

# Formatos de mallas

## Informática Gráfica I

Material de: **Antonio Gavilanes**  
Adaptado por: **Elena Gómez y Rubén Rubio**  
`{mariaelena.gomez,rubenrub}@ucm.es`



# Contenido

## 1 Tipos de formatos

- Formato PLY

## 2 Ejemplos

# Tipos de formatos

Los objetos se pueden almacenar de dos formas:

- Una definición a alto nivel de abstracción del objeto.
  - Por ejemplo: esfera de radio y centro dados (más posiblemente el número de meridianos y paralelos).
  - La descripción es compacta y fácilmente entendible, pero el programa tiene que saber dibujar muchos tipos distintos de objetos.
- Una definición detallada de su superficie (mallas poligonales).
  - Como los atributos de la clase [Mesh](#) de las prácticas.
  - La descripción es larga y difícil de procesar manualmente, pero el código que dibuja es genérico y no está limitado a un repertorio de objetos.

# Formato basados en superficie del objeto

Formatos del segundo tipo son:

- .obj → Wavefront Technologies
- .stl (**STereoLithography**) → 3D Systems
- .ply (**Polygon File Format**) → Universidad de Stanford
  - Es un formato basado en triángulos.
  - Es una forma estándar para guardar información tridimensional de objetos, generada por escáneres.
  - Guarda información de vértices y caras.
- .glTF (**GL Transmission Format**) → Khronos Group
  - Basado en JSON, soporta grafos de la escena y animación.

Todos estos formatos se pueden importar y exportar en Blender.

# Formato PLY

- El contenido de un archivo .ply es generalmente el siguiente:
  - Identificación
  - Cabecera
  - Datos
- La identificación es la palabra ply.

# Formato PLY

- La cabecera suele tener este aspecto:

```
1  format ascii 1.0
2  element vertex 4
3  property float x
4  property float y
5  property float z
6  element face 4
7  property list uchar int vertex_indices
8  end_header
```

- La primera línea especifica el formato de los datos que puede ser ASCII (texto) o binario.
- La segunda línea especifica el número de vértices del objeto.
- Las siguientes tres líneas especifican el formato de los vértices (en este ejemplo indica que serán tres floats).

# Formato PLY

- La cabecera suele tener este aspecto:

```
1 format ascii 1.0
2 element vertex 4
3 property float x
4 property float y
5 property float z
6 element face 4
7 property list uchar int vertex_indices
8 end_header
```

- La siguiente línea especifica el número de caras.
- Sigue otra línea con el formato de las caras. En este ejemplo se dice que será una lista de caracteres de índices de vértices.
- La cabecera se cierra con una línea `end_header`.
- A partir de la cabecera empieza la información de los vértices y caras del objeto.

# Formato PLY

- Datos del tetraedro que aparecen en el archivo .ply:

0	0	0	
1	0	0	
0	1	0	
0	0	1	
3	1	2	3
3	0	2	1
3	0	3	2
3	1	3	0



# Formato PLY

- Hay un repositorio de objetos guardados en formato ply en la dirección `graphics.stanford.edu/data/3Dscanrep`
- El formato PLY es sencillo de parsear en C++
- La biblioteca assimp (Open Asset Importer Library) también se puede utilizar para importar este y otros formatos más complejos (más de 40 distintos)

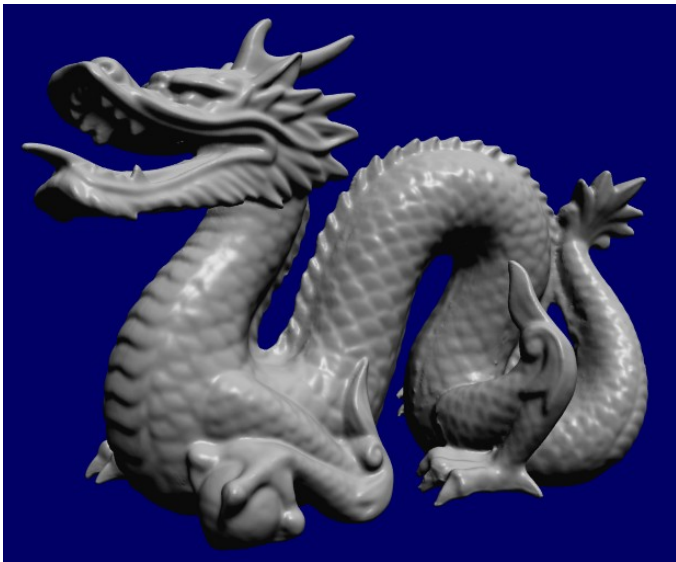
# Ejemplos de imágenes



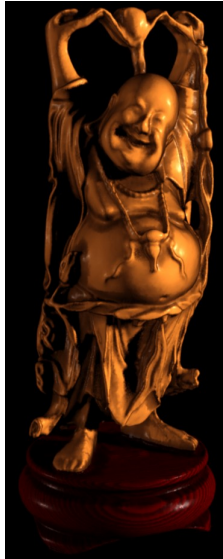
# Ejemplos de imágenes



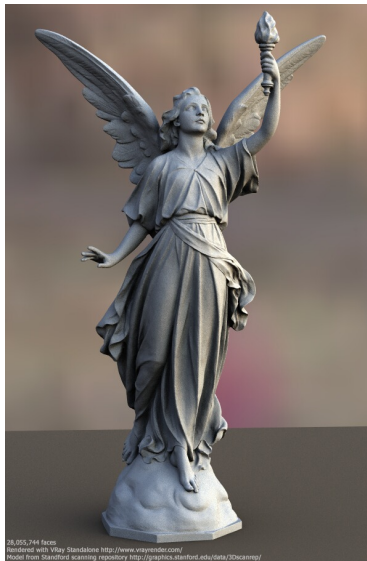
# Ejemplos de imágenes



# Ejemplos de imágenes



# Ejemplos de imágenes



# Ejemplos de imágenes

