

Apuntes teoria y ejercicios de ...



Anónimo



Informática Gráfica



3º Grado en Ingeniería Informática



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



MÁSTER EN

Inteligencia Artificial & Data Management

MADRID

Formamos
talento para un futuro
Sostenible

saber más



Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo espacio



Necesito concentración

ali ali ooh
esto con 1 coin me
lo quito yo...

WUOLAH

Apuntes Teoría – IG

PREGUNTAS EXÁMENES

1. (2.5pt) Escriba el código necesario para disponer de la funcionalidad "Zoom +" y "Zoom -", tanto en una cámara con proyección perspectiva como en una cámara con proyección ortogonal.

Suponiendo que para proyección perspectiva se utiliza una llamada con los siguientes parámetros:

```
glFrustum(left, right, bottom, top, near, far);
```

Y para la proyección ortogonal se utiliza una llamada con los siguientes parámetros:

```
glOrtho(left, right, bottom, top, near, far);
```

Siendo left, right, bottom, top, near y far atributos de una clase Cámara, tendríamos los siguientes métodos:

```
void Camara::zoomMas( ) {  
    left/=1.1;  
    right/=1.1;  
    bottom/=1.1;  
    top/=1.1;  
}  
void Camara::zoomMenos( ) {  
    left*=1.1;  
    right*=1.1;  
    bottom*=1.1;  
    top*=1.1;  
}
```

2. Describe brevemente para que sirven, en un programa OpenGL/glut, cada una de estas cuatro funciones:

- glutDisplayFunc(function);** Esta función se llamará cada vez que se dibuje la ventana.
- glutReshapeFunc(function);** Función de control del cambio de tamaño de la ventana de visualización.
- glutKeyboardFunc(function);** Función de control de eventos con el teclado.
- glutSpecialFunc(function);** Función de control de eventos con el teclado para cuando se ha pulsado una tecla especial (F1, F2, F3, ...).
- glutMoveFunc(function);** Función de control de eventos con el ratón con tecla pulsada.
- glutPassiveMoveFunc(function);** Función de control de eventos con el ratón pero sin tecla pulsada.

3. Indicar cuál será la posición de la luz en los casos siguientes:

- | | |
|--|---|
| a) GLfloat Position[4]={1,1,1,0}; glMatrixMode(GL_MODELVIEW); glPushMatrix(); glLoadIdentity(); glLightfv(GL_LIGHT0, GL_POSITION, Position); glPopMatrix(); | b) GLfloat Position[4]={1,1,1,0}; glMatrixMode(GL_MODELVIEW); glPushMatrix(); glLoadIdentity(); glRotate(45,0,1,0); glLightfv(GL_LIGHT0, GL_POSITION, Position); glPopMatrix(); |
|--|---|

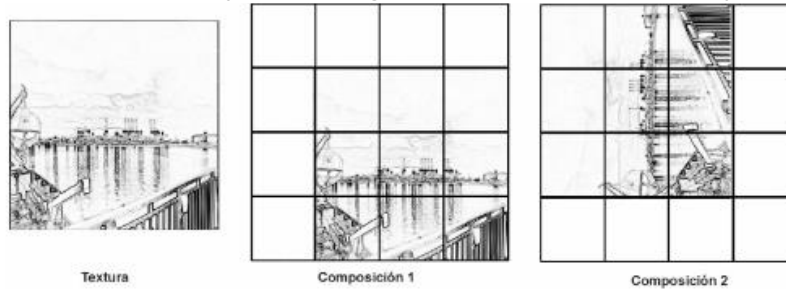
Ambos están inicialmente posicionados en las coordenadas x=1, y=1, z=1. En el caso "a" no existe modificación posterior por lo que la luz se encuentra en las coordenadas 1, 1, 1.

En el caso "b" se realiza posteriormente una rotación de 45 grados en el eje y, con lo cual la luz quedaría en las coordenadas así:

Si gira hacia derechas x=1, y=1, z=-1, y si gira hacia izquierdas x=-1, y=1, z=1.

WUOLAH

4. Se ha creado una retícula de 4x4 cuadrados. Indicar las coordenadas de textura de las esquinas de cada uno de ellos para obtener las dos composiciones siguientes con la textura de la izquierda.



Composición 1

```
GLfloat vertices[] = {
    0.25, 0.0, 0.0,    1.0, 0.0, 0.0,    1.0, 0.75, 0.0,    0.25, 0.75, 0.0 };
GLfloat texVertices[] = {
    0.0, 1.0,    1.0, 1.0,    1.0, 0.0,    0.0, 0.0 };
```

Composición 2

```
GLfloat vertices[] = {
    0.0, 0.25, 0.0,    0.75, 0.25, 0.0,    0.75, 1.0, 0.0,    0.0, 1.0, 0.0 };
GLfloat texVertices[] = {
    0.0, 0.0,    0.0, 1.0,    1.0, 1.0,    1.0, 0.0 };
```

5. Exponer las 4 características de la proyección de perspectiva.

PUNTOS DE FUGA: Se pueden utilizar 1, 2 ó 3 puntos de fuga.

ACORTAMIENTO PERSPECTIVO: Esto es debido a que cuanto más lejos esté un objeto más pequeño se verá con respecto a otros que se encuentran más cerca.

LINEAS OBLICUAS AL HORIZONTE?:

Distorsión topológica

Acortamiento perspectivo

Inversión de vista

Líneas oblicuas al horizonte

Acercamiento del infinito

Perseverancia cíclica

Ajuste de vista

6. Explique los diferentes métodos que se pueden usar para realizar la selección o pick.

La selección permite al usuario diferenciar componentes de un modelo geométrico. Para poder seleccionar las componentes deben tener asociados identificadores. Existen tres métodos de selección:

1– **Intersección rayo escena:** El punto de la pantalla se pasa de coordenadas de dispositivo a coordenadas mundiales 2D y se lanza un rayo que pasa por dicho punto y CP. Se seleccionan los objetos de la escena que intersecten al rayo.

2– **Subvolumen de visión:** Identificando los objetos incluidos en un **subvolumen de visión** centrado en la posición dada. Igual que el anterior pero se crea un volumen de vista alrededor del punto y primero se calculan los objetos que están contenidos o en parte contenidos en ese subvolumen, para así solo tener que calcular la intersección con ellos. (Más eficiente)

3– **Selección/Codificación por color:** Se crea una imagen falsa no visible donde cada elemento está identificado con un color diferente y se lee del frame buffer para ver el color de pixel donde está el punto de la pantalla y se selecciona al objeto con ese color. (Este método solo permite seleccionar el objeto más cercano.)

Imagínate aprobando el examen

Necesitas tiempo y concentración

| Planes |  PLAN TURBO |  PLAN PRO |  PLAN PRO+ |
|---|--|---|---|
|  Descargas sin publi al mes | 10  | 40  | 80  |
|  Elimina el video entre descargas |  |  |  |
|  Descarga carpetas |  |  |  |
|  Descarga archivos grandes |  |  |  |
|  Visualiza apuntes online sin publi |  |  |  |
|  Elimina toda la publi web |  |  |  |
|  Precios Anual <input type="checkbox"/> | 0,99 € / mes | 3,99 € / mes | 7,99 € / mes |

Ahora que puedes conseguirlo,
¿Qué nota vas a sacar?



WUOLAH

Informática Gráfica



Comparte estos flyers en tu clase y consigue más dinero y recompensas



Banco de apuntes de la

- 1** Imprime esta hoja
- 2** Recorta por la mitad
- 3** Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes

- 4** Llévate dinero por cada descarga de los documentos descargados a través de tu QR



7. Explica el funcionamiento del Z-buffer.

El algoritmo del Z-buffer es del tipo espacio-imagen. Cada vez que se va a renderizar un pixel, comprueba que no se haya dibujado antes en esa posición un pixel que esté más cerca respecto a la cámara (posición en eje Z). Este algoritmo funciona bien para cualquier tipo de objetos: cóncavos, convexos, abiertos y cerrados. Cuando dibujamos un objeto, la profundidad de sus pixeles se guardan en este buffer. Si utilizamos dichos pixeles en pantalla para dibujar otro objeto, se producirá la comparación de las profundidades de dichos pixeles. Si la profundidad del último pixel es mayor que la nueva (está más lejos) el pixel nuevo no se pinta, mientras que si está más cerca (la profundidad es menor), se dibuja el pixel y se guarda la nueva profundidad en el z-buffering.

Para activarlo, hay que hacer una llamada a

`glEnable(GL_DEPTH_TEST)`

Esta llamada le dice a OpenGL que active el test de profundidad. Además, cada vez que se redibuje la escena, a parte de borrar el buffer de color, hay que borrar el buffer de profundidad. Esto se hace con la llamada

`glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT).`

Por último, pero no menos importante, al inicializar OpenGL se le tiene que definir el buffer de profundidad en el modo de visualización.

8. Explique los pasos que se siguen en OpenGL para utilizar una imagen como textura.

Para utilizar una textura lo primero que tenemos que cargar la imagen mediante algún programa o biblioteca que lea este tipo de ficheros.

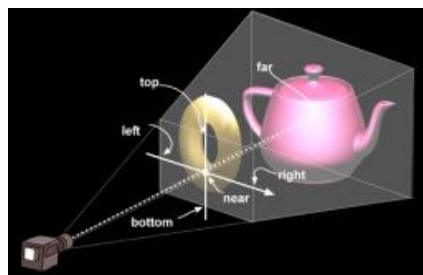
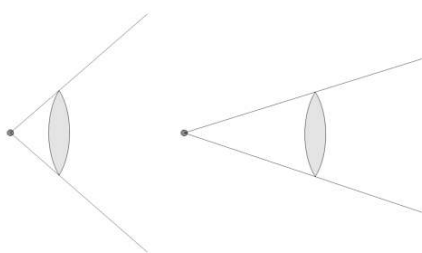
Una vez hemos cargado la textura deseada, debemos definir la textura en OpenGL, para ello utilizamos la función **`glTexImage2D();`**

Por otro lado tendremos una función que relacione los vértices de nuestro objeto a las coordenadas de la textura, éstas coordenadas determinan qué texsel en el mapa de texturas es asignado a cada vértice, haciendo así que encaje la textura con nuestro objeto.

Por último, para que esta textura sea visible debemos activarla con la función de OpenGL siguiente:

`glEnable(GL_TEXTURE_2D);`

12. Las lentes de las cámaras con zoom permiten cambiar la zona visible, desde ángulos más grandes (gran angular) a más pequeños (tele). ¿Cómo se podría conseguir el mismo efecto con los parámetros del `glFrustum`? Explicarlo y poner ejemplos de valores.



Los parámetros de `glFrustum` son 4: **`glFrustum(left, right, bottom, top, near, far);`**

Para que sea gran angular el parámetro top habrá que aumentarlo en positivo y el parámetro bottom habrá que aumentarlo en negativo. ***Con left y right igual o no hace falta?***

Para que sea tipo tele habría que hacer todo lo contrario, a los mismo parámetro disminuirlos.

Dado que `glOrtho` se define con los mismos parámetros se haría de la misma forma.

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo espacio



Necesito concentración

ali ali ooh
esto con 1 coin me
lo quito yo...

WUOLAH

TRANSFORMACIONES

-**Escalado:** `glScalef (ang,x,y,x)`

-**Rotado:** `glRotate (ang, x,y,z)`

-**Traslación:** `glTranslate(x,y,x)`

Primero se escala, después se rota y por último se traslada

ILUMINACIÓN

Ambiental: luz ambiente no tiene luz definida pero da claridad de forma uniforme.

Especular: luz intensa que produce el brillo.

Difusa: luz que genera sombra, incide pero no tiene dirección clara.

Hay 8 luces diferentes (GL_LIGHT_0 ... GL_LIGHT_7)

`glEnable(GL_LIGHT_0);`

`glDisable(GL_LIGHT_0);`

TIPO DE LUZ

Posicional: Puntuales y focales.

Direccional: Está lejos y todos los rayos van paralelos.

`Gl_Lightfu(gl_light, gl_position, vector)` **vector** $(x,y,z,k) \rightarrow k=1$ posicional y $k=0$ direccional

TRANSFORMACIÓN DE LAS VISTAS

VRP: Punto de referencia (Posición o punto que le pongo)

VPN: Plano normal (Perpendicular al plano de proyección)

VUP:

WUOLAH