

Tema-4-Resumen.pdf



LosCocos



Informática Gráfica



3º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada



**El más PRO del lugar
puedes ser Tú.**

**¿Quieres eliminar toda la publi
de tus apuntes?**



¡Hazte PRO!

4,95€ / mes



WUOLAH



El más PRO del lugar puedes ser Tú.



¿Quieres eliminar toda la publi de tus apuntes?



¡Fuera Publi!

Concéntrate al máximo



Apuntes a full.

Sin publi y sin gastar coins

Para los amantes de la inmediatez, para los que no desperdician ni un solo segundo de su tiempo o para los que dejan todo para el último día.



Quiero ser PRO

4,95 / mes

TEMA 4 : INTERACCIÓN Y ANIMACIÓN

INTERACCIÓN

La interacción puede ser llevada a cabo por cualquier dispositivo, al final solo tenemos unas coordenadas (x, y) de pantalla. Esto no es más que una abstracción del dispositivo hardware al dispositivo lógico.

Es gracias a la abstracción de dispositivos lógicos que OpenGL es independiente del dispositivo de entrada que se use. Dos dispositivos lógicos:

- Locator: lee posiciones (x, y) de la pantalla. Están en coordenadas de la pantalla, por lo que tendremos que hacer la inversa de la transformación para saber las coordenadas del mundo.
- Pick: que lee identificadores de componentes del modelo geométrico. Consulta la escena y devuelve el objeto bajo el cursor.

GESTIÓN DE EVENTOS EN GLUT

Los eventos son generados por la acción de los usuarios o de forma automática por el sistema. GLUT gestiona las entradas en modo evento. Tiene un buffer donde se van almacenando las peticiones de los usuarios (pulsar tecla...). Cuando el gestor de eventos de GLUT va leyendo del buffer y el evento tiene una función gestora asociada, se ejecuta y se devuelve al evento. Si no tiene ninguna función, se ignora.

Algunas funciones:

- glutMouseFunc() - glutKeyboardFunc() - glutMotionFunc() - glutIdleFunc()

CONTROL DE LA CÁMARA

Una cámara tiene muchos parámetros para ser controlada con el ratón. Lo que suele ocurrir es que se modifican de forma interactiva solo algunos de ellos.

- En visualización de modelos modificamos rápidamente la posición VNP e inclinamos el VNP. Se trata de una cámara que siempre mira al mismo sitio pero que se va moviendo como si estuviera recorriendo una escena. Se conoce como cámara orbital

- En exploración de escenarios se modifica de forma rápida la dirección VRPN e indirectamente el VRP. Tenemos una cámara fija en un punto. Ello no quiere decir que no se mueva, sino que se mueve igual que el objeto al que está asociado. Podemos controlar la dirección de la cámara con giros en X e Y del personaje. En las cámaras en tercera persona se sigue a un objeto, no tenemos más control que el movimiento del avatar.

Selección

La selección permite al usuario referenciar componentes de un modelo geométrico y es esencial en cualquier aplicación que requiera la edición de un modelo.

En la selección se realiza una búsqueda en el modelo geométrico a partir de la rayo del usuario (una coordenada en pantalla). La asociación entre posiciones de pantalla y elementos:

- Intersección rayo escena
- Codificando el id de un objeto como color y leyendo del frame buffer

SELECCIÓN EN OPENG

Para distinguir un objeto de otro, OpenGL utiliza una pila con enteros como identificadores. Dos primitivas se considera que son de objetos diferentes si su contenido en la pila es distinto. Funciones:

- `glLoadName(i)` → Sustituye nombre activo por `i`
- `glPushName(i)`
- `glPopName()`
- `glInitNames()` → Vacía la pila de nombres.

Además de la pila, cuando hacemos selección debemos decirle a OpenGL que no queremos dibujar, solo saber cuál es el objeto bajo el ratón. OpenGL tiene 3 modos:

- **Render** → Se dibujan las primitivas empujadas.
- **SELECT** → Devuelve el identificador de las primitivas en una región.
- **FEEDBACK** → Devuelve rufo geométrica de las primitivas que se dibujaron.

Esto se establece mediante la función `glRenderMode(GL_SELECT);`

BUFFER DE SELECCIÓN

Al realizar la selección usando el modo GL-SELECT OpenGL devuelve el número de objetos seleccionados como resultado y la información asociada a estos en el buffer. Debemos inicializar el buffer antes de pasar a GL-SELECT

```
GLuint selectBuf [BUFSIZE];
glSelectBuffer (BUFSIZE, selectBuf);
glRenderMode (GL-SELECT);
```

OpenGL devuelve en el buffer la secuencia de objetos pintados. El buffer tiene:

- El número de primitivas que contiene la pila
- El intervalo de profundidad de cada primitiva Z_{min} y Z_{max}
- Los nombres asociados a cada primitiva

0	1	2	3	...	$2+u-1$	$3+u-1$	$4+u-1$	$5+u-1$	
u_1	$MinZ_1$	$MaxZ_1$	Id_1	...	Id_u	u_2	$MinZ_2$	$MaxZ_2$	Id_2 ...

Para detectar qué objeto se estaba visualizando en ese instante, tomamos por ejemplo el valor con menor Z_{min} .

SELECCIÓN EN OPENGL POR CODIFICACIÓN DE COLORES

Hay otro método que consiste en crear una nueva función de dibujo para cuando queramos seleccionar y cuando el usuario hace clic se pinta la escena para seleccionar en el buffer trasero y se lee el color del pixel donde se clicó. Si no se realiza el intercambio de buffers, el usuario no ve esta escena. Pasos:

- Llamar a `dibujarSelección()`
- Leer pixel (x, y)
- Averiguar que objeto está asignado al pixel
- No intercambiar buffers.

El color del pixel se lee con la función `glReadPixels()`

ANIMACIÓN

- ANIMACIÓN POR INTERPOLACIÓN → El animador define una serie de parámetros en ciertos instantes de tiempo (puntos clave) y es el sistema el que se encarga de calcular los fotogramas intermedios por interpolación.

Estudiar **sin publi** es posible.

Compra Wuolah Coins y que nada te distraiga durante el estudio.



- ANIMACIÓN POR ESQUELETOS → Para hacer que la animación de personajes resulte plausible se suelen usar esqueletos. Un esqueleto no es más que un modelo jerárquico de segmentos rígidos unidos por articulaciones. Vinculados a los esqueletos se encuentran las mallas poligonales, que se denominan piel.

La animación del esqueleto se puede modelar utilizando interpolación, pero interpolando posición y dirección del esqueleto, no de la piel.

Dos aproximaciones.

- Cinemática directa: se definen en cada paso los ángulos de rotación de las articulaciones.
- Cinemática inversa: se define la posición inicial y final de los extremos y el sistema se encarga de calcular ángulos en cada frame.
- ANIMACIÓN PROCEDURAL → El comportamiento del objeto se controla con una función. Comportamientos fáciles de generar pero difíciles de simular físicamente.
- ANIMACIÓN POR SIMULACIÓN FÍSICA → Ciertas animaciones de la física se pueden realizar usando las leyes de la mecánica clásica. Librerías como ODE, Newton o Bullet se encargan de simular colisiones, dinámica de sólidos...
- SISTEMAS DE PARTÍCULAS → Es un conjunto de muchas partículas diminutas que juntas forman un objeto difuso. Las partículas se generan, transforman y mueren con el paso del tiempo.

Ciclo de Vida

- Generación → Las partículas se generan de forma aleatoria en el sensor ~~con~~ con una forma y posición determinada que puede variar.
- Dinámica de partículas → Los atributos cambian en cada frame
- Extinción → Cuando la edad (nº de frames-vive) de la partícula llega a su tiempo de vida esta muere. Extinción prematura:
 - Se sale del área de visualización
 - Golpea el suelo o una chispa
 - El atributo alcanza el umbral (color o negro).



WUOLAH

Scanned by CamScanner