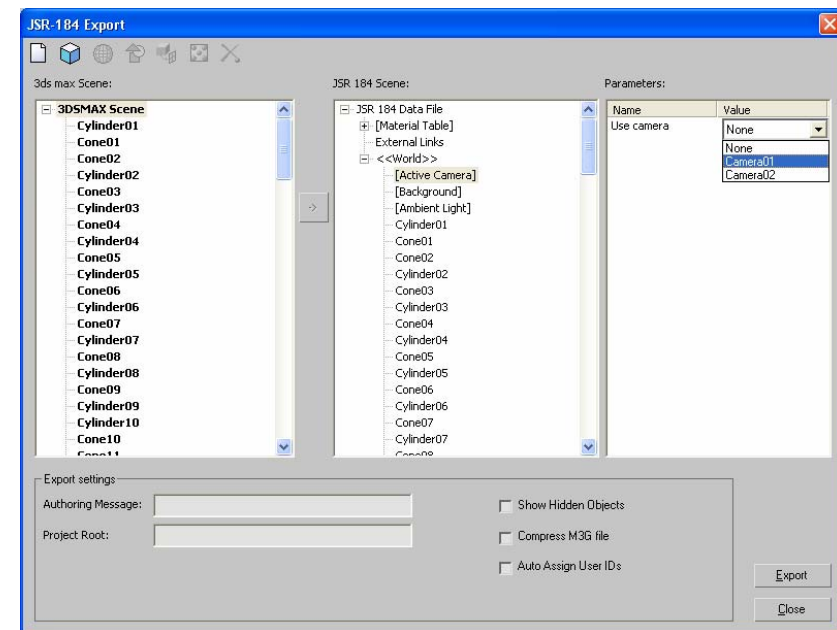
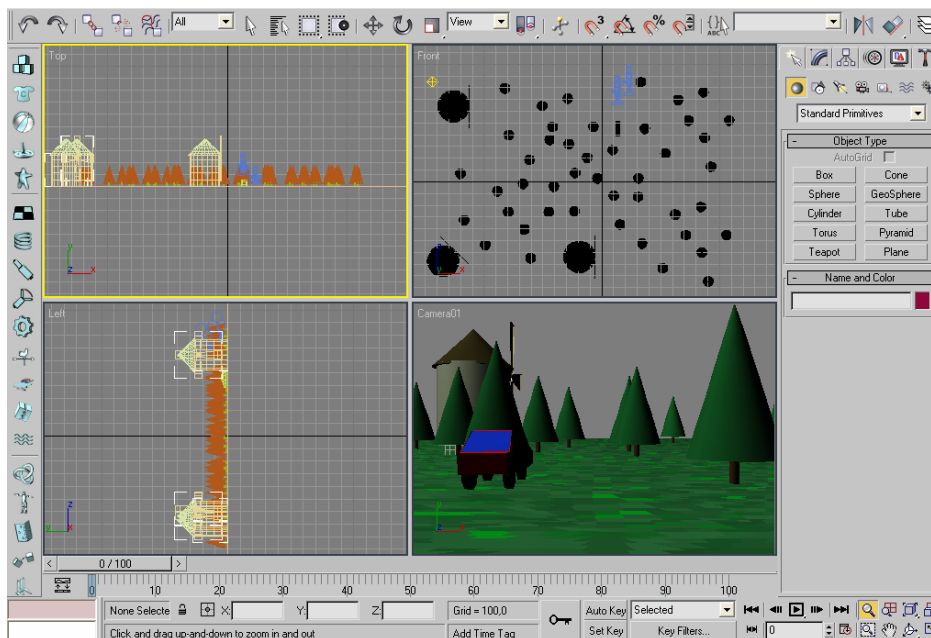


- Cargamos este grafo de un fichero M3G

Modelado



- Podemos modelar los gráficos 3D con herramientas como 3D Studio MAX
 - A partir de 3DSMAX 7.0 se incluye una herramienta para exportar a ficheros M3G



Ejemplo de modo *retained*



```
public class Visor3DRetained extends Canvas {

    Graphics3D g3d;
    World mundo;

    public Visor3DRetained() {
        g3d = Graphics3D.getInstance();
        try {
            mundo = (World)Loader.load("/mundo.m3g")[0];
        } catch (IOException e) { // Error al cargar mundo }
    }
    protected void paint(Graphics g) {
        try {
            g3d.bindTarget(g);
            g3d.render(mundo);
        } finally {
            g3d.releaseTarget();
        }
    }
}
```