

Manual de Informática Gráfica

Tema 1

El proceso de visualización produce una imagen a partir de un modelo de escena y unos parámetros:

Modelo de escena: estructura de datos en memoria que representa lo que se quiere ver, esta formado por varias partes:

- Modelo geométrico: conjunto de primitivas (típicamente polígonos planos), que definen la forma de los objetos a visualizar.
- Modelo de aspecto: conjunto de parámetros que definen el aspecto de los objetos: tipo de material, color, texturas, fuentes de luz, etc.

Parámetros de visualización: es un conjunto amplio de valores que determinan como se visualiza la escena en la imagen, algunos elementos esenciales son:

- Cámara virtual: posición, orientación y ángulo de visión del observador ficticio que vería la escena como aparece en la imagen.
- Viewport: resolución de la imagen, y, si procede, posición de la misma en la ventana.

Visualización basada en rasterización

inicializar el color de todos los pixels

para cada primitiva P del modelo a visualizar

- encontrar el conjunto S de pixels cubiertos por P
- para cada pixel q de S:
 - calcular el color de P en q
 - actualizar el color de q

Las primitivas son los elementos más pequeños que pueden ser visualizados (típicamente triángulos en 3D, aunque pueden ser otros: polígonos, puntos, segmentos de recta, círculos, etc...).

La complejidad en tiempo es claramente del orden del número de primitivas (n) por el número de pixels (p) (tiempo en $O(n \cdot p)$).

Visualización basada en Ray-Tracing

inicializar el color de todos los pixels

para cada pixel q de la imagen a producir

- calcular T, el conjunto de primitivas que cubren q
- para cada primitiva P del conjunto T
 - calcular el color de P en q
 - actualizar el color de q

Cauce gráfico: conjunto de etapas de cálculo que permiten la síntesis de imágenes por rasterización.

- Las entradas al cauce gráfico se denominan primitivas. Una primitiva es una forma visible que no se puede descomponer en otros más simples.
- Un vértice es un punto del espacio 2D o 3D, extremo de una arista de un triángulo, o de un segmento de recta, o donde se dibuja un punto. Una o varias primitivas se especifican mediante una lista de coordenadas de vértices, más alguna información adicional.
- El cauce escribe en el framebuffer, que es una zona de memoria donde se guardan uno o varios arrays con los colores RGB de los pixels de la imagen (y alguna información adicional). Está conectado al monitor.

Rasterización: para cada primitiva, se calcula que pixels tienen su centro cubierto por ella.

Sombreado: (*shading*) se usan los atributos de la primitiva para asignar color a cada pixel que cubre.

Etapas del cauce gráfico El cauce gráfico tiene las siguientes etapas:

1. **Procesado de vértices:** parte de una secuencia de vértices (puntos del espacio) y produce una secuencia de primitivas (puntos, segmentos o triángulos). Tiene estas sub-etapas:
 1. **Transformación:** los vértices de cada primitiva son transformados en diversos pasos hasta encontrar su proyección en el plano de la imagen. Es realizado por un sub-programa llamado **Vertex Shader** (modificable por el programador, o programable).
 2. **Teselación y nivel de detalle:** transformaciones adicionales avanzadas, realizadas por varios programas, entre ellos el **geometry shader** (programable).
2. **Post-procesado de vértices y montaje de primitivas** incluye varios cálculos como el recortado (*clipping*) y el cribado de caras (*face culling*), ninguno de ellos programable.
3. **Rasterización (rasterization)** cada primitiva es rasterizada (discretizada), y se encuentran los pixels que cubre en la imagen de salida, no es programable.
4. **Sombreado (*shading*)**: en cada pixel cubierto se calcula el color que se le debe asignar. Se realiza por un programa llamado **fragment shader o pixel shader**, programable

Evento: ocurrencia de un suceso relevante para la aplicación. Hay varios tipos de eventos, entre otros cabe destacar:

- Teclado: pulsación o levantado una tecla, de tipo carácter o de otras teclas.
- Ratón: pulsación o levantado de botones del ratón, movimiento del ratón, movimiento de la rueda del ratón para scroll.
- Cambio de tamaño: cambio de tamaño de alguna ventana de la aplicación.

Funciones gestoras de eventos: funciones del programa que se invocan cuando ocurre un evento de un determinado tipo.

- Establece que tipos de eventos se quieren gestionar y que funciones lo harán.
- Tras invocar a una de estas funciones, se dice que el correspondiente evento ya ha sido procesado o gestionado.
- Para cada tipo de evento, la función que lo gestione debe aceptar unos determinados parámetros.

Estructura de un programa

- Variables, estructuras de datos y definiciones globales.
- Código de las funciones gestoras de eventos.
- Código de inicialización:

- Creación y configuración de la ventana (o ventanas) donde se visualizan las primitivas.
- Establecimiento de las funciones del programa que actuarán como gestora de eventos.
- Configuración inicial de OpenGL, si es necesario.
- Función de visualización de un frame o cuadro.
- Bucle principal (gestiona eventos y visualiza frames).

Bucle principal o de gestión de eventos

- GLFW mantiene una cola de eventos: es una lista (FIFO) con información de cada evento que ya ha ocurrido pero que no ha sido gestionado aún por la aplicación.
- En cada iteración se espera hasta que hay al menos un evento en la cola, entonces:
 1. Se extrae el siguiente evento de la cola: si hay designada una función gestora para ese tipo de evento, se ejecuta dicha función.
 2. Si la ejecución de la función ha cambiado el modelo de escena o algún parámetro, se visualiza un cuadro nuevo.
- El bucle termina típicamente cuando en alguna función gestora se ordena cerrarla.

Especificación de primitivas

En OpenGL, cada primitiva o conjunto de primitivas se especifica mediante una secuencia ordenada de vértices:

- Un vértice es un punto de un espacio afín 3D.
- Se representa en memoria mediante una tupla de coordenadas en algún marco de coordenadas de dichos espacio afín.
- Puede tener asociados otros valores, llamados atributos.

Existen tres tipos de primitivas: puntos, segmentos y polígonos. Por tanto, además de la secuencia de vértices, es necesario tener información acerca de que tipo de primitiva representa dicha secuencia.

Primitivas de tipo puntos y segmentos