

Examen-Resuelto-Enero-2023.pdf



la_zowi



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

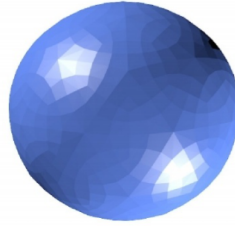
Puedo eliminar la publi de este documento con 1 coin

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

1. (1 punto) La imagen siguiente corresponde a la visualización de un poliedro regular de radio 1, centrado en el origen que aproxima una esfera de color azul medio. Indica:

a) Que fuentes de luz y que propiedades de material se han usado para generarla.

b) Que cambios se deben hacer en el código para que la visualización simule una esfera, indicando como se calcularían las normales.



Viva la-zowi ♡

a) Se han utilizado dos fuentes de luz blancas, si la cámara estuviese situada en el eje Z, estas luces podrían estar en $(-1, 1, 1)$ y en $(1, -1, 1)$.

Para la esfera se han utilizado una luz ambiente nula, una luz difusa $(0, 0, 255)$, una especular $(1, 1, 1)$ y un brillo medio.

b) Se debería hacer de forma que la iluminación se calculase con las normales de los vértices puesto que es ser un objeto suave. Para calcular las normales de los vértices de una esfera es muy simple puesto que solo hace falta transformar el vector que une el va del centro de la esfera (conocido) con el vértice y transformarlo en unitario. Si el centro se encuentra en el origen es más fácil puesto que nuestro vector \vec{OV} será el valor de V.

$$\vec{N} = \frac{\vec{OV}}{|\vec{OV}|}$$

perdo espacio



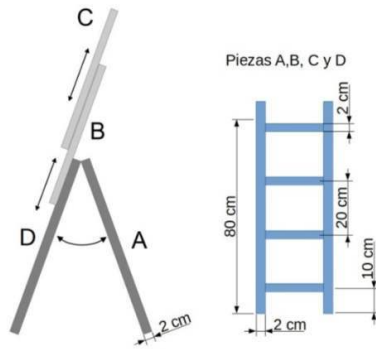
Necesito concentración

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

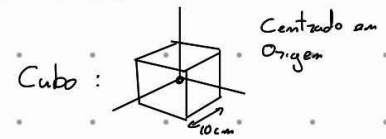
WUOLAH

WUOLAH

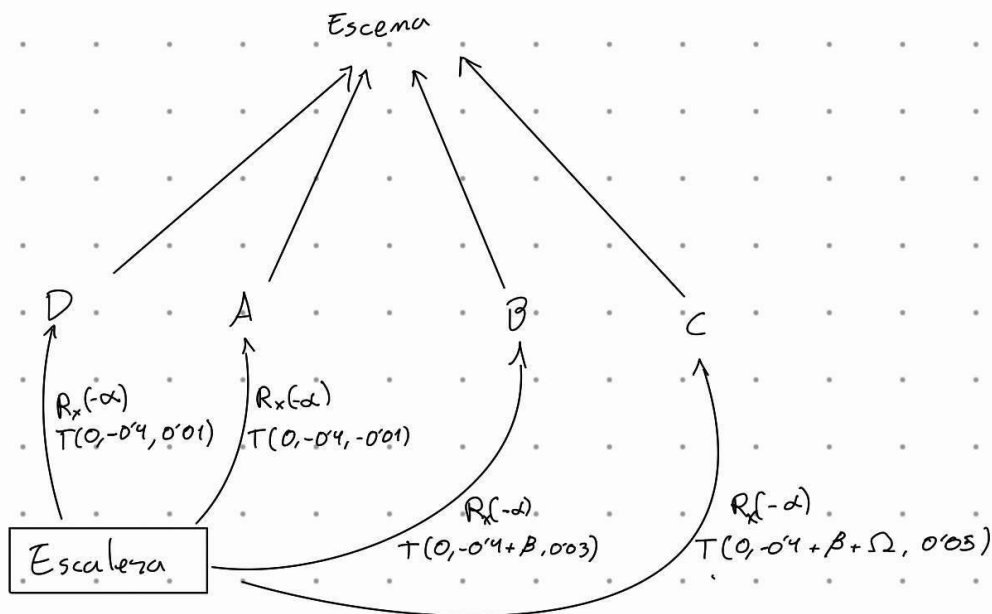
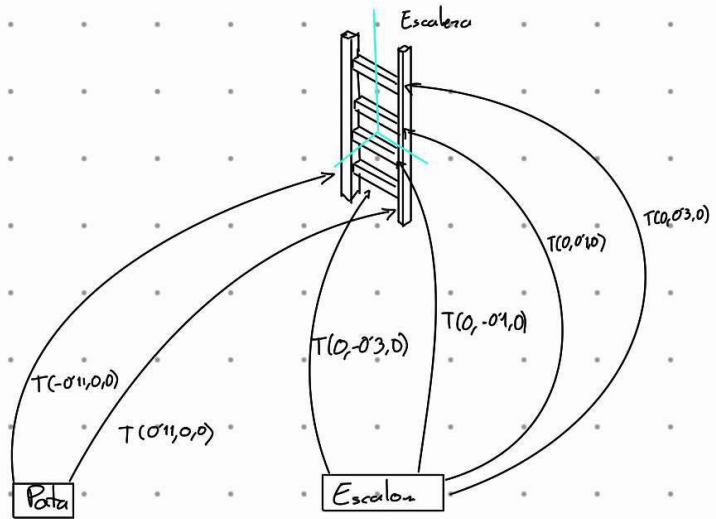
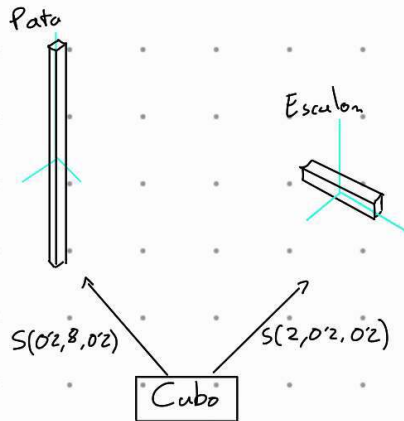
2. (3 puntos) Diseñar el grafo de escena para la escalera de la figura (tres grados de libertad). Se creará a partir de un cubo de lado unidad alineado con los ejes de coordenadas con el centro de la cara inferior en el origen de coordenadas. Las cuatro piezas (A, B, C y D) son iguales.



Suponemos:



Longitud escalon = 20 cm



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



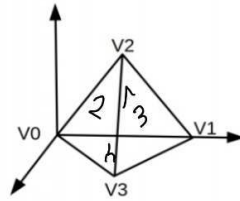
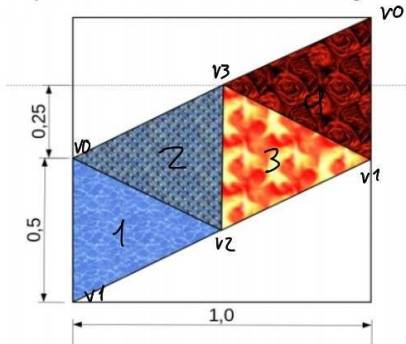
Banco de apuntes de la

- 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



3. (2 puntos) Escribe el código en C++ con OpenGL que dibuje un tetraedro con lados de tamaño unidad aplicándole la textura que se muestra en la figura. Anotar las coordenadas usadas para los vértices del tetraedro en la figura de la derecha.



$$\begin{aligned} v_0 &= (0, 0, 0) & v_1 &= (1, 0, 0) \\ v_2 &= (0.5, 0.866, 0.433) & v_3 &= (0.5, 0, 0.866) \end{aligned}$$

```
GLfloat vertices[] = { 0, 0, 0, 1, 0, 0, 0.5, 0.866, 0.433, // Cara 1
                      0, 0, 0, 0.5, 0.866, 0.433, 0.5, 0, 0.866, // Cara 2
                      0.5, 0.866, 0.433, 0.5, 0.866, 0.433, 1, 0, 0, // Cara 3
                      0.5, 0.866, 0.433, 1, 0, 0, 0, 0, 0 };
```

```
GLfloat texVertices[] = { 0, 0.5, 0, 1, 0.5, 0.75, // Cara 1
                          0, 0.5, 0.5, 0.75, 0.5, 0.25, // Cara 2
                          0.5, 0.25, 0.5, 0.75, 1, 0.5, // Cara 3
                          0.5, 0.75, 1, 0.5, 1, 0 };
```

Esto es lo que importa más

Ahora solo quedaria habilitar las texturas, los arrays de vertices y texturas, asignar la textura, el numero de vertices por cara y de coordenadas por textura, y ya ejecutar la función de dibujar los arrays.

```
glEnable(GL_TEXTURE_2D);
glEnableClientState(GL_VERTEX_ARRAY);
glEnableClientState(GL_TEXTURE_COORD_ARRAY_EXT);
glBindTexture(GL_TEXTURE_2D, textureid);
glVertexPointer(3, GL_FLOAT, 0, vertices);
glTexCoordPointer(2, GL_FLOAT, 0, texVertices);
glDrawArrays(GL_TRIANGLES, 0, 4);
```

Importante

Puedo eliminar la publi de este documento con 1 coin

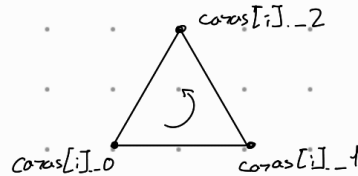
¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

4. (2 puntos) Sea la siguiente representación de una malla de triángulos. Describe cómo se puede determinar si la orientación de las normales de las caras es consistente (están orientadas hacia el exterior del objeto).

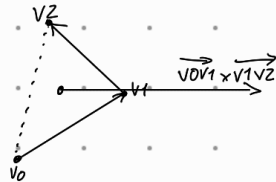
```
class _puntos3D
{
...
vector<_vertex3f> vertices;
};

class _triangulos3D: public _puntos3D
{
    _triangulos3D();
    ...
    void calcular_normales_caras();
    vector<_vertex3i> caras;
    vector<_vertex3f> normales_caras;
}
```

Tomando como por hecho que los vertices de los caras siguen este orden:



Se que si hago el producto vectorial de los vectores $\vec{v_0v_1} \times \vec{v_1v_2}$ obtengo el vector normal de la cara que forman esos tres vertices:



Solo bastaría con cada normal comprobar que el signo de los valores x, y, z de la normal de la cara coinciden con los signos de nuestro producto vectorial con los vertices de esa cara. Si son los mismos signos entonces la normal será correcta.

5. (2 puntos) Explica como se construye un volumen de vista en perspectiva señalando los parámetros necesarios. Para OpenGL indica las distintas formas de realizar un zoom para una cámara en perspectiva (`glFrustum(...)`) y además, como orientar este tipo de cámara para tener un plano de proyección picado (oblicuo a 45 grados respecto al objeto enfocado).

Se hace mediante la función `glFrustum(left, right, bottom, top, nearVal, farVal)`.

'left', 'right': Coordenadas x

'bottom', 'top': Coordenadas y

'nearVal', 'farVal': Distancias desde la cámara al plano cercano y lejano del volumen de vista.

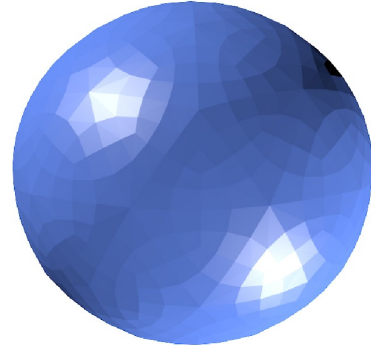
Para realizar un zoom bastaría con disminuir los valores de coordenadas x e y o aumentando la distancia "nearVal" y/o reduciendo la distancia "farVal".

Examen Informática Gráfica (24/01/23)

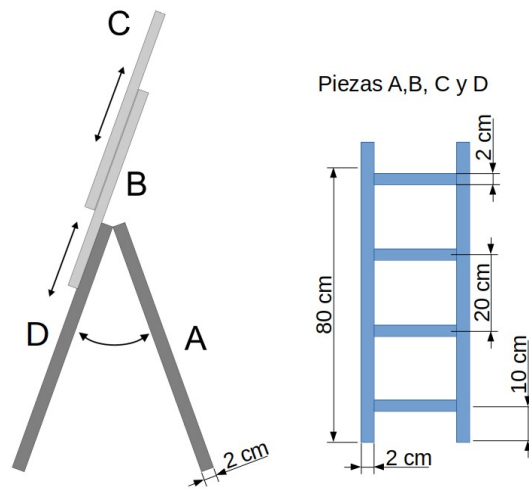
Apellidos: _____ Nombre _____ Grupo _____
_(puntuación expresada sobre 10. Mínimo 4,0 para que cuente)

1. (1 punto) La imagen siguiente corresponde a la visualización de un poliedro regular de radio 1, centrado en el origen que aproxima una esfera de color azul medio. Indica:

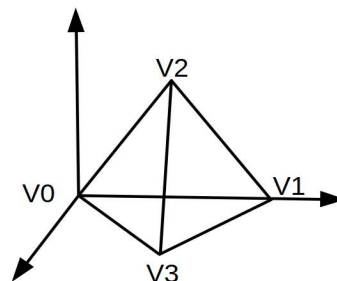
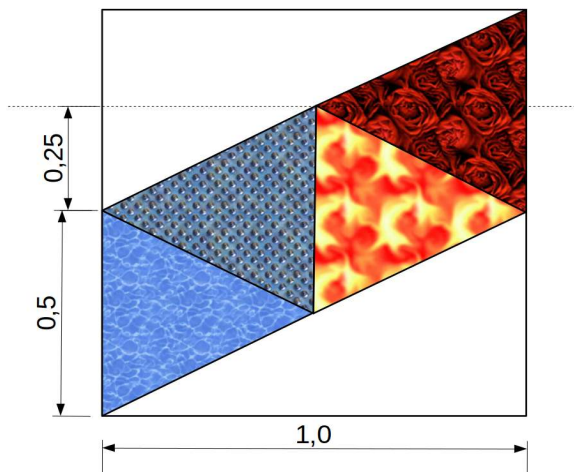
- a) Que fuentes de luz y que propiedades de material se han usado para generarla.
b) Que cambios se deben hacer en el código para que la visualización simule una esfera, indicando como se calcularían las normales.



2. (3 puntos) Diseñar el grafo de escena para la escalera de la figura (tres grados de libertad). Se creará a partir de un cubo de lado unidad alineado con los ejes de coordenadas con el centro de la cara inferior en el origen de coordenadas. Las cuatro piezas (A,B, C y D) son iguales.



3. (2 puntos) Escribe el código en C++ con OpenGL que dibuje un tetraedro con lados de tamaño unidad aplicándole la textura que se muestra en la figura. Anotar las coordenadas usadas para los vértices del tetraedro en la figura de la derecha.



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



4. (2 puntos) Sea la siguiente representación de una malla de triángulos. Describe cómo se puede determinar si la orientación de las normales de las caras es consistente (están orientadas hacia el exterior del objeto).

```
class _puntos3D
{
...
vector<_vertex3f> vertices;
};

class _triangulos3D: public _puntos3D
{
    _triangulos3D();
    ...
    void  calcular_normales_caras();
    vector<_vertex3i> caras;
    vector<_vertex3f> normales_caras;
}
```

5. (2 puntos) Explica como se construye un volumen de vista en perspectiva señalando los parámetros necesarios. Para OpenGL indica las distintas formas de realizar un zoom para una cámara en perspectiva (glFrustum(...)) y además, como orientar este tipo de cámara para tener un plano de proyección picado (oblicuo a 45 grados respecto al objeto enfocado).

WUOLAH