UNIVERSIDAD
DE GRANADA

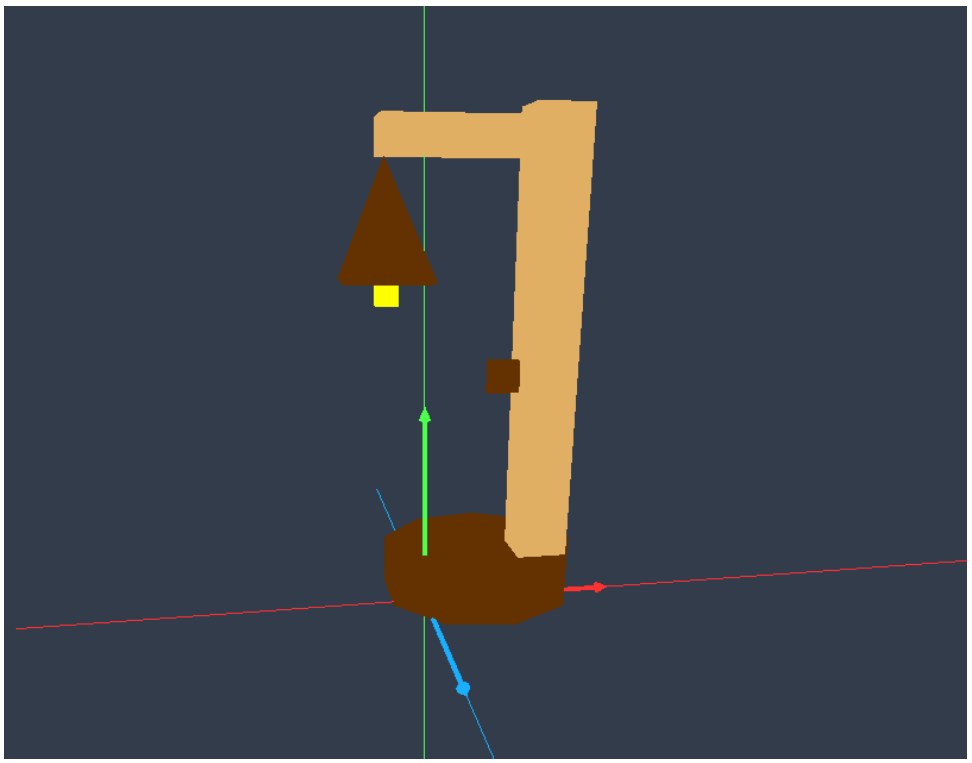
Isabel Morro Tabares
79095945A
Informática Gráfica
Curso Académico 2024/2025
Doble Grado de Ingeniería Informática y Matemáticas

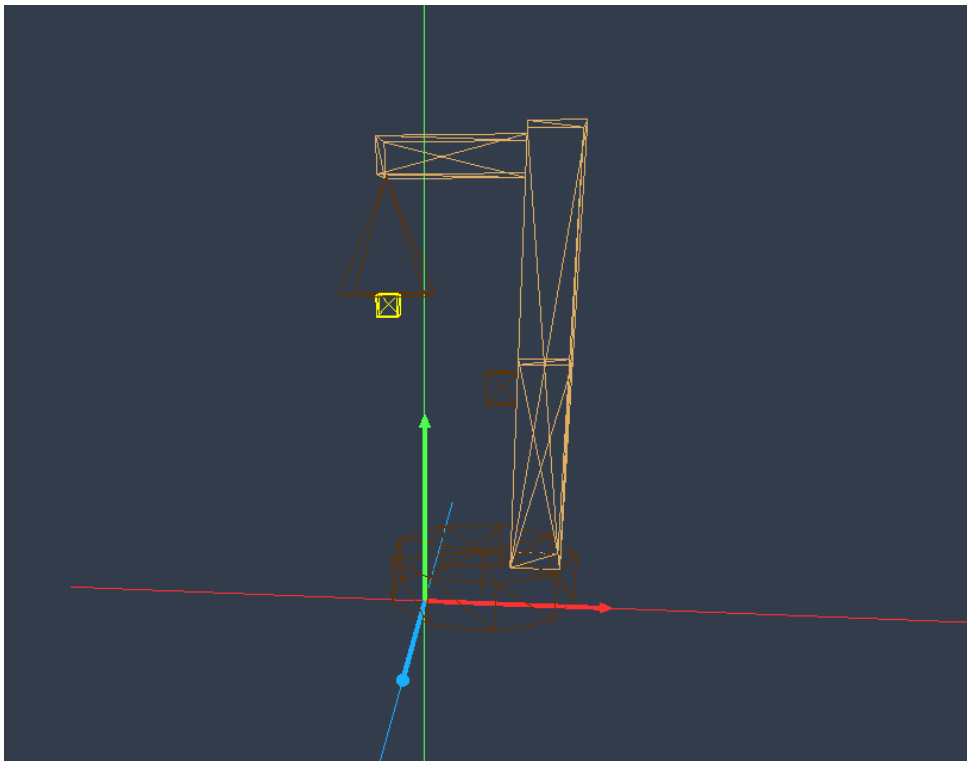
Memoria de Prácticas

ÍNDICE

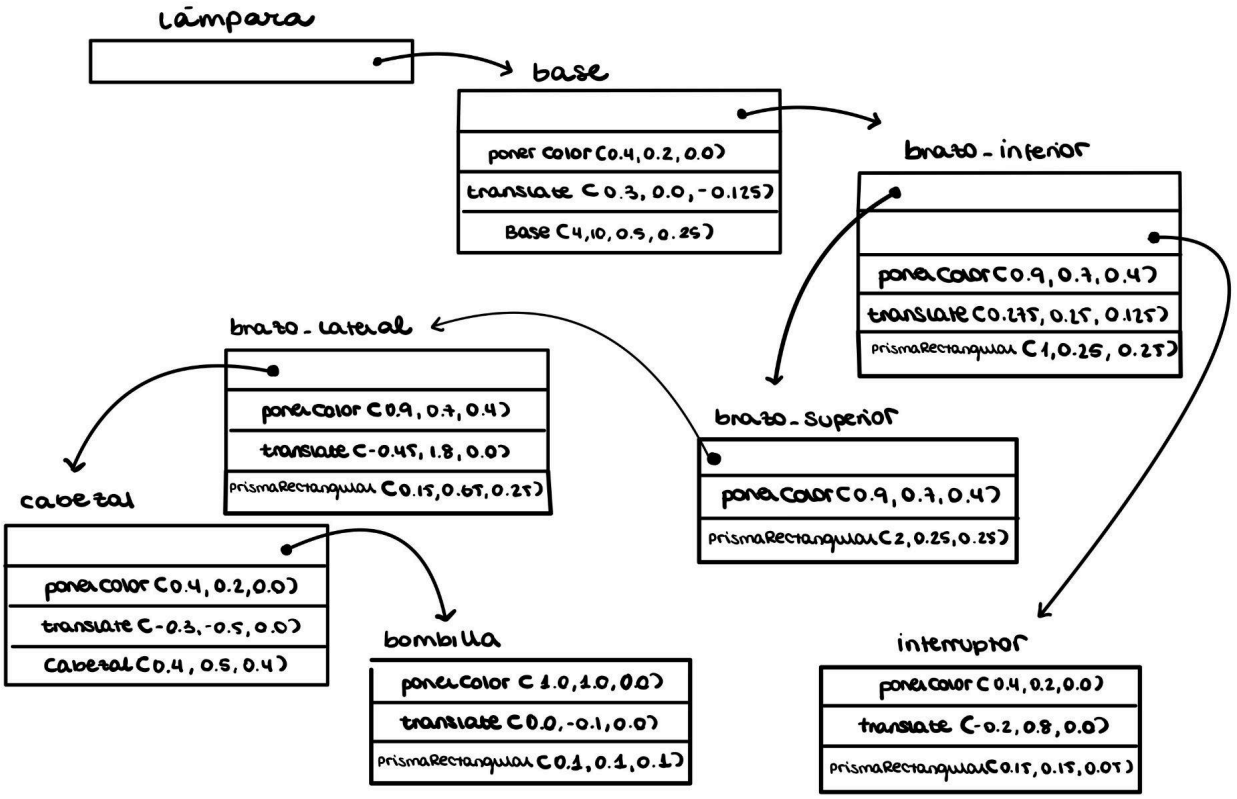
Práctica 3

1. Foto del modelo





2. Grafo de escena tipo PHIGS



3. Nodos del grafo

3.1. Base

La *base* es una instancia de la clase Base, se trata de un cilindro construido por revolución con color marrón oscuro representado por la terna RGB (0.4, 0.2, 0.0). Su identificador es el 1. La clase Base está definida en el archivo modelo-jer.h y su constructor implementado en el archivo modelo-jer.cpp, en concreto de la línea 13 a la 32.

3.2. Brazo inferior

El *brazo_inferior* es una instancia de la clase PrismaRectangular, se trata de un prisma de 8 vértices construido por mallas indexadas con color marrón claro representado por la terna RGB (0.9, 0.7, 0.4). Su identificador es el 2. La clase PrismaRectangular está definida en el archivo modelo-jer.h y su constructor implementado en el archivo modelo-jer.cpp, en concreto de la línea 69 a la 114.

3.3. Brazo superior

El *brazo_superior* es una instancia de la clase PrismaRectangular, se trata de un prisma de 8 vértices construido por mallas indexadas con color marrón claro representado por la terna RGB (0.9, 0.7, 0.4). Su identificador es el 3. La clase PrismaRectangular está definida en el archivo modelo-jer.h y su constructor implementado en el archivo modelo-jer.cpp, en concreto de la línea 69 a la 114.

3.4. Brazo Lateral

El *brazo_lateral* es una instancia de la clase PrismaRectangular, se trata de un prisma de 8 vértices construido por mallas indexadas con color marrón claro representado por la terna RGB (0.9, 0.7, 0.4). Su identificador es el 4. La clase PrismaRectangular está definida en el archivo modelo-jer.h y su constructor implementado en el archivo modelo-jer.cpp, en concreto de la línea 69 a la 114.

Este tiene el parámetro de libertad **m_traslacion_brazo_lateral*.

3.5. Cabezal

El *cabezal* es una instancia de la clase Cabezal, se trata de un tetraedro construido por mallas indexadas con color marrón oscuro representado por la terna RGB (0.4, 0.2, 0.0). Su identificador es el 5. La clase Cabezal está definida en el archivo modelo-jer.h y su constructor implementado en el archivo modelo-jer.cpp, en concreto de la línea 37 a la 64

Este tiene el parámetro de libertad **m_rotacion_cabezal*.

3.6. Bombilla

La *bombilla* es una instancia de la clase PrismaRectangular, se trata de un prisma de 8 vértices construido por mallas indexadas con color amarillo representado por la terna RGB (1.0, 1.0, 0.0). Su identificador es el 6. La clase PrismaRectangular está definida en el archivo modelo-jer.h y su constructor implementado en el archivo modelo-jer.cpp, en concreto de la línea 69 a la 114.

Este tiene el parámetro de libertad **m_scale_bombilla*.

3.7. Interruptor

El *interruptor* es una instancia de la clase PrismaRectangular, se trata de un prisma de 8 vértices construido por mallas indexadas con color marrón oscuro representado por la terna RGB (0.4, 0.2, 0.0). Su identificador es el 7. La clase PrismaRectangular está definida en el archivo modelo-jer.h y su constructor implementado en el archivo modelo-jer.cpp, en concreto de la línea 69 a la 114.

3.8. Lámpara

La *lampara* es una instancia de la clase Lámpara. La clase Lámpara está definida en el archivo modelo-jer.h y su constructor implementado en el archivo modelo-jer.cpp, en concreto de la línea 118 a la 230. Este nodo agrupa todos los anteriores.

Este tiene el parámetro de libertad **m_traslacion_lampara*.

4. Grados de libertad

4.1. m_traslacion_brazo_lateral

El puntero **m_traslacion_brazo_lateral* de índice 0 se encuentra en el nodo *brazo_lateral*. Esta matriz permite trasladar de manera oscilante el brazo lateral a lo largo del eje Y, en particular, en el rango [Y_posición_original - 0.5, Y_posición_original]. La expresión que constituye la matriz a partir del tiempo es:

```
float A = -0.5;
float B = 0.0;
translate(vec3(0.0, A + ((B-A)/2)*(1 + sin((M_PI/2)*t_sec)), 0.0));
```

4.2. m_rotacion_cabezal

El puntero **m_rotacion_cabezal* de índice 1 se encuentra en el nodo *cabezal*. Esta matriz permite rotar el cabezal de la lámpara en torno al eje Y. La expresión que constituye la matriz a partir del tiempo es:

```
float angulo = 2*(M_PI/4)*t_sec;
*m_rotacion_cabezal = translate(vec3(-0.3, -0.5, 0.0))*
                      rotate(angulo, vec3(0.0, 1.0, 0.0))*
                      translate(vec3(0.3, 0.5, 0.0));
```

4.3. m_scale_bombilla

El puntero **m_scale_bombilla* de índice 2 se encuentra en el nodo *bombilla*. Esta matriz permite escalar el objeto, aumentando y disminuyendo su tamaño de forma oscilante. La expresión que constituye la matriz a partir del tiempo es:

```
float A  = 0.75;
float B = 1.4;
float escala = A + ((B-A)/2)*( 1 + sin((M_PI/2)*t_sec));
*m_scale_bombilla = scale (vec3(escala, escala, escala));
```

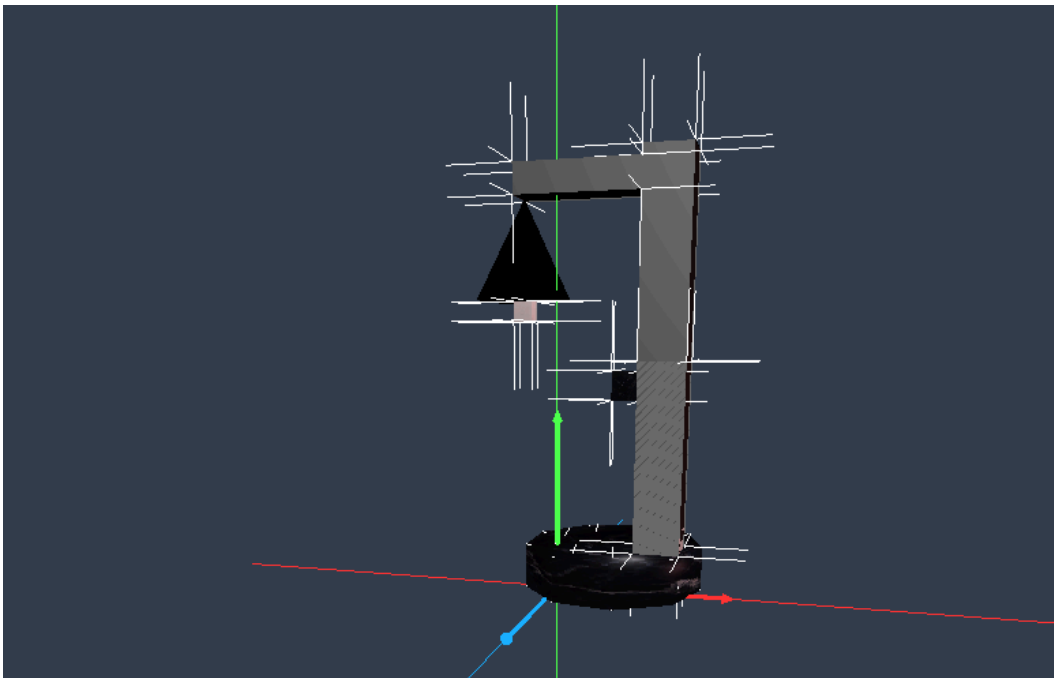
4.4. m_traslacion_lampara

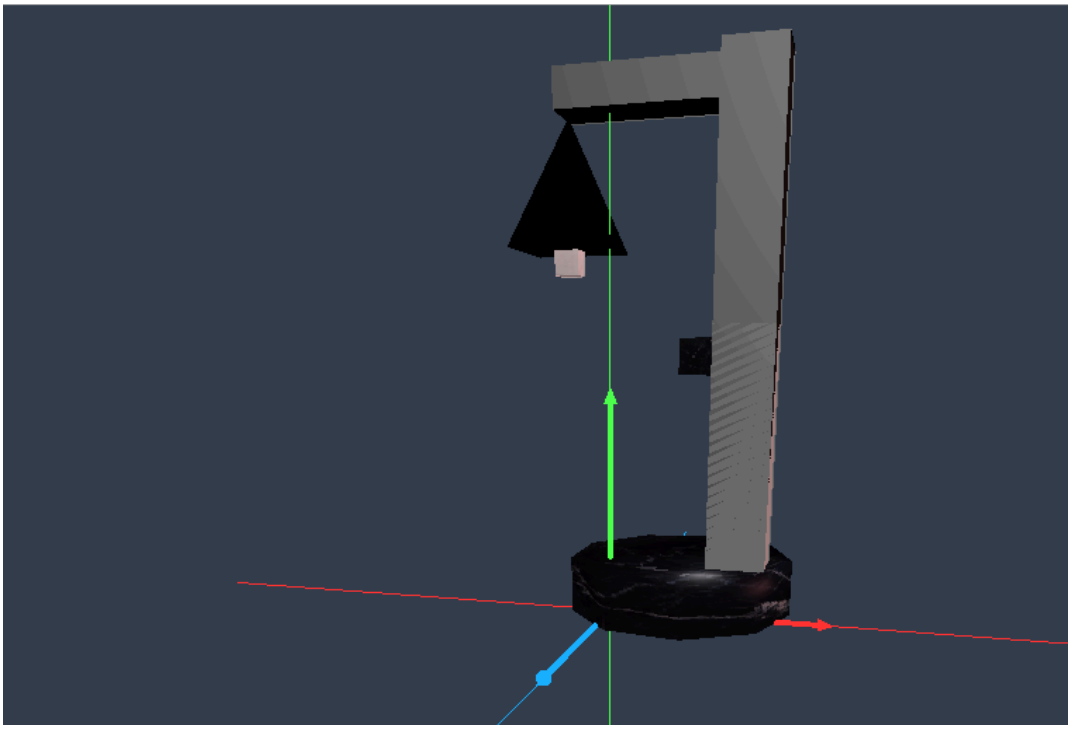
El puntero **m_traslacion_lampara* de índice 3 se encuentra en el nodo *lampara*. Esta matriz permite trasladar de manera oscilante el objeto completo de manera horizontal a lo largo del eje X, en particular, en el rango [X_posición_original, X_posición_original + 1]. La expresión que constituye la matriz a partir del tiempo es:

```
float A = 0.0;
float B = 1.0;
*m_traslacion_lampara=translate(vec3(A+((B-A)/2)*(1+sin((M_PI/2)*t_sec)),0,0));
```

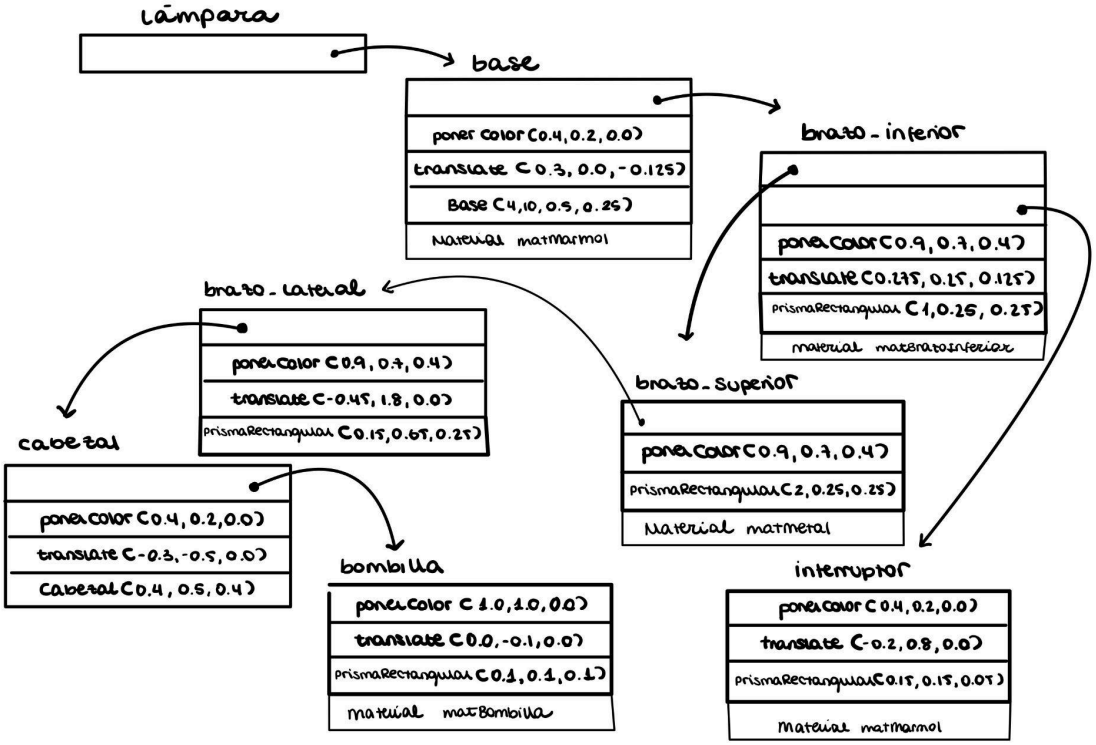
Práctica 4 - Materiales y Texturas

1. Foto del modelo





2. Grafo de escena tipo PHIGS



3. Materiales

3.1. matMarmol

El material *matMarmol* se usa tanto en el nodo *base* como en el nodo *interruptor*.
Los coeficientes que caracterizan este material son:

- *p_k_amb = 0.5*
- *p_k_dif = 0.6*
- *p_k_pse = 0.5*
- *p_exp_pse = 50.0*

Dicho material tiene asociada la textura *textMarmol* tomada de la imagen *text-marmol-negro.jpg*.

3.2. matBrazoInferior

El material *matBrazoinferior* se usa en el nodo *brazo_inferior*, y se trata de un material difuso.
Los coeficientes que caracterizan este material son:

- *p_k_amb = 0.0*
- *p_k_dif = 1.0*
- *p_k_pse = 0.5*
- *p_exp_pse = 50.0*

Dicho material no tiene una textura asociada.

3.3. matMetal

El material *matMetal* se usa en el nodo *brazo_superior*, y se trata de un material pseudo-especular.
Los coeficientes que caracterizan este material son:

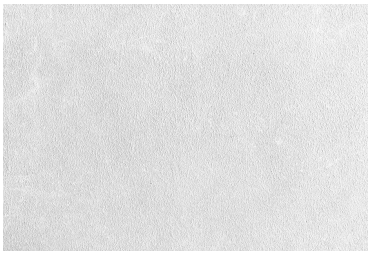
- *p_k_amb = 0.0*
- *p_k_dif = 0.0*
- *p_k_pse = 1.0*
- *p_exp_pse = 5.0*

Dicho material tiene asociada la textura *textMetalOscuro* tomada de la imagen *text-metalico-oscuro.jpg*. Cuyos vectores de coeficientes asociados son

coefs_s[4] = {1.0,0.0,0.0,0.0}
coefs_t[4] = {0.0,1.0,0.0,0.0}

3.4. matBombilla

El material *matBombilla* se usa en el nodo *bombilla*. Los coeficientes que caracterizan este material son:

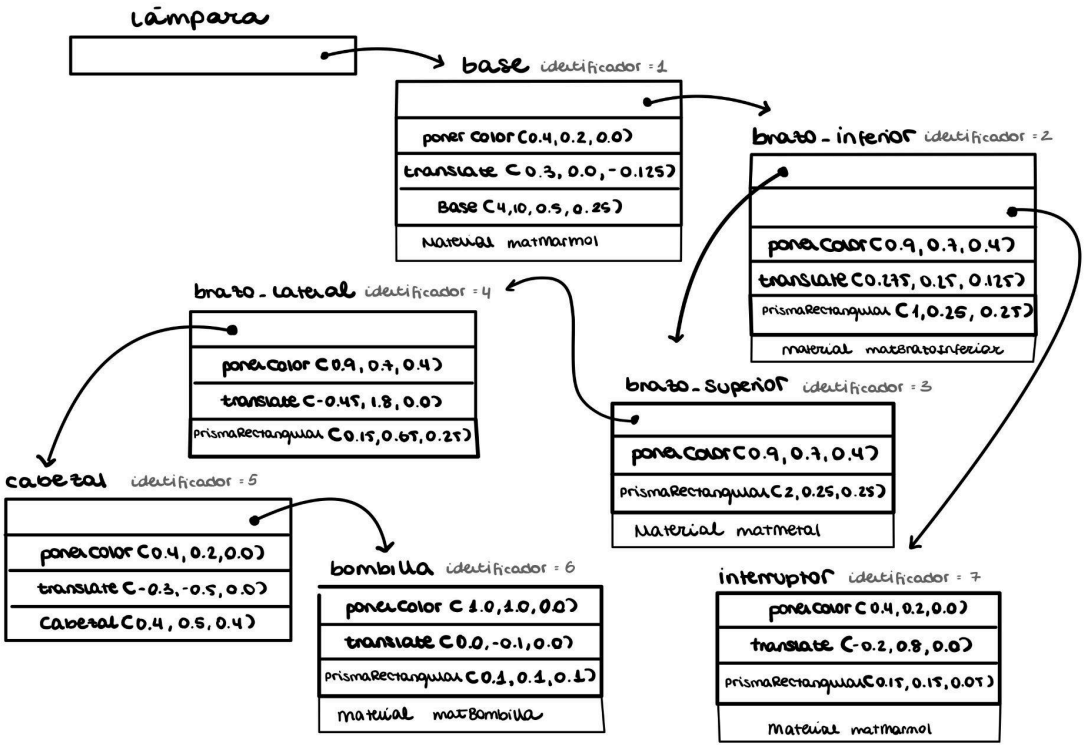


- *p_k_amb = 0.5*
- *p_k_dif = 0.6*
- *p_k_pse = 0.6*
- *p_exp_pse = 50.0*

Dicho material tiene asociada la textura *textPlasticoBlanco* tomada de la imagen *text-plastico-blanco.jpg*.

Práctica 5 - Identificadores de Selección

1. Grafo de escena tipo PHIGS



2. Identificadores de selección

Para el diseño de mi modelo jerárquico he decidido ir asignando los identificadores de manera creciente a medida que definía los nodos del grafo en el constructor de mi objeto, la lámpara. Para ello, en el comienzo del constructor delfino

unsigned identificador = 1.

Los trozos de código mostrados a continuación se encuentran en el mismo orden en el código, por ello al incrementar la variable *identificador* obtenemos los valores descritos para cada nodo. Todos en el archivo *modelo-jer.cpp* y en el constructor de la clase *Lampara.*

2.1. Base

La *base* tiene como identificador el valor 1.

```
NodoGrafoEscena *base = new NodoGrafoEscena();
base->ponerNombre("Base de la lámpara");
base->ponerIdentificador(identificador); // 1
identificador++;
```

2.2. Brazo inferior

El *brazo inferior* tiene como identificador el valor 2.

```
NodoGrafoEscena *brazo_inferior = new NodoGrafoEscena();
brazo_inferior->ponerNombre("Brazo inferior de la lámpara");
brazo_inferior->ponerIdentificador(identificador); // 2
identificador++;
```

2.3. Brazo superior

El *brazo superior* tiene como identificador el valor 3.

```
NodoGrafoEscena *brazo_superior = new NodoGrafoEscena();
brazo_superior->ponerNombre("Brazo superior de la lámpara");
brazo_superior->ponerIdentificador(identificador); // 3
identificador++;
```

2.4. Brazo lateral

El *brazo lateral* tiene como identificador el valor 4.

```
NodoGrafoEscena *brazo_lateral = new NodoGrafoEscena();
brazo_lateral->ponerNombre("Brazo lateral de la lámpara");
brazo_lateral->ponerIdentificador(identificador); // 4
identificador++;
```

2.5. Cabezal

El *cabezal* tiene como identificador el valor 5.

```
NodoGrafoEscena *cabezal = new NodoGrafoEscena();
cabezal->ponerNombre("Cabezal de la lámpara");
cabezal->ponerIdentificador(identificador); // 5
identificador++;
```

2.6. Bombilla

La *bombilla* tiene como identificador el valor 6.

```
NodoGrafoEscena *bombilla = new NodoGrafoEscena();
bombilla->ponerNombre("Bombilla de la lámpara");
bombilla->ponerIdentificador(identificador); // 6
identificador++;
```

2.7. Interruptor

El *interruptor* tiene como identificador el valor 7.

```
NodoGrafoEscena *interruptor = new NodoGrafoEscena();
interruptor->ponerNombre("Interruptor de la lámpara");
interruptor->ponerIdentificador(identificador); // 7
identificador++;
```