
Práctica 2

Modelos PLY y Poligonales

2.1. Objetivos

Aprender a:

- A leer modelos guardados en ficheros externos en formato PLY (Polygon File Format) y su visualización.
- Modelar objetos sólidos poligonales mediante técnicas sencillas. En este caso se usará la técnica de modelado por revolución de un perfil alrededor de un eje de rotación.

2.2. Desarrollo

PLY es un formato para almacenar modelos gráficos mediante listas de vértices, caras poligonales y diversas propiedades (colores, normales, etc.) que fue desarrollado por Greg Turk en la universidad de Stanford durante los años 90. Para más información consultar:

<http://www.dcs.ed.ac.uk/teaching/cs4/www/graphics/Web/ply.html>

Para la realización de la práctica, en primer lugar, se visualizarán modelos de objetos guardados en formato PLY usando los modos de visualización implementados en la primera práctica. Para ello, se entregará el código de un lector básico de ficheros PLY para objetos únicamente compuestos por vértices y caras triangulares, que devuelve un vector de coordenadas de los vértices y un vector de los índices de vértices que forman cada cara. Se creará una estructura de datos que guarde los vectores anteriores (se recomienda usar STL y el fichero auxiliar vertex.h).

En segundo lugar, se ha de crear un código que a partir del conjunto de puntos que representen un perfil respecto a un plano principal ($X = 0$ ó $Y = 0$ ó $Z = 0$), de un parámetro que indique el número de lados longitudinales del objeto a generar por revolución y de un eje de rotación, calcule el conjunto de vértices y el conjunto de caras que representan el sólido obtenido.

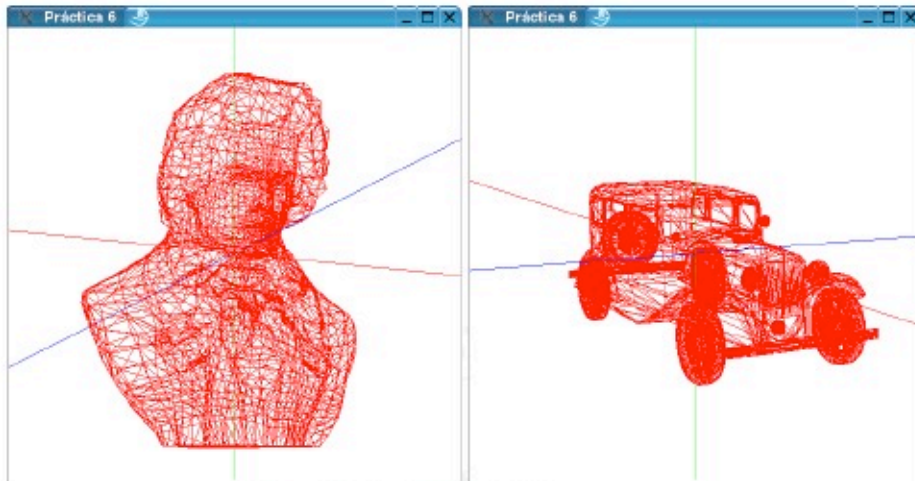


Figura 2.1: Objetos PLY.



(a) Puntos

(b) Polilínea

Figura 2.2: Perfil inicial.

Los pasos para crear un sólido por revolución serían:

- Sea, por ejemplo, un perfil inicial Q_1 en el plano $Z = 0$ definido como:

$$Q_1 (p_1 (x_1, y_1, 0), \dots, p_M (x_M, y_M, 0)) ,$$

siendo $p_i (x_i, y_i, 0)$ con $i = 1, \dots, M$ los puntos que definen el perfil (ver figura 2.2).

- Se toma como eje de rotación el eje Y y si N es número lados longitudinales, se obtienen los puntos o vértices del sólido poligonal a construir multiplicando Q_1 por N sucesivas transformaciones de rotación con respecto al eje Y , a las que notamos por

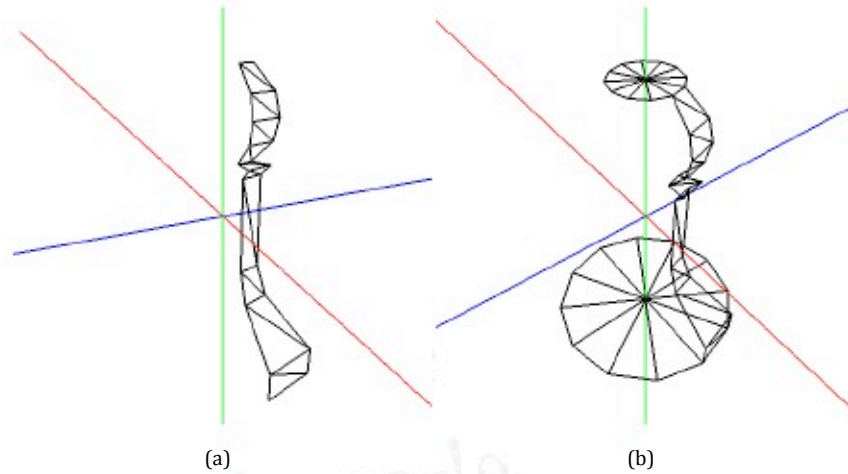


Figura 2.3: Caras del sólido a construir: (a) longitudinales (solo un lado es mostrado) y (b) incluyendo las tapas superior e inferior.

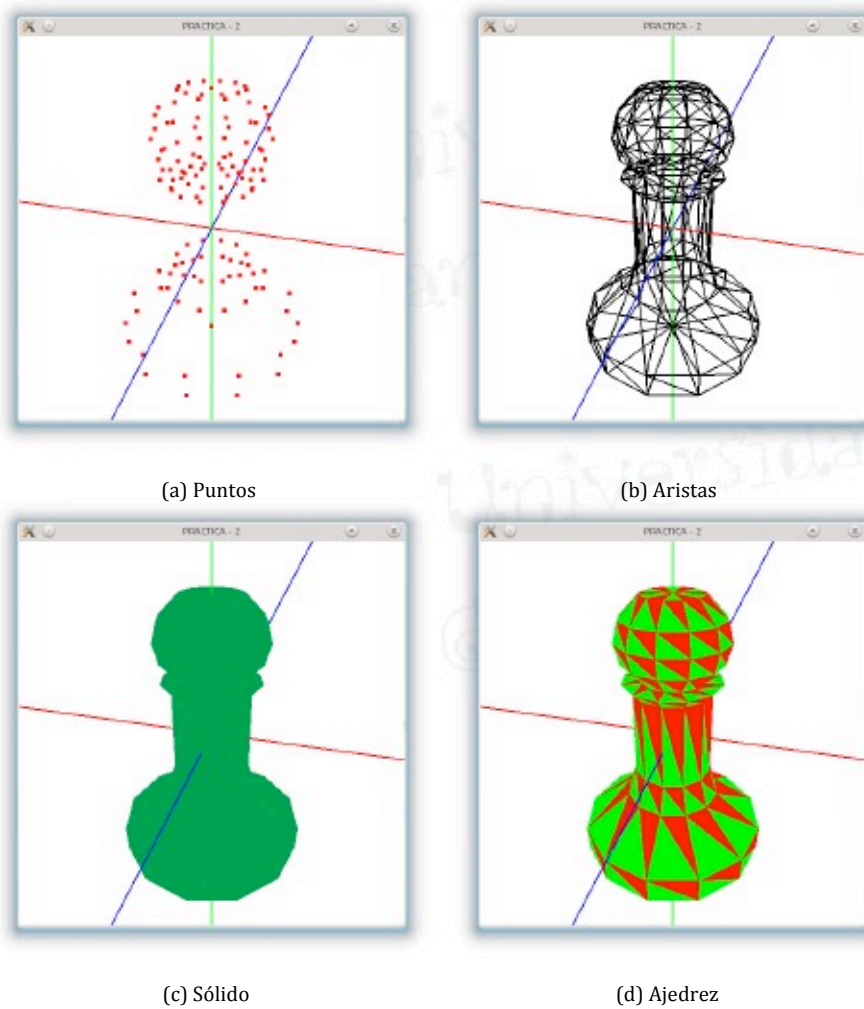


Figura 2.4: Sólido generado por revolución con distintos modos de visualización.

$R_Y(\alpha_j)$ siendo α_j los valores de N ángulos de rotación equiespaciados. Se obtiene un conjunto de $N \times M$ vértices agrupados en N perfiles Q_j , siendo:

$$Q_j = Q_1 R_Y(\alpha_j), \text{ con } j = 1, \dots, N$$

- Se guardan $N \times M$ los vértices obtenidos en un vector de vértices según la estructura de datos creada en la práctica anterior.
- Las caras longitudinales del sólido (triángulos) se crean a partir de los vértices de dos perfiles consecutivos Q_j y Q_{j+1} . Tomando dos puntos adyacentes en cada uno de los dos perfiles Q_j y Q_{j+1} y estando dichos puntos a la misma altura, se pueden crear dos triángulos. En la figura 2.3(a) se muestran los triángulos así obtenidos solamente para un lado longitudinal para una mejor visualización. Los vértices de los triángulos tienen que estar ordenados en el sentido contrario a las agujas del reloj.
- A continuación creamos las tapas del sólido tanto inferior como superior (ver figura 2.3(b)). Para ello se han de añadir dos puntos al vector de vértices que se obtienen por la proyección sobre el eje de rotación del primer y último punto del perfil inicial. Estos dos vértices serán compartidos por todas las caras de las tapas superior e inferior.
- Todas las caras, tanto las longitudinales como las tapas superior e inferior, se almacenan en la estructura de datos creada para las caras en la práctica anterior.

El modelo poligonal finalmente obtenido también se podrá visualizar usando cualquiera de los distintos modos de visualización implementados para la primera práctica (ver figura 2.4).

Dos consideraciones sobre la implementación del código para el objeto por revolución: primera, se puede hacer un tratamiento diferenciado cuando uno o ambos puntos extremos del perfil inicial están situados sobre el eje de rotación y segunda, el perfil inicial se puede leer de un fichero PLY cuyo contenido sólo ha de tener las coordenadas de los puntos de éste (no es difícil crear manualmente un perfil con un fichero PLY, véase el siguiente ejemplo, donde hay 11 vértices y una sola cara que no se utilizará)

```
ply
format ascii 1.0
element vertex 11
property float32 x
property float32 y
property float32 z
element face 1
property list uchar uint vertex_indices
end_header
1.0 -1.4 0.0
1.0 -1.1 0.0
0.5 -0.7 0.0
0.4 -0.4 0.0
0.4 0.5 0.0
```

0.5 0.6 0.0
0.3 0.6 0.0
0.5 0.8 0.0
0.55 1.0 0.0
0.5 1.2 0.0
0.3 1.4 0.0
3 0 1 2

2.3. Evaluación

La evaluación de la práctica, sobre 10 puntos, se hará del modo siguiente:

- Lectura y visualización de ficheros PLY. y en modo puntos (3 pts.).
- Creación del código para el modelado de objetos por revolución (7 pts.).

2.4. Duración

La práctica se desarrollará en 3 sesiones

2.5. Bibliografía

- Mark Segal y Kurt Akeley; The OpenGL Graphics System: A Specification (version 4.1); <http://www.opengl.org/>
- P. Shirley y S. Marschner; Fundamentals of Computer Graphics, 3rd Edition; A K Peters Ltd. 2009.
- J. Vince; Mathematics for Computer Graphics; Springer 2006.