

Examen Ordinaria 21-22 Resuelto.pdf



DonOreo



Informática Gráfica



3º Grado en Ingeniería Informática



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



Escuela de
organización
Industrial

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)



Examen Ordinaria 2021-22

☰ Resumen	
📅 Fecha	
☰ Status	Terminado

1. (2 puntos) Explique, lo más detallado que pueda, las distintas formas de realizar un pick en OpenGL

Esta búsqueda se puede realizar de varios modos:

- Usando el buffer de selección (OpenGL hasta versión 3): Devuelve los identificadores de los objetos incluidos en un subvolumen de visión centrado en una posición del viewport.
- Intersección rayo escena:
 1. Se dibuja la escena con iluminación desactivada usando como colores los identificadores de los objetos en un frame buffer no visible.
 2. Se lee el pixel que corresponde a la posición del cursor.
 3. Se decodifica el color para obtener el identificador.
- Codificando el id de objeto como color y leyendo el frame buffer:
 1. Creando un objeto OpenGL de tipo frame-buffer object (FBO), y haciendo rasterización con ese objeto como imagen de destino (rendering target).
 2. Usando el modo de doble buffer.

En este modo siempre existen dos FBOs creados por OpenGL: un buffer trasero (back buffer), que es donde se visualizan las primitivas, y un buffer delantero (front buffer), que es el que se visualiza en pantalla.

En este caso, para evitar que la imagen generada para seleccionar sea visible se debe redibujar la escena después de seleccionar sin intercambiar los buffers (sin llamar a `glutSwapBuffers()`).

```
int pick(int x, int y, int * i, int * parte)
{
    GLint viewport[4];
    unsigned char data[4];

    glGetIntegerv(GL_VIEWPORT, viewport);
    glDisable(GL_DITHER);
    glDisable(GL_LIGHTING);
    dibujoEscena();
    glEnable(GL_LIGHTING);
    glEnable(GL_DITHER);
    glFlush();
    glFinish();

    glReadPixels(x, viewport[3] - y, 1, 1, GL_RGBA, GL_UNSIGNED_BYTE, data);

    *i=data[0];
    *parte=data[1];

    dibujoEscena();
    return *i;
}
```

- Recortado (clipping): no hay info 😊

2. (2 puntos) Describa el flujo de transformaciones que se realizan en OpenGL desde que proporcionamos las coordenadas 3D de un modelo hasta que tenemos una imagen

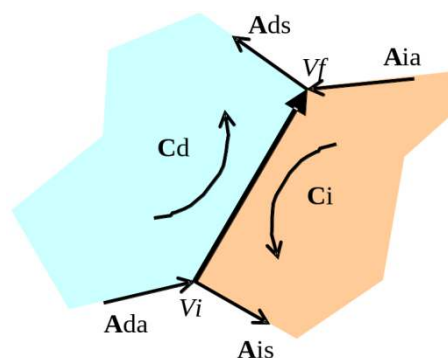
Consulta
condiciones aquí



do your thing

en pantalla. Indique el propósito de cada etapa y el resultado obtenido tras cada una de las transformaciones.

1. Transformación del modelo: Situarlo en escena, cambiarlo de tamaño y crear modelos compuestos de otros más simples.
 2. Transformación de vista: Poner al observador en la posición deseada
 3. Transformación de perspectiva: pasar de un mundo 3D a una imagen 2D.
 4. Rasterización: Calcular para cada pixel su color, teniendo en cuenta la primitiva que se muestra, su color, material, texturas, luces, etc.
 5. Transformación del dispositivo: Adaptar la imagen 2D a la zona de dibujado.
3. (2 puntos) Describe la estructura de aristas aladas, y explica (apoyando la explicación con pseudocódigo si lo ves conveniente) el proceso que se debe seguir para construir una estructura de aristas aladas a partir de un archivo PLY



Podemos representar los objetos como:

- Sólido: Lista de aristas
- Cada arista contiene: 2 vértices y 2 caras (arista anterior y posterior en cada una)

Vi: Vértice inicial de la arista

Vf: Vértice final de la arista

Cd: Cara que recorre la arista al derecho

Ci: Cara que recorre la arista invertida

Ada: Arista anterior según Cd

Ads: Arista siguiente según Cd

Aia: Arista anterior según Ci

Ais: Arista siguiente según Ci

Aristas	Vi	Vf	Cd	Ci	Ada	Ads	Aia
---------	----	----	----	----	-----	-----	-----

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

MUOLAH

- 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



Creación a partir de PLY

```
ply
format ascii 1.0
element vertex 510272
property float32 x
property float32 y
property float32 z
element face 108176
property list uint8 int32 vertex_indices
end_header
0.971510 -1.341210 0.568620
0.982640 -1.344680 0.575240
0.989330 -1.341980 0.579250
.....
0.044910 5.677040 12.747991
3 57004 114239 114246
3 57003 57004 114239
3 57003 114232 114239
3 57002 57003 114232
.....
3 57002 114225 114232
3 57001 57002 114225
```

(Comprobar que la cara no existe)
 Añadir a la tabla de caras
 Incrementar número de caras
 Para cada arista
 Buscarla
 Añadir información de cara
 Enlazar aristas

Informática Gráfica, Grupo C, 2022/23 Página: 91/96

Creación a partir de PLY

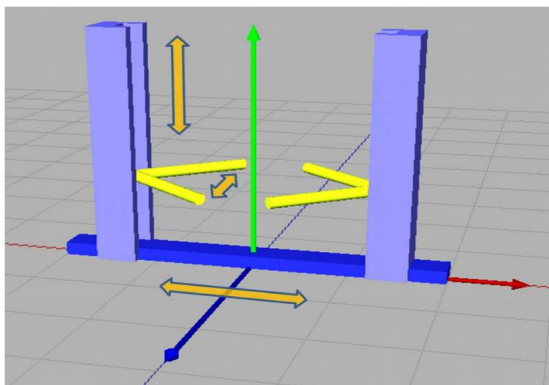
```
ply
format ascii 1.0
element vertex 510272
property float32 x
property float32 y
property float32 z
element face 108176
property list uint8 int32 vertex_indices
end_header
0.971510 -1.341210 0.568620
0.982640 -1.344680 0.575240
0.989330 -1.341980 0.579250
.....
0.044910 5.677040 12.747991
3 57004 114239 114246
3 57003 57004 114239
3 57003 114232 114239
3 57002 57003 114232
.....
3 57002 114225 114232
3 57001 57002 114225
```

(Comprobar que el vértice no existe)
 Añadir a la tabla de vértices
 Incrementar número de vértices

Informática Gráfica, Grupo C, 2022/23 Página: 90/96

4. (2 puntos) Obtenga el grafo de escena, incluyendo las transformaciones, de modo que partiendo de los objetos básicos cubo unidad y de cilindro unidad centrados en el origen, se pueda obtener un modelo de un elevador para automóviles (ver imágenes de abajo). Haga dibujos del posicionamiento y dimensiones de las piezas como apoyo.

Hay tres grados de libertad: movimiento horizontal de las barras verticales (azules), movimiento vertical y cierre y apertura de los soportes (cilindros amarillos).



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

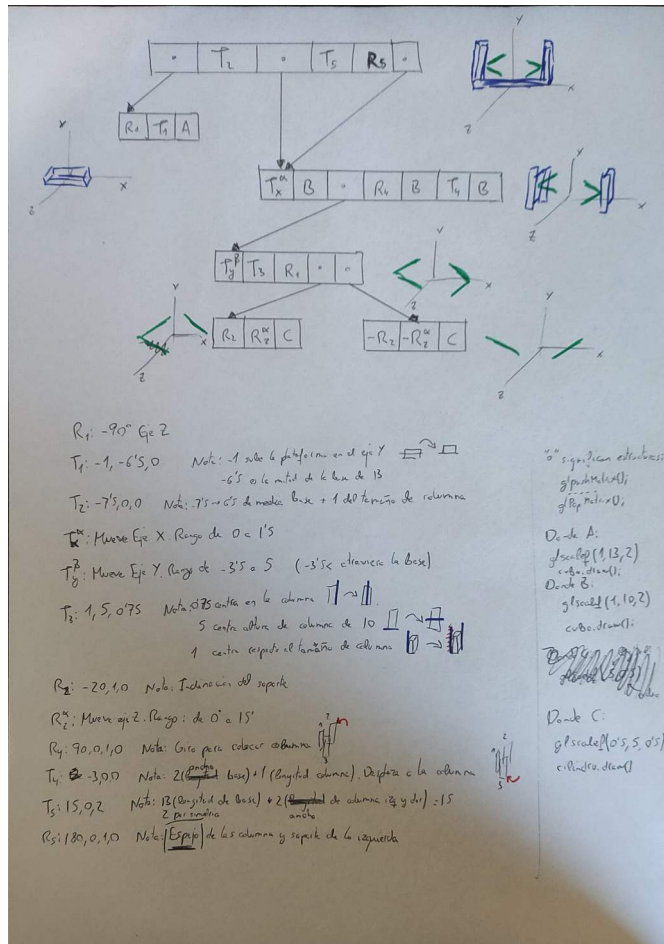
Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 5/5 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)



Ejecución:

<https://drive.google.com/file/d/1EV8CjhLLeeExJod-EV7jFeERW-d1u3a0/view?usp=sharing>

Código:

```
materialortoedro(light_blue);
glPushMatrix();
glRotatef(-90, 0, 0, 1); //R1
glTranslatef(-1, -6.5, 0); //T1
ortoedro(0.5, 0, 1, 0.5, 13, 1, 2); //A(Base) Equivale a un cubo unidad escalado 1x13x2
glPopMatrix();

glTranslatef(-7.5, 0, 0); //T2
columnasysoportes();

glTranslatef(15, 0, 2); //T5
glRotatef(180, 0, 1, 0); //R5
columnasysoportes();
```

Examen Ordinaria 2021-22

4

WUOLAH

```

int colsigno = 1, rsopsigno = 1, ysopsigno = 1;
float xcolumna = 0, rsoporte = 20, ysoporte = 0;
//Codigo de elevador
//Dibuja 3 columnas en forma de U y sus 2 soportes(cilindros)
void columnasysoportes(){
    glPushMatrix();
    glTranslatef(xcolumna,0,0); //Movimiento columna Talpa
    ortoedro(0.5,0,1,0.5,10,1,1,2); //B (Columna frontal) Equivale a un cubo unidad escalado 1x10x2

    glPushMatrix();
    materialOrtoedro(light_yellow);
    glTranslatef(0,ysoporte,0); //Movimiento soportes en y (Tbeta)
    glTranslatef(1,5,0.75); //T3
    glRotatef(-90,0,0,1); //R1
    glPushMatrix();
    glRotatef(-20,1,0,0); //R2
    glRotatef(rsoporte,1,0,0); //Rotacion soporte
    cilindro(0.25,0,0.25,0.25,5,0.25,0.5,0.5); //C(soporte izquierdo) Equivale a un cilindro unidad escalado 0.5x5x0.5
    glPopMatrix();
    glPushMatrix();
    glRotatef(20,1,0,0); //R3
    glRotatef(-rsoporte,1,0,0); //Rotacion soporte
    cilindro(0.25,0,0.25,0.25,5,0.25,0.5,0.5); //C(soporte derecho) Equivale a un cilindro unidad escalado 0.5x5x0.5
    glPopMatrix();
    materialOrtoedro(light_blue);
    glPopMatrix();

    glRotatef(90,0,1,0); //R4
    ortoedro(0.5,0,1,0.5,10,1,1,2); //B(columna lateral 1) Equivale a un cubo unidad escalado 1x10x2
    glTranslatef(-3,0,0); //T4
    ortoedro(0.5,0,1,0.5,10,1,1,2); //B(columna lateral 2) Equivale a un cubo unidad escalado 1x10x2
    glPopMatrix();
}

```

```

void idle(int v)
{
    //elevador
    xcolumna += 0.01*colsigno;
    if(xcolumna > 1.5)
        colsigno = -1;
    else if(xcolumna < 0)
        colsigno = 1;

    ysoporte += 0.1*ysopsigno;
    if(ysoporte > 5)
        ysopsigno = -1;
    else if(ysoporte < -3.5)
        ysopsigno = 1;

    rsoporte += 0.5*rsopsigno;
    if(rsoporte > 15)
        rsopsigno = -1;
    else if(rsoporte < 0)
        rsopsigno = 1;
}

```