

RELACION-PROBLEMAS-IG-COMPLETA.pdf



AngMGR



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



Esto no son apuntes pero tiene un 10 asegurado (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandés con una garantía de hasta 100.000 euros por depositante. Consulta más información en [ing.es](https://www.ing.es)



RELACIÓN PROBLEMAS:

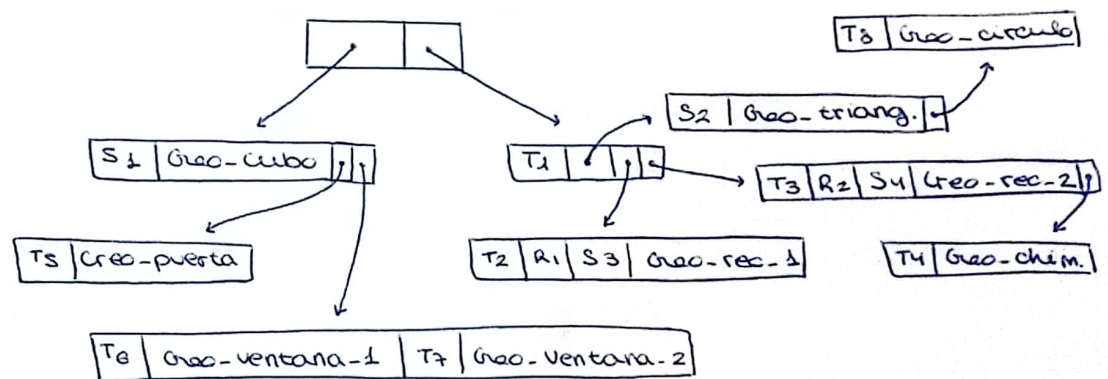
1. Dado el modelo 2D de la figura 1, indica como se haría en OpenGL.

La figura nos muestra una casa que podemos dividir como formas geométricas aisladas. La base de la casa sería un cubo y sobre este montaríamos un triángulo (en 3D) y dos rectángulos, uno a cada lado del triángulo para montar el tejado.

Para las puertas y ventanas haríamos rectángulos en 2D y le aplicaríamos texturas para que aparezcan en los dibujos que vemos y la ventana circular sería simplemente un círculo en 2D.

Todo lo descrito debe hacerse aplicando transformaciones geométricas. Si vemos la figura como un grafo de escena quedaría de la siguiente forma:

- Partimos de figuras de tamaño unidad y en el origen



S1 - escalado del cubo, quedando en (2, 2, 2) (escalado al doble)

T1 - traslación del triángulo a la posición (1, 2, 1), lo pongo centrado encima del cubo

S2 - escalado del triángulo al doble \rightarrow (2, 2, 2)

T2 - traslación de la primera parte del tejado, quedaría en la posición (1, 4, 2)

R1 - rotación de la pieza para que quede inclinada sobre el triángulo, sería de 45° sobre el eje z

S3 - escalado de la pieza al doble de tamaño, (2, 2, 2)

T3 - traslación de la segunda parte del tejado, quedando en (2, 4, 2)

Consulta condiciones aquí



do your thing

WUOLAH

Escaneado con CamScanner

R2 - rotación de la piera 45° en z

S4 - escalado a $(2, 2, 2)$

T4 - traslación de la chimenea $(1'S, 3, 1'S)$

S7 - escalado del círculo a $(2, 2, 2)$

T8 - traslación sobre el triángulo a $(1, 2, 2)$ centrada en él

T5 - traslación de la puerta a la posición $(1, 1'S, 2)$

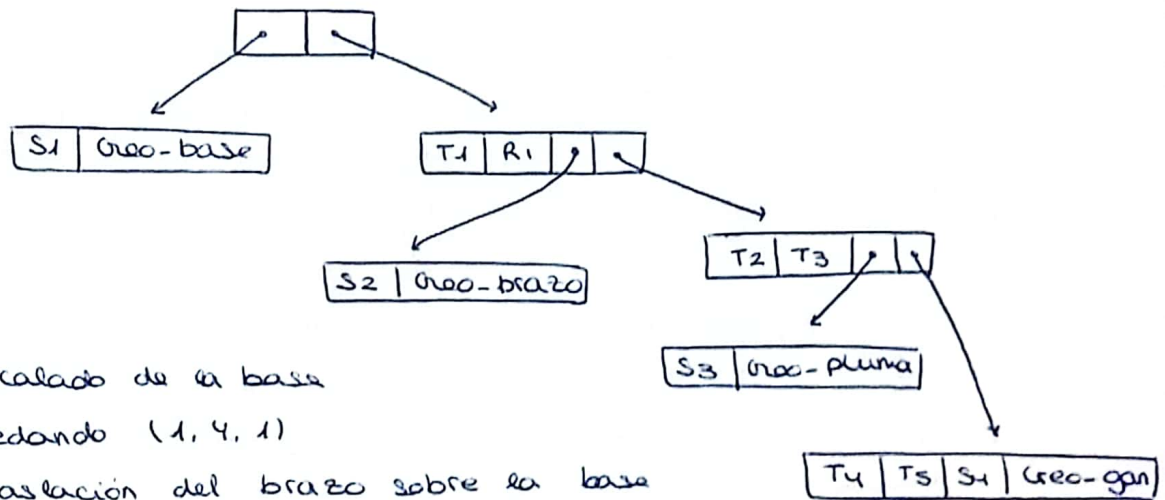
T6 - traslación de la ventana 1 a $(0'S, 1, 2)$

T7 - traslación de la ventana 2 a $(1'S, 1, 2)$

Las piezas de "decoración" son de tamaño unidad y no escala ya que las demás van a aumentar el tamaño al doble y si escalara al mismo tamaño las "decoraciones" ocuparía el espacio de la figura base.

2. Diseñar el gráfico de escena de la figura.

Supongo que parto de un cubo de tamaño unidad en el origen



S1 - escalado de la base
quedando $(1, 4, 1)$

T1 - traslación del brazo sobre la base
quedando en la posición $(0'S, 4, 0'S)$ centrado sobre el base.

R1 - movimiento rotatorio de 45° en el eje z

S2 - escalado del brazo $(1, 2, 1)$

T2 - traslación de la pluma bajo el brazo a la pos $(-3, 4, 0'S)$

T3 - movimiento de subida y bajada

S3 - escalado de la pluma $(1, 1'S, 1)$

T4 - traslación del gancho al centro de la pluma $(-2, 2, 0'S)$

T5 - movimiento independiente de subida y bajada

S4 - escalado a $(1'S, 1'S, 1'S)$

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandés con una garantía de hasta 100.000 euros por depositante. Consulta más información en ing.es

Que te den **10 € para gastar**
es una fantasía.
ING lo hace realidad.

Abre la **Cuenta NoCuenta** con el código
WUOLAH10, haz tu primer pago y llévate 10 €.

Quiero el cash

[Consulta condiciones aquí](#)



do your thing

Informática Gráfica



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



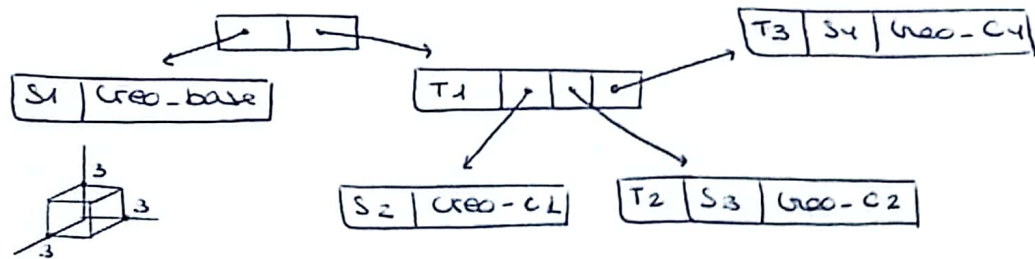
Banco de apuntes de la

- 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



3. Realizar en OpenGL el modelo de la T mostrado en la figura



S1 - escalado de la base, $S(3, 3, 3)$

T1 - traslación de C1 encima de la base, $T(0, 3, 3)$

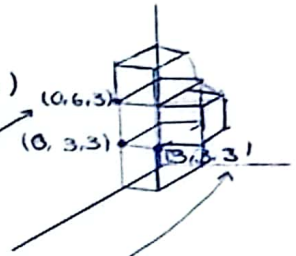
S2 - escalado $S(3, 3, 3)$

T2 - traslación de C2 sobre C1, $T(0, 6, 3)$

S3 - escalado $S(3, 3, 3)$

T3 - traslación de C4, $T(3, 3, 3)$

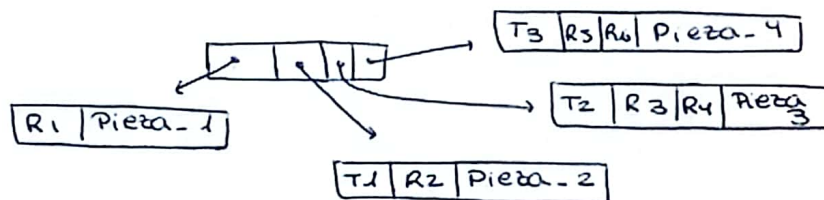
S4 - escalado de C4, $S(3, 3, 3)$



6. ¿Es posible construir un modelo recursivo en OpenGL?

La recursividad es una técnica en programación donde una función se llama a sí misma para resolver un problema más grande. Se puede implementar una función recursiva en OpenGL diseñando una que tenga un comportamiento como el descrito. Por ejemplo, renderice un objeto y luego se llama a sí misma para renderizar sub-partes o instancias del mismo objeto.

7. Generar el modelo necesario para obtener el mosaico de la figura 5, partiendo de la figura generada en el ejercicio 3



R1 - rotación en el eje z de 270°

R2 - rotación de la pieza 2 -90° en el eje x

T1 - traslación a $T(3, 3, 9)$

R3 - rotación de la pieza 3 90° en el eje y

R4 - rotación de la pieza 3 -90° en el eje x

T2 - traslación a $T(3, 3, 0)$



R_5 - rotación para que el saliente mire hacia dentro 180° en Y

R_6 - rotación para que se cambie 270° en el Z

T_3 - traslación a $T(9, 3, 3)$

3. Indicar la forma en que se podría conseguir en OpenGL el modelo de la figura 6, de tal modo que simplemente añadiendo una transformación geométrica se puedan separar los dos cubos de forma simétrica del centro.

la transformación que necesitamos es una traslación. la idea es trasladar uno de los cubos en una dirección y el otro en la opuesta.

`glTranslatef(-1, 0, 0);`

`drawCubo();`

`glLoadIdentity();` // restablecemos la matriz modelo-visual

`glTranslatef(1, 0, 0);`

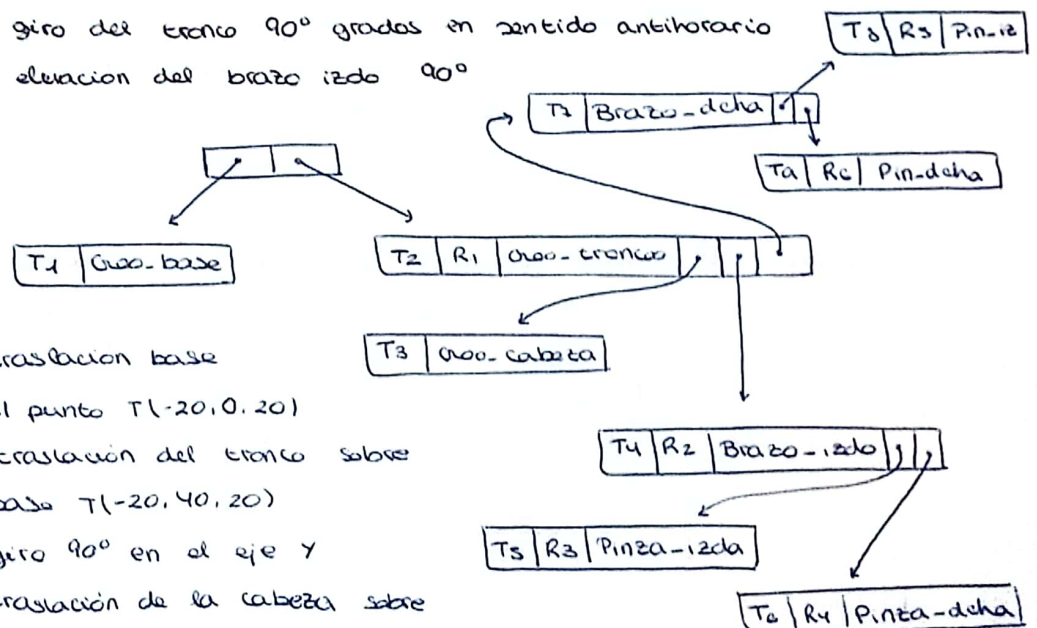
`drawCubo();`

9. Diseñar el robot de la figura 7

10. Introducir en el modelo los siguientes cambios

1. giro del tronco 90° grados en sentido antihorario

2. elevación del brazo izdo 90°



T_1 - traslación base

al punto $T(-20, 0, 20)$

T_2 - traslación del tronco sobre la base $T(-20, 40, 20)$

R_1 - giro 90° en el eje Y

T_3 - traslación de la cabeza sobre el tronco $T(-10, 120, 20)$

T_4 - traslación del brazo izdo al lateral del tronco $T(-20, 115, 20)$

R_2 - elevación del brazo 90° en Z



T5 - traslación pinza izda a $T(-35, 75, 20)$

R3 - rotación pinza izda -90° en Y

T6 - traslación pinza deha a $T(20, 75, 20)$

R4 - rotación pinza izda -90° en Y

T7 - traslación brazo deha a $T(20, 115, 20)$

T8 - traslación pinza izda a $T(35, 75, 20)$

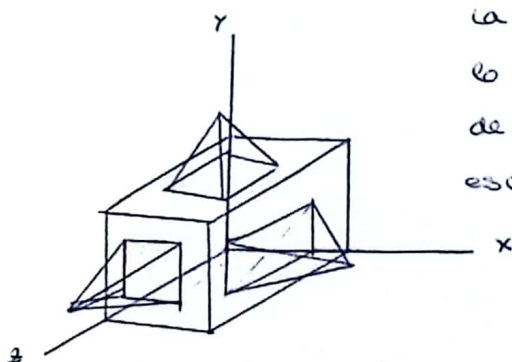
R5 - rotación pinza izda -90° en Y

T9 - traslación pinza deha a $T(20, 75, 20)$

R6 - rotación pinza izda -90° en Y

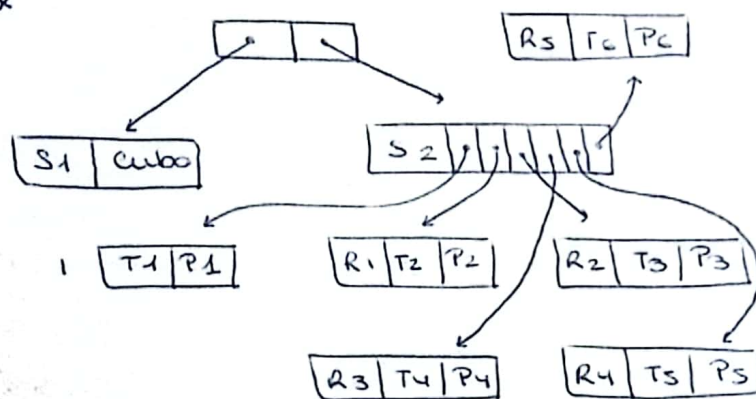
(Se ha tomado como tamaño el máximo de las piezas)

4. Generar un modelo formado por un cubo central de tamaño 10 y 6 pirámides cuadradas de base 5, adosadas a sus caras a partir de un cubo de lado 1 y una pirámide de base cuadrada y lado 1.



Cada una de las pirámides las rotamos (excepto la primera que va arriba) y las ponemos en posición

la figura pedida sería algo como lo mostrado. Para crearla vamos a partir de un cubo de tamaño unidad que escalaremos a tamaño 10 $S(10, 10, 10)$



$S_1 \rightarrow S(10, 10, 10)$

$S_2 \rightarrow S(5, 5, 5)$

arriba $\leftarrow T_1 \rightarrow T(2'S, 10, 2'S)$

delante $\left\{ \begin{array}{l} R_1 \rightarrow R(12, 0, 0, 12) \\ T_2 \rightarrow T(2'S, 75, 10) \end{array} \right.$

destr. $\left\{ \begin{array}{l} R_3 \rightarrow (12, 12, 0, 0) \\ T_4 \rightarrow (2'S, 2'S, 0) \end{array} \right.$

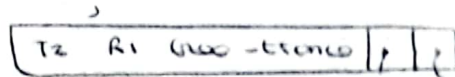
deha $\left\{ \begin{array}{l} R_2 \rightarrow (12, 0, 0, 12) \\ T_3 \rightarrow (10, 2'S, 2'S) \end{array} \right.$

izda $\left\{ \begin{array}{l} R_4 \rightarrow (12, 0, 0, 12) \\ T_5 \rightarrow (0, 2'S, 2'S) \end{array} \right.$

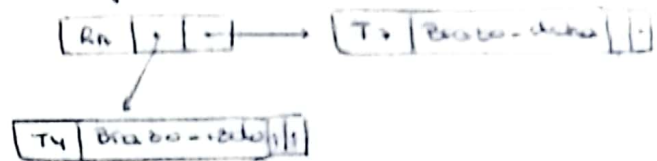
emp $\left\{ \begin{array}{l} R_5 \rightarrow (12, 12, 0, 0) \\ T_6 \rightarrow (25, 0, 2'S) \end{array} \right.$

11. Modificar el grafo para que el movimiento de elevación de los dos brazos sea solidario, conservándose el movimiento independiente de cada una de las muñecas y pinzas

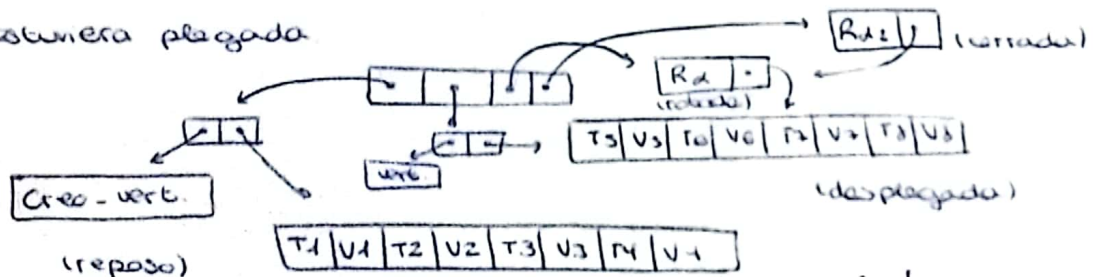
Para que la elevación de los brazos sea conjunta tendremos que poner una rotación común para ambos quedando la parte de los brazos



siendo R_1 una rotación $R(90, 1, 0, 0)$ y quedando el resto de transformaciones intactas



12. Queremos obtener el modelo de una persona de varillas. Primero vamos a crear un ventano de varillas de base uno si estuviera plegada.



plegada
tiene
c/s de
separ.
entre
varillas

$T_1 \rightarrow (-2.5, 9.5, 0)$
 $T_2 \rightarrow (-2.5, 9, 0)$
 $T_3 \rightarrow (-2.5, 8.5, 0)$
 $T_4 \rightarrow (-2.5, 8, 0)$

$T_5 \rightarrow (-2.5, 9.5, 0)$
 $T_6 \rightarrow (-2.5, 8.5, 0)$
 $T_7 \rightarrow (-2.5, 7.5, 0)$
 $T_8 \rightarrow (-2.5, 6.5, 0)$

desplegada
hay una
separación
de 1 entre
una varilla

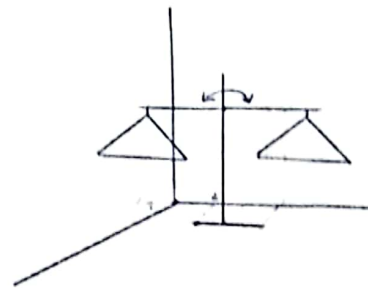


$R_1 = R(45, 1, 0, 0)$ (para rotarla)

$R_2 = R(45, 1, 0, 0)$ (para cerrarla, hacemos 45 de α_1 y 45 de α_2 por lo que un giro total da 90)



15 Modelar la balanza como la de la figura 10. Redactar un procedimiento de edición para girar la balanza un determinado ángulo.



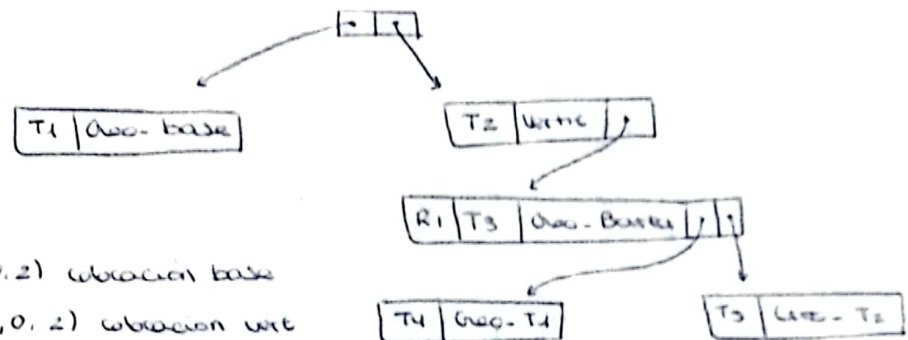
Medidas

Base → 3 largo

Vertical → 10 largo

Barra → 10 largo

Triángulos → base 3, alto 3



$T_1 \rightarrow (2, 0, 2)$ ubicación base

$T_2 \rightarrow (2, 5, 0, 2)$ ubicación vert

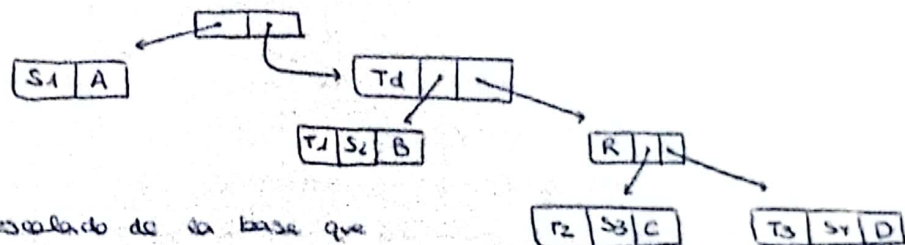
$R_1 \rightarrow (2, 1, 0, 0)$ movimiento balanza

$T_3 \rightarrow (2, 5, 8, 2)$ traslación de la barra al centro de la vertical

$T_4 \rightarrow (1, 1, 5, 8, 2)$

$T_5 \rightarrow (4, 5, 8, 2)$ | traslación de los triángulos bajo la barra

15 Diseñar un modelo jerárquico para el brazo robot a partir de un cilindro de altura 1 y radio de base 1.



S_1 - escalado de la base que

esta situada en el origen $S(4, 4, 4)$

S_2 - escalado de la pieza B $S(2, 4, 2)$

T_1 - traslación de B sobre A $T(2, 4, 2)$

T_d - movimiento de la pieza B que afectará a toda la parte superior, es un movimiento que aumentará y decrementará y



R - movimiento rotatorio de C que apunta a D, $R(1, 1, 0, C)$

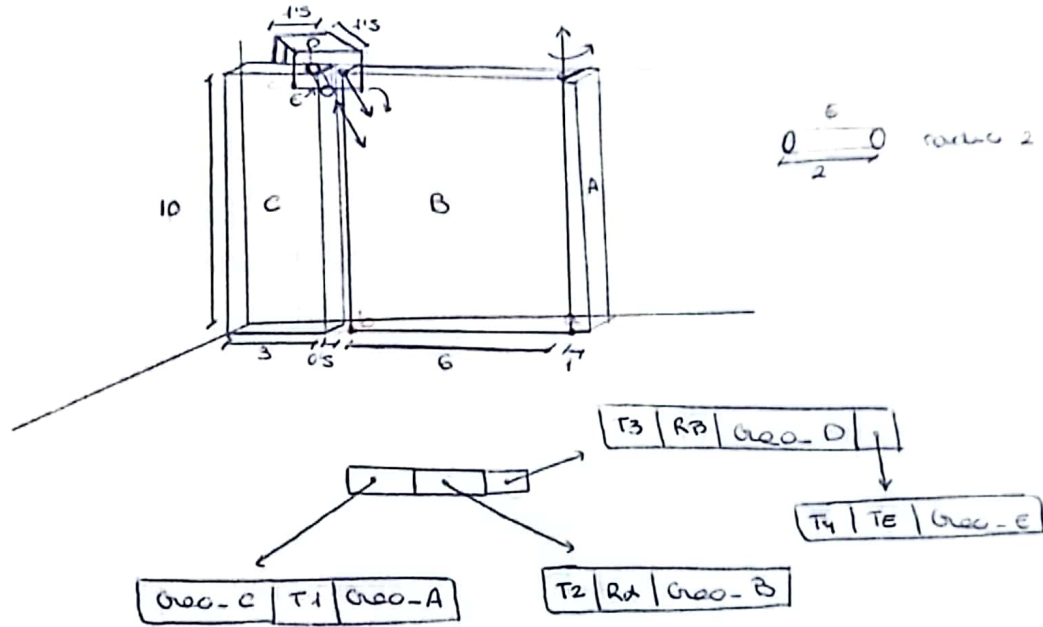
S3 - escalado de C $(2, 2, 2)$

T2 - traslación de C sobre B, $T(2, 8, 2)$

S4 - escalado de la pieza D, $S(2, 4, 2)$

T3 - traslación de D a la cara lateral de C $T(4, 8, 2)$

16 Diseñar el modelo jerárquico para la escena



T1 → traslación de A al punto $T(9.5, 0, 1)$ (a)

R4 → movimiento rotatorio entorno a la bisagra vertical $R(1, 0, 1, C)$

T2 → traslación de B $T(3.5, 0, 1)$ (b)

R3 → movimiento del cubo $R(3, 0, 0, 1)$

T3 → traslación sobre C y B $T(2, 9, 1)$ (c)

TE → deslizamiento de E sobre D, me variara el valor de 2

T4 → traslación de E al frente de D, $T(2.25, 4.5, 1.5)$

17. Indica la posición y parámetros aproximados de las fuentes de luz de la figura 14, y las propiedades y modo de sombreado de los materiales

PROPIEDADES Y MODO DE SOMBRADO

- esfera → difusa
- rosco → especular + brillo + difusa
- tetra → ambiente por la parte en sombra + especular + difusa
- base → la zona donde están apoyados los objetos difusa + ambiente de las piezas laterales

FUENTES LUZ

Habrán dos ya que la iluminación del vaso y la tetera no tienen relación. Una de ellas viene desde arriba e incide en la tapa de la tetera. La otra viene desde detrás de la tetera incidiendo en el lateral del vaso. Ambas de forma perpendicular.

13. Indica la posición y parámetros aproximados de las fuentes de luz de la figura y las propiedades y modos de sombreo de los materiales.

FUENTES DE LUZ

Tenemos una fuente de luz que incide desde arriba a los dos objetos y con una posición perpendicular sobre el tejado

MATERIALES

- tejado → especular + difuso
- cubo-casa → ambiente
- carro - parte-arriba → difuso
- carro resto → ambiente

19. Indica la posición y parámetros aproximados de las fuentes de luz de la figura 16. y las propiedades y los modos de sombreo de los materiales

FUENTES LUZ

No tenemos ninguna fuente incidiendo directamente sobre la escena ya que no vemos sombras.

MATERIALES

Todos los objetos presentan ambiente

20. Indica la posición y parámetros aproximados de las fuentes de luz de la Figura 17, y las propiedades y modo de sombreo de los materiales

FUENTES DE LUZ

Tenemos una fuente de luz que incide perpendicularmente por el

Esto no son apuntes pero tiene un 10 asegurado (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandés con una garantía de hasta 100.000 euros por depositante. Consulta más información en [ing.es](https://www.ing.es)



lateral derecho

MATERIALES

- bola → especular + difuso + brillo
- casco → especular + difuso
- barra → ambiente (laterales y arriba) + difuso (cara - delantera)
- base → difuso (parte arriba) + ambiente (laterales)

21. Indica la posición y parámetros aproximados de las fuentes de luz de la figura 19, y las propiedades y modo de sombreado de los materiales

FUENTES LUZ

tenemos una fuente de luz para el grupo de las 3 unidades de la derecha que incide perpendicularmente desde el fondo-lateral derecho y otra para el árbol de la izquierda que incide perpendicularmente desde el frente si nos ponemos en la posición del árbol

MATERIALES

- suelo → ambiente
- casas → ambiente
- árboles - copa → difuso
- árboles - tronco → difuso

22. Indica la posición y parámetros aproximados de las fuentes de luz de la Figura 19, y las propiedades y modo de sombreado de los materiales.

MATERIALES

- cubo → cada cara tiene un color y difuso
- pirámide → difusa con un color para cada vertice que es lo que permite que se vea ese degradado

FUENTES DE LUZ

Consulta condiciones aquí



do your thing

23. Indica las llamadas que tendrías que hacer en OpenGL para las escenas de los ejercicios anteriores

la fuentes de luz en establecamos en la llamada a la función `glLightfv(GL_LIGHT1, GL_POSITION, position)`, siendo `GL_LIGHT1` la luz que aplicamos, en OpenGL tenemos un máximo de 8, `GL_POSITION` es la especificación de la posición de la luz respecto a las coordenadas y posición será el vector de coordenadas donde poner la luz.
los materiales los aplicaremos a cada objeto en particular:

• DIFUSO: `GLfloat diffuse[] = {1, 1, 1, 1}` valores

`glMaterialfv(GL_FRONT, GL_DIFFUSE, diffuse)`

• AMBIENTE: `GLfloat amb[] = {1, 1, 1, 1}`

`glMaterialfv(GL_FRONT, GL_AMBIENT, amb)`

• ESPECULAR: `GLfloat esp[] = {1, 1, 1, 1}`

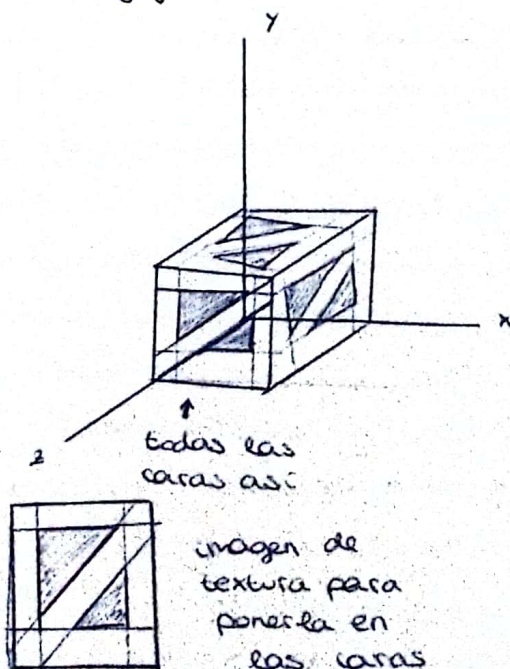
`glMaterialfv(GL_FRONT, GL_SPECULAR, esp)`

• BRILLO: `GLfloat brill[] = {1, 1, 1, 1}`

`glMaterialfv(GL_FRONT, GL_SHININESS, brill)`

Para que las propiedades del material se vean correctas tenemos que ajustar los valores de cada uno y posicionar adecuadamente la fuente de luz.

24. Indicar la forma en que se podría construir en OpenGL el modelo de la figura 20.



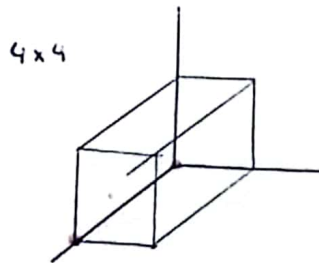
Aplicaríamos la misma textura a cada una de las caras del cubo. Primero buscaríamos una imagen que corresponda con ese patrón que vemos. Una vez obtenida la imagen la aplicaríamos a cada una de las caras del cubo cambiando y ajustando las coordenadas de textura usando la función `glTexCoord(x, y)` antes de la construcción de cada vertice de la cara del cubo.

Se codificaría de la siguiente forma una cara, por ejemplo la delantera:

```
glBegin(GL_QUAD), 1
    2FglTexCoord(0,0); glVertex3F(x, y, z),
    2FglTexCoord(0,1); glVertex3F(x, y, z);
    2FglTexCoord(1,1); glVertex3F(x, y, z),
    2FglTexCoord(1,0); glVertex3F(x, y, z),
    ...
    glEnd();
```

y repetiríamos esto para el resto de caras

25. Explica en detalle como crearías un dado en OpenGL como generarías las coordenadas de textura?



```
glBegin(GL_QUADS), 1
```

cara
delantera

```
glEnable(GL_AUTOWRHAL),
glVertex3F(0,0,4),
glVertex3F(0,4,4),
glVertex3F(4,4,4),
glVertex3F(4,0,4),
```

cara
fondo

```
glEnable(GL_AUTO-NORMAL)
glVertex3F(0,0,0),
glVertex3F(0,0,4),
glVertex3F(4,0,4),
glVertex3F(4,0,0),
```

cara

```
glEnable(GL_AUTO-NORMAL),
glVertex3F(0,0,0),
glVertex3F(0,4,0),
glVertex3F(4,4,0),
glVertex3F(4,0,0),
```

cara
super.

```
glEnable(GL_AUTO-NORMAL)
glVertex3F(0,4,4),
glVertex3F(0,4,0),
glVertex3F(4,4,0),
glVertex3F(4,4,4),
glEnd();
```

lateral
dcha.

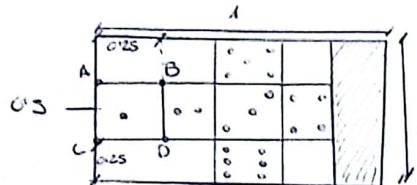
```
glEnable(GL_AUTO-NORMAL),
glVertex3F(4,0,4),
glVertex3F(4,4,4),
glVertex3F(4,4,0),
glVertex3F(4,0,0),
```

lateral
izda.

```
glEnable(GL_AUTO-NORMAL),
glVertex3F(0,0,4),
glVertex3F(0,0,0),
glVertex3F(0,4,0),
glVertex3F(0,4,4)
```



Para crear un dado lo primero que haría sería crear un cubo de la forma anterior y para que se vea como un dado buscaría una imagen como:



E iría aplicando las coordenadas de textura a cada uno de los vértices del cubo con la función `glTexCoord(x, y)`.

Las coordenadas de textura para la cara del 1 serían:

- (A) `glTexCoord(0, 0.25); glVertex3f(0, 0, 4);` aplico a cada vértice
- (B) `glTexCoord(0.25, 0.25); glVertex3f(0, 4, 4);` de la cara, en este
- (C) `glTexCoord(0, 0.75); glVertex3f(4, 4, 4);` caso la cara delantera
- (D) `glTexCoord(0.25, 0.75); glVertex3f(4, 0, 4);` del dado

Así lo haríamos con las 6 caras para que el cubo de origen se vea como un dado.

26. Explica como crearías un modelo de la Tierra usando OpenGL. ¿Cómo generarías las coordenadas de textura?

Para crear un modelo de la Tierra partiremos de una esfera que podemos crear con la función `glutSolidSphere`. Una vez tenemos la esfera buscaríamos una imagen de la Tierra y después aplicaríamos las coordenadas de textura a la esfera. Para definir las coordenadas de textura de cada vértice de la esfera utilizaremos las coordenadas esféricas y las mapearemos a las coordenadas de textura. Utilizamos la función `glTexCoord2f` para especificar esas coordenadas de textura, que se calcularán en función de los coordenados esféricos de cada vértice.

27. Explica como crearías un modelo del sistema solar en el que se viera el cambio de iluminación en la tierra a lo largo del día.

Para crear el modelo del sistema solar implementaríamos lo descrito

WUOLAH



en el ejercicio 26 para cada uno de los planetas que forman el sistema y para el sol

la tierra orbita alrededor del sol que es la estrella que nos aporta la luz. Por tanto, de esta dependerá la cantidad de iluminación, también depende de la variación de la órbita. Por tanto cuando se aleja del sol la iluminación cambia.

Para ver este efecto vamos a poner un foco de luz desde la posición en la que tengamos ubicado el sol y daremos un movimiento rotatorio a la Tierra (simulando la rotación) y se irá alejando cada cierto tiempo. Dependiendo de la distancia y del giro se iluminarán unas caras u otras de la Tierra con una mayor o menor intensidad.

23. Explica como se pueden simular eclipses en el modelo anterior. Un eclipse se dará cuando:

- la Tierra se encuentra entre el sol y la luna
- la luna esta entre el sol y la Tierra

cuando tenemos un cuerpo en medio de ambas y la fuente de luz viene del sol lo que ocurre es que el objeto central bloquea esa luz. Por tanto, para ver un eclipse tendríamos, por ejemplo, que poner la luna entre la Tierra y el Sol en una trayectoria tal que tape la luz que llegaría a la Tierra. desde el sol (el sol es desde donde viene el foco de luz según el modelo del ejercicio 27)

29. Escribe una función para dibujar un cono.

Para dibujar un cono en OpenGL, podemos utilizar la función `glutSolidCone` utilizando la biblioteca GLUT. Esta función tiene los parámetros: radio de la base, altura, slices y stack en ese orden.

Slice → nº de subdivisiones alrededor del cono

stack → nº de subdivisiones a lo largo del cono.