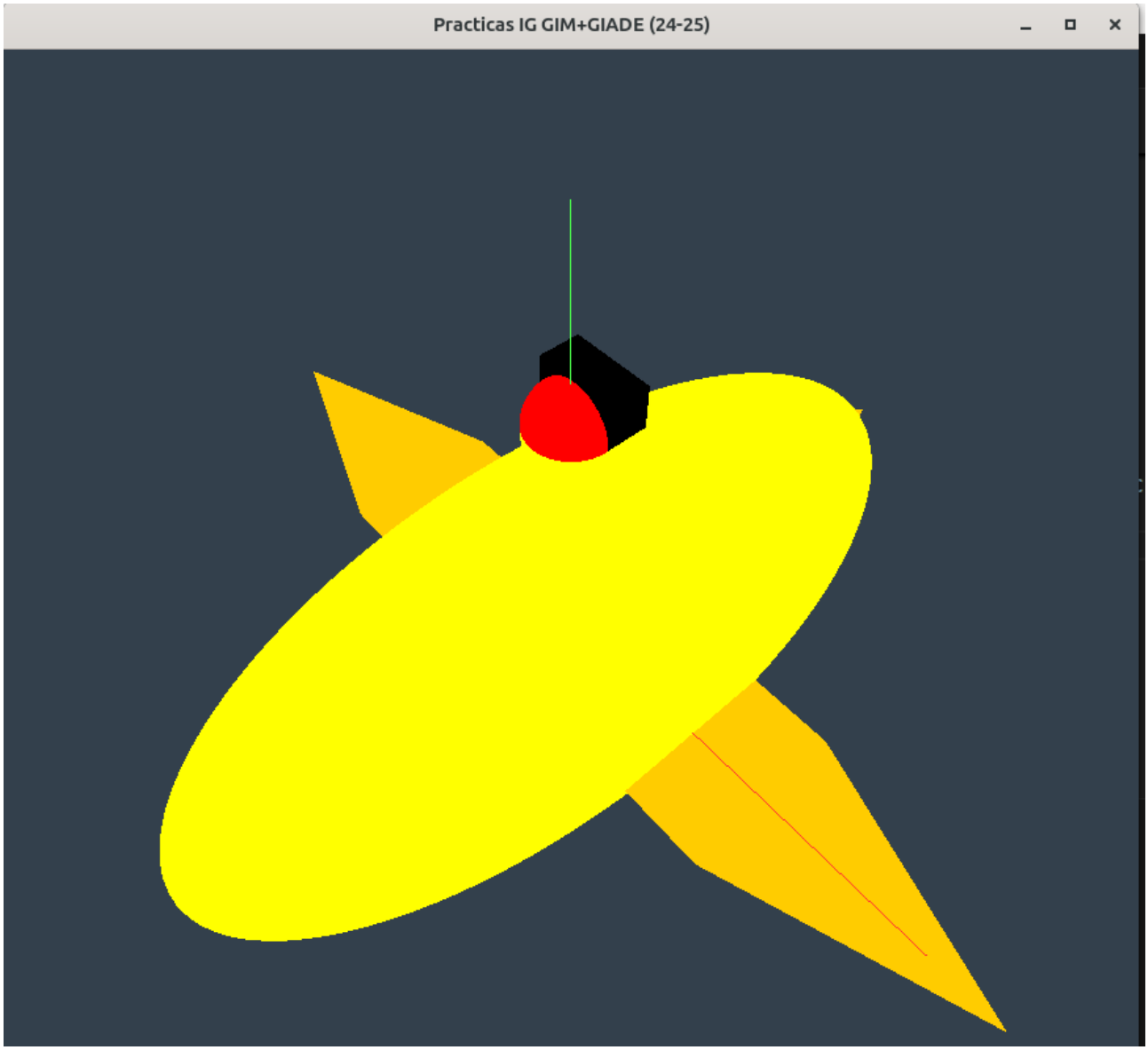
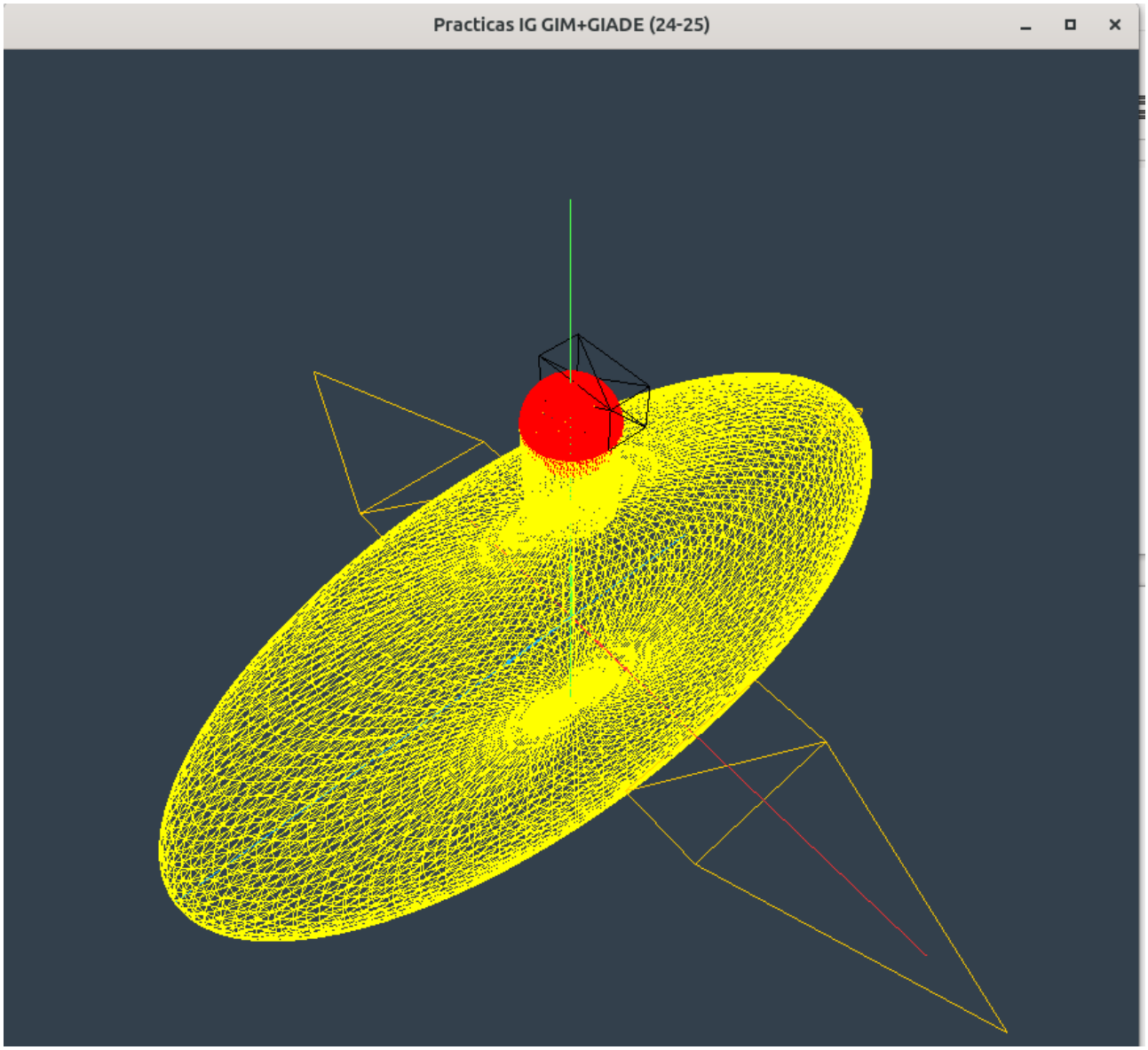
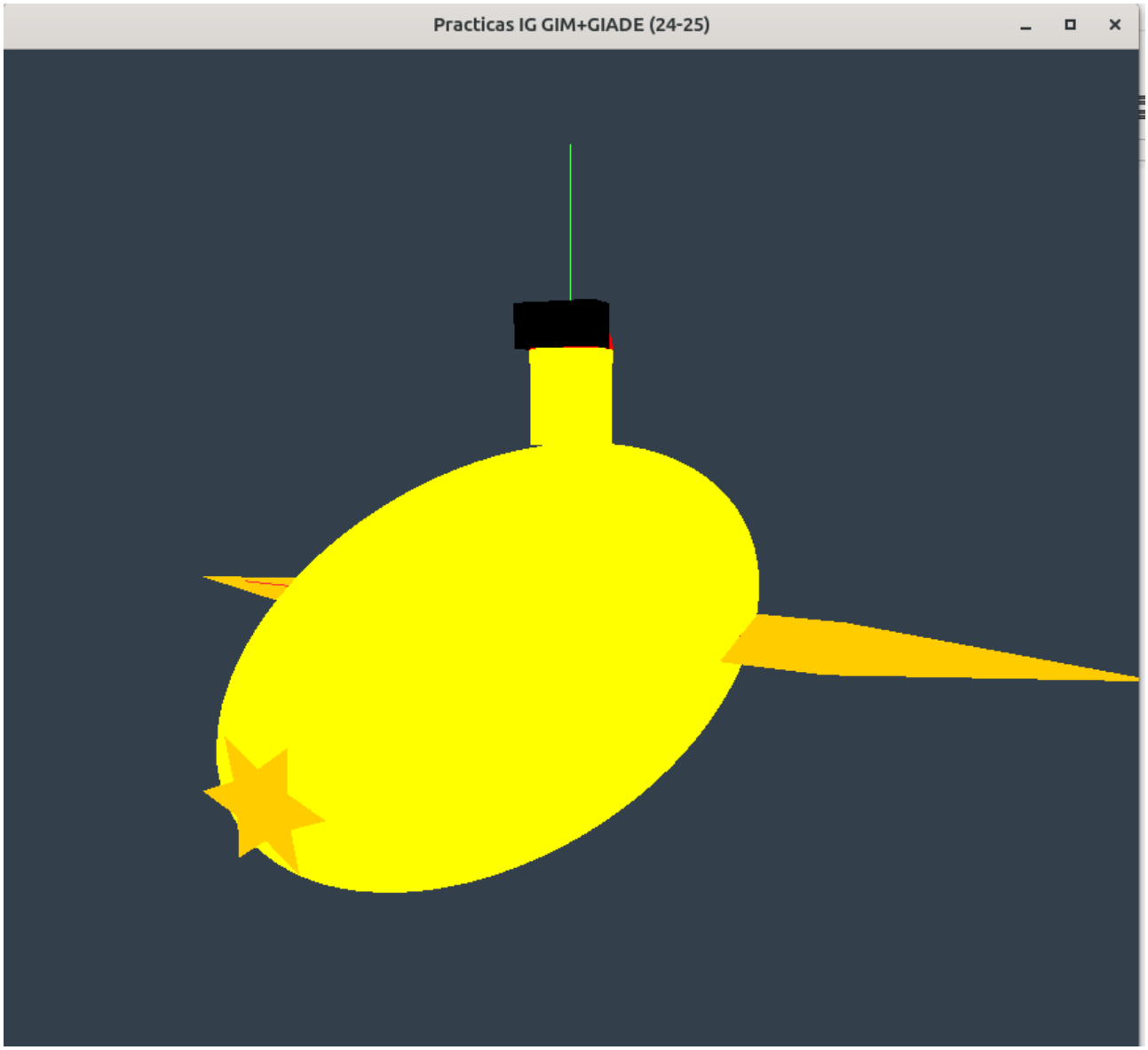
INFORMÁTICA GRAFICA

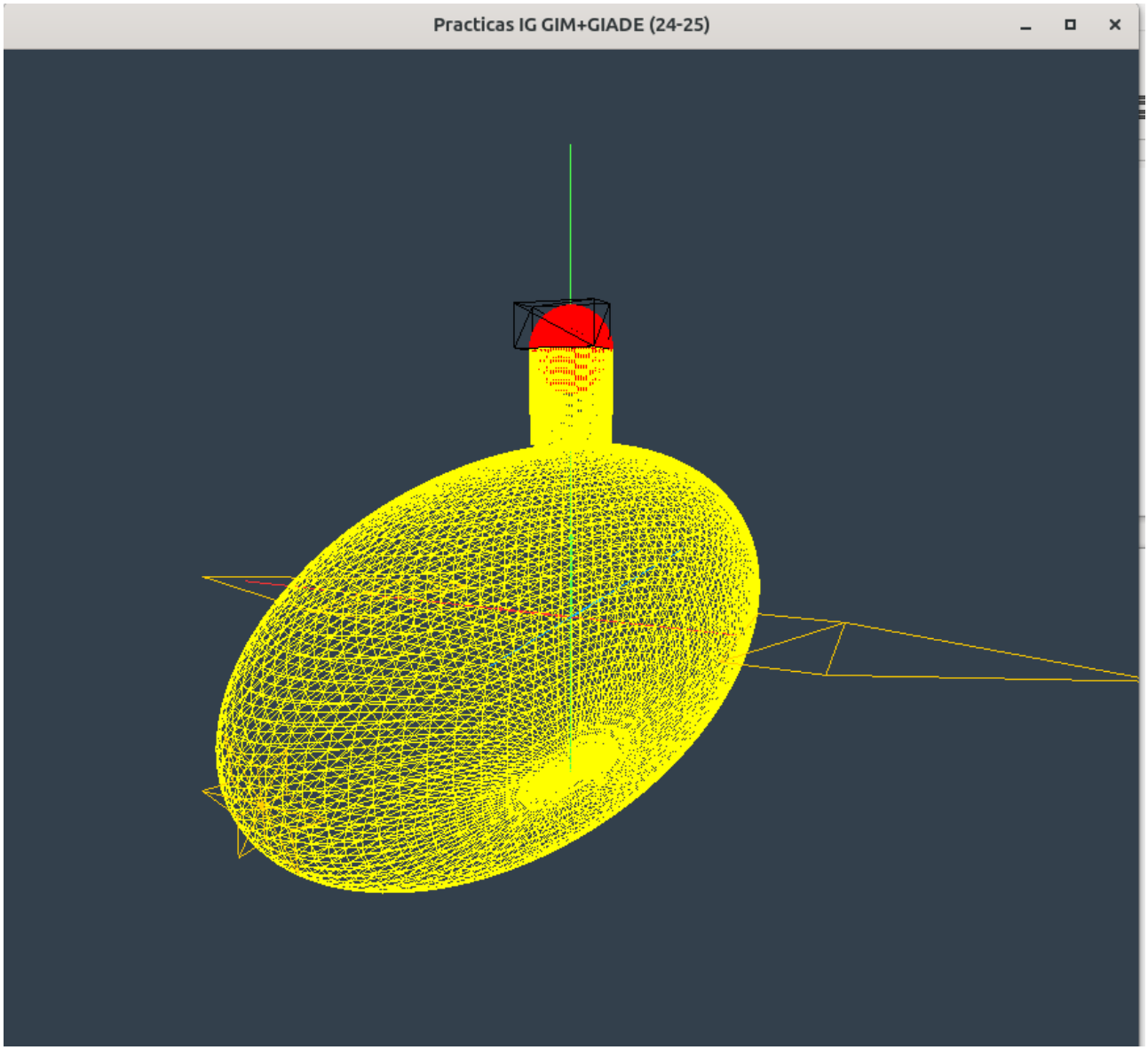
- Curso académico: 2024/2025
- Nombre y apellidos del autor: Jaime Corzo Galdó
- Titulación: Doble Grado en Ingeniería Informática y Matemáticas

1) Captura de pantalla del modelo:

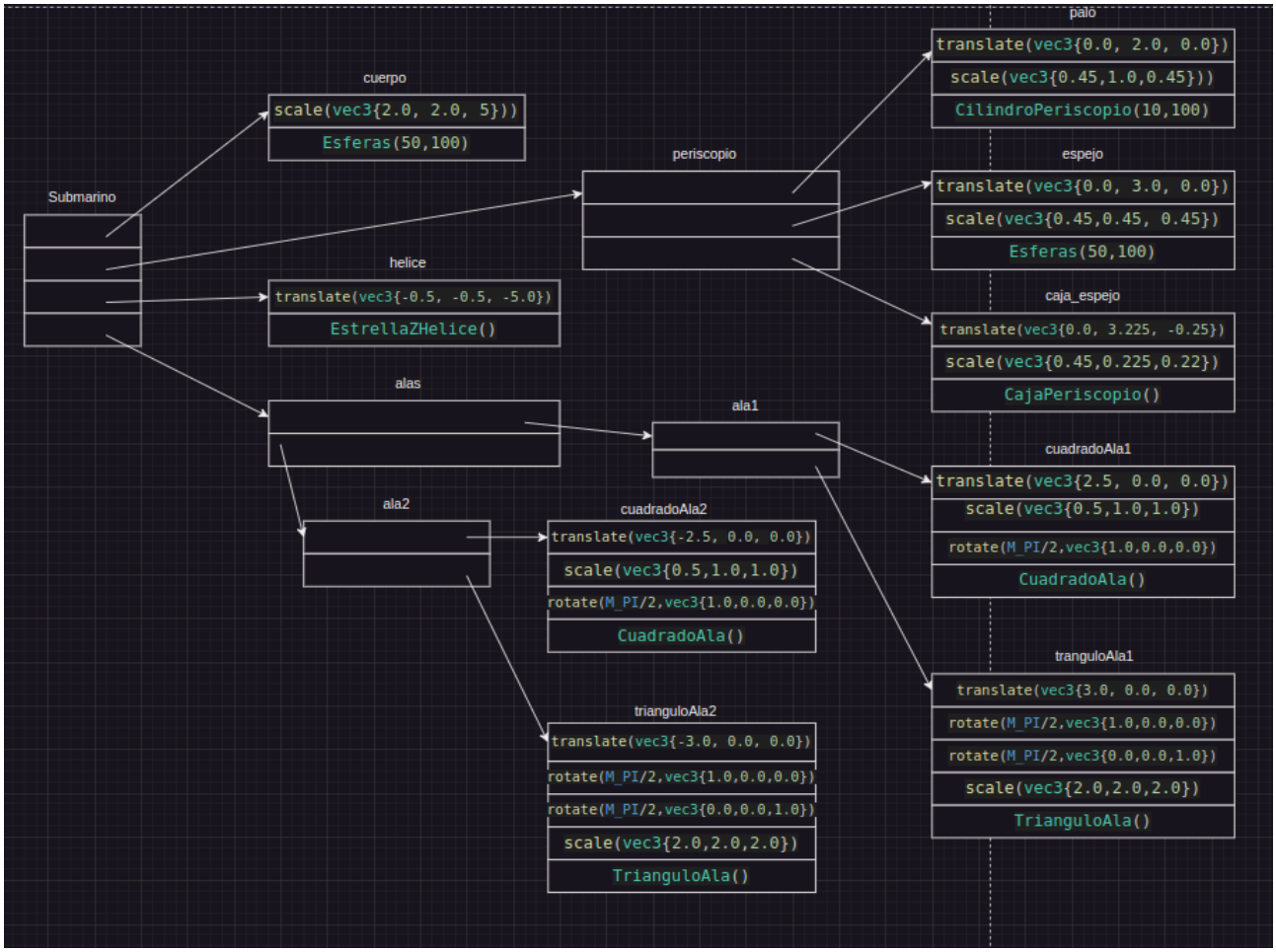

Practicas IG GIM+GIADE (24-25)


Practicas IG GIM+GIADE (24-25)


Practicas IG GIM+GIADE (24-25)



2) Grafo de escena tipo PHIGS:



3) Lista con información de todos y cada uno de los nodos del grafo

Todas las clases asociadas están declaradas en modelo-jer.h y modelo-jer.cpp

- cuerpo(objeto): No tiene asociado parámetros o grados de libertad. Color: amarillo → {1.0,1.0,0.0}

Rango de líneas en el .cpp donde se construye el nodo: 201-208.
Instancia de Esferas

- periscopio(objeto): No tiene asociado parámetros o grados de libertad.

Rango de líneas en el .cpp donde se construye el nodo: 211-252
Agrupa palo, espejo y caja_espejo

- palo(objeto): No tiene asociado parámetros o grados de libertad. Color: amarillo → {1.0,1.0,0.0}

Rango de líneas en el .cpp donde se construye el nodo: 214-222
Instancia de CilindroPeriscopio

- espejo(objeto): No tiene asociado parámetros o grados de libertad. Color: rojo → {1.0,0.0,0.0}

Rango de líneas en el .cpp donde se construye el nodo: 225-234
Instancia de Esferas

- caja_espejo(objeto): Tiene asociado un parámetro o grado de libertad:
  1. *pm_rot_espejo

  Color: negro → {0.0,0.0,0.0}
  Rango de líneas en el .cpp donde se construye el nodo: 237-247
  Instanca de CajaPeriscopio
- hélice(objeto): Tiene asociado un parámetro o grado de libertad:
  1. *pm_rot_helice

  Color: amarillo → {1.0,0.8,0.0}
  Rango de líneas en el .cpp donde se construye el nodo: 255-265
  Instancia de EstrellaZHelice
- alas(objeto): No tiene asociado parámetros o grados de libertad. Color: amarillo → {1.0,0.8,0.0}

Rango de líneas en el .cpp donde se construye el nodo: 268-333
Agrupa ala1 y ala2

- ala1(objeto): No tiene asociado parámetros o grados de libertad

Rango de líneas en el .cpp donde se construye el nodo: 271-298
Agrupa cuadradoAla1 y trianguloAla1

- cuadradoAla1(objeto): No tiene asociado parámetros o grados de libertad

Rango de líneas en el .cpp donde se construye el nodo: 272-281
Instancia de CuadradoAla

- trianguloAla1(objeto): Tiene asociado un parámetro o grado de libertad:
  1. *pm_tras_ala1

  Rango de líneas en el .cpp donde se construye el nodo: 283-294
  Instanciade trianguloAla
- ala2(objeto): No tiene asociado parámetros o grados de libertad.

Rango de líneas en el .cpp donde se construye el nodo: 301-330
Agrupa cuadradoAla2 y trianguloAla2

- cuandradoAla2(objeto): No tiene asociado parámetros o grados de libertad

Rango de líneas en el .cpp donde se construye el nodo: 302-312
Instancia de CuadradoAla

- trianguloAla2(objeto): Tiene asociado un parámetro o grado de libertad:
  1. *pm_tras_ala2

  Rango de líneas en el .cpp donde se construye el nodo: 314-326

Instancia de TrianguloAla

- submarino(objeto): Tiene asociado dos parámetros o grados de libertad:
  1. *pm_tras_cuerpo
  2. *pm_rot_cuerpo

  Rango de líneas en el .cpp donde se construye el nodo: 197-343
  Agrupa cuerpo, periscopio, helice y alas

4) Lista con información de todos y cada uno de los parámetros o grados de libertad del grafo

1) *pm_rot_espejo: Se encuentra en el nodo cabeza_espejo.
es un desplazamiento oscilante en el eje X, con un período de 2 segundos y una amplitud de 1.4 unidades de distancia, o bien rotación entorno al eje Y, con una frecuencia de 1/25 ciclos por segundo).

```
float a = 0.0;
float b = (2.0*M_PI)/25; //frecuencia = 1/25 ciclos por segundo
*pm_rot_espejo = rotate(a+(b*t_sec), vec3{0.0, 1.0, 0.0});
```

2) *pm_rot_helice: Se encuentra en el nodo helice.
Rotación entorno al eje Z, con una frencuencia de 1 ciclo por segundo

```
float a = 0.0;
float b = (2.0*M_PI); //frecuencia = 1 ciclos por segundo
*pm_rot_helice = rotate(a+(b*t_sec), vec3{0.0,0.0,1.0});
```

3) *pm_tras_ala1: Se encuentra en el nodo trianguloAla1.
Rotación oscilante en el eje que forma el lado exterior de cuadradoAla1, con una frecuencia de 1 ciclo por segundo. Está acotada entre -pi/6 y pi/6.

```
float vmin = -M_PI/6;  // -45 grados en radianes
float vmax = M_PI/6;   // 45 grados en radianes
float a_oscilante = (vmax + vmin)/2; // Punto medio = 0
float b_oscilante = (vmax - vmin)/2; // Amplitud = π/4
float n = 1.0f; // Frecuencia de oscilación
*pm_tras_ala1 = translate(vec3{3.0,0.0,0.0}) * rotate(float(a_oscilante+b_oscilante*sin(M_PI*n*t_sec)),
                vec3{0.0, 0.0, 1.0}) * translate(vec3{-3.0,0.0,0.0});
```

4) *pm_tras_ala2: Se encuentra en el nodo trianguloAla2.
Rotación oscilante en el eje que forma el lado exterior de cuadradoAla2, con una frecuencia de 1 ciclo por segundo. Está acotada entre -pi/6 y pi/6.

```
float vmin = -M_PI/6;  // -45 grados en radianes
float vmax = M_PI/6;   // 45 grados en radianes
float a_oscilante = (vmax + vmin)/2; // Punto medio = 0
float b_oscilante = (vmax - vmin)/2; // Amplitud = π/4
float n = 1.0f; // Frecuencia de oscilación
*pm_tras_ala2 = translate(vec3{-3.0,0.0,0.0}) * rotate(float(a_oscilante+b_oscilante*sin(-M_PI*n*t_sec)),
                vec3{0.0, 0.0, 1.0}) * translate(vec3{3.0,0.0,0.0});
```

5) *pm_tras_cuerpo: Se encuentra en el nodo submarino.
Es una traslación oscilante respecto al eje Z con frecuencia de 0.5 ciclos por segundo y acotada por -pi/4 y pi/4.

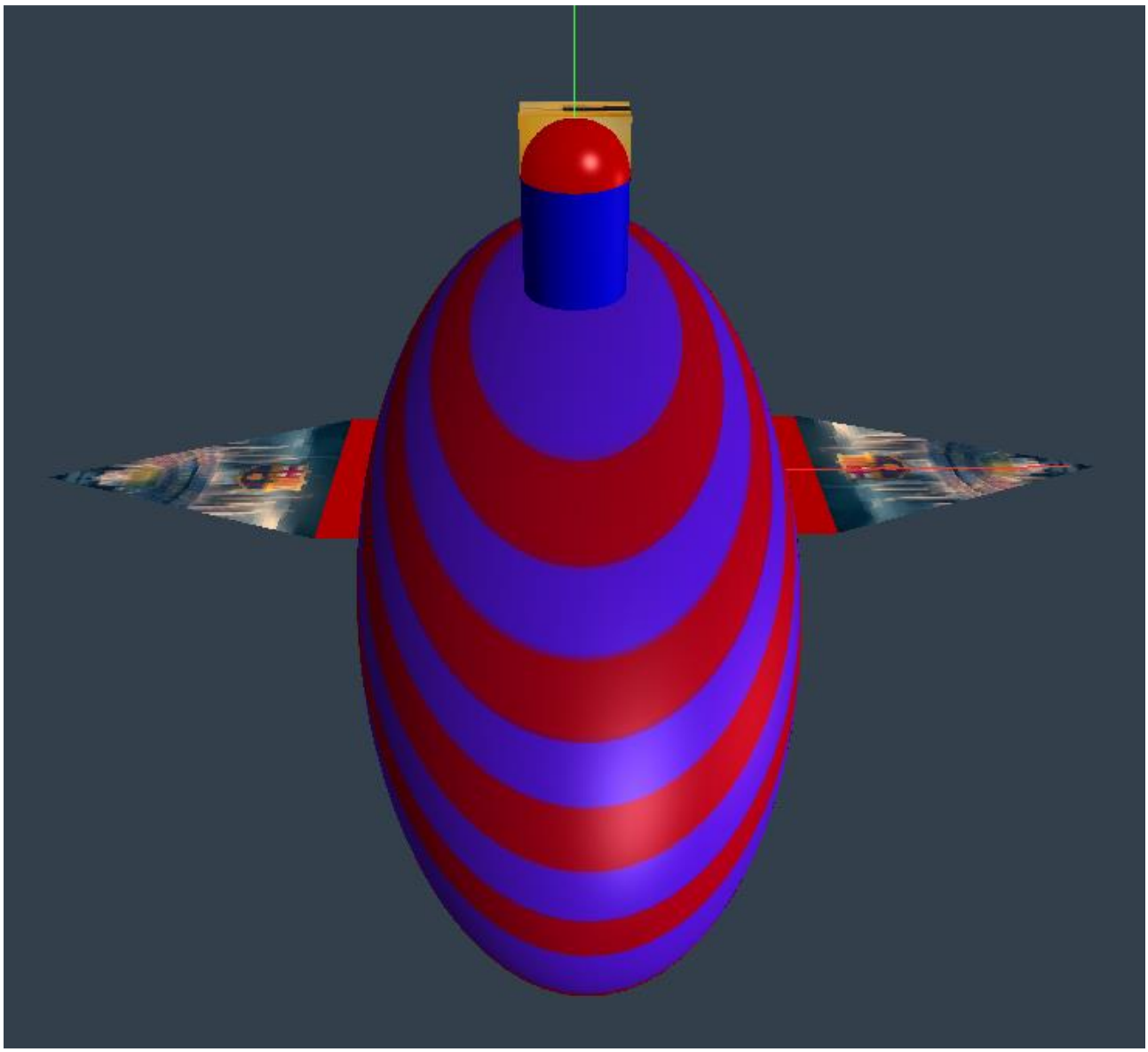
```
float vmin = -M_PI/4;  // -45 grados en radianes
float vmax = M_PI/4;   // 45 grados en radianes
float a_oscilante = (vmax + vmin)/2; // Punto medio = 0
float b_oscilante = (vmax - vmin)/2; // Amplitud = π/4
float n = 0.5f; // Frecuencia de oscilación
*pm_tras_cuerpo = translate(vec3{0.0, 0.0, a_oscilante + b_oscilante*sin(2*M_PI*n*t_sec)});
```

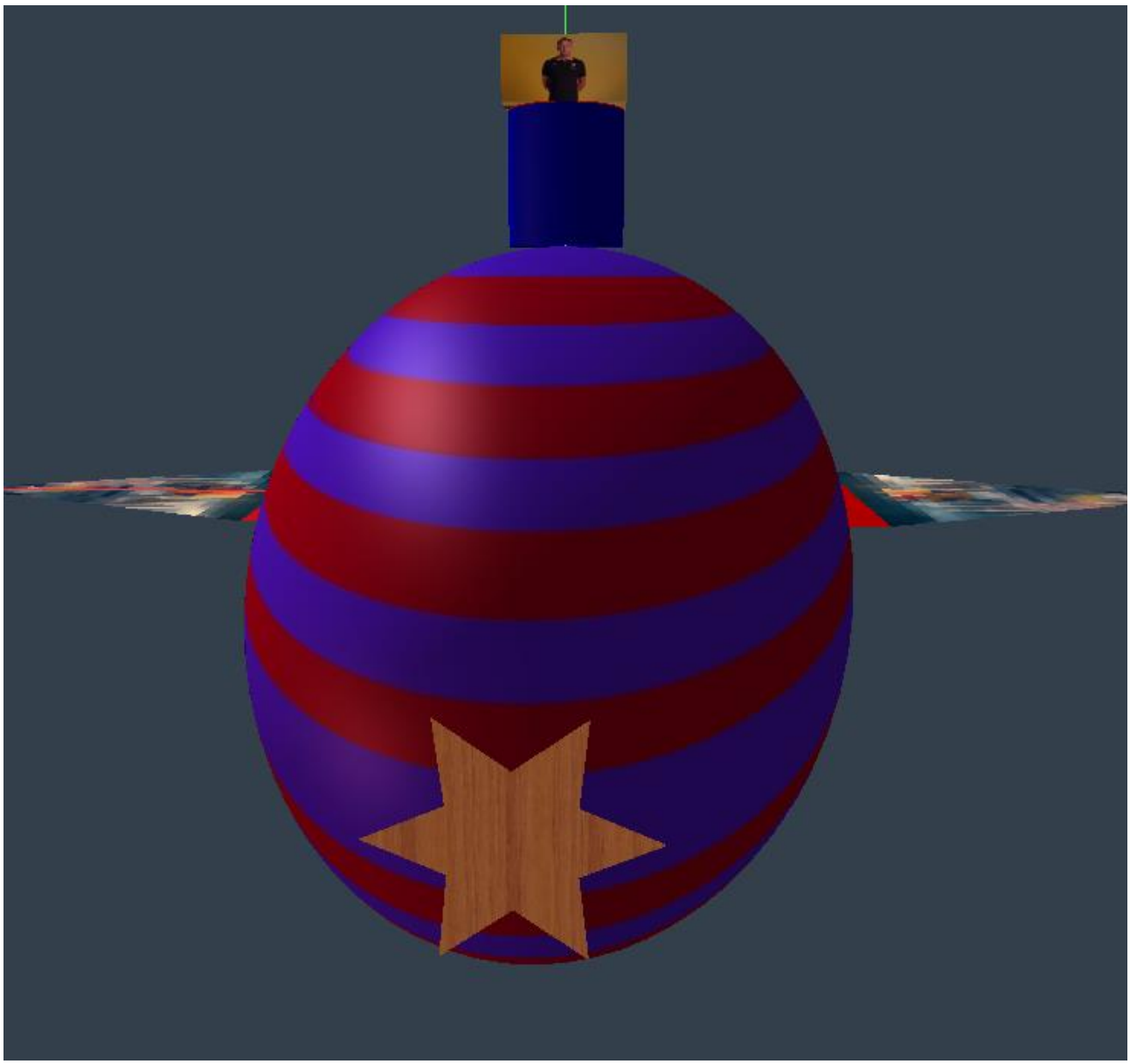
6) *pm_rot_cuerpo: Se encuentra en el nodo submarino.
Es una rotación entorno al eje Y, con una frecuencia de 1/1000 ciclos por segundo.

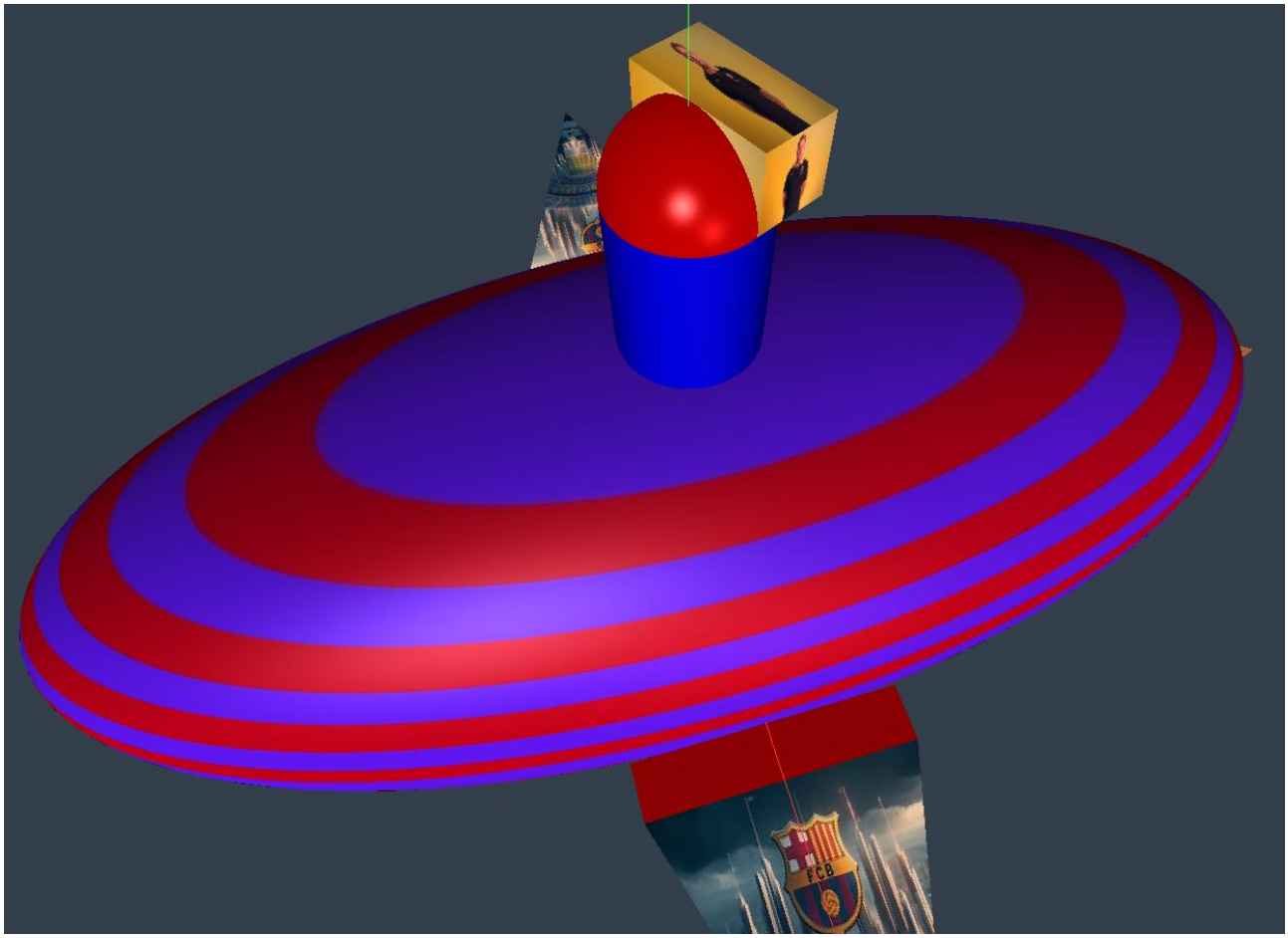
```
float a_lineal = 0.0;
float b_lineal = (2.0*M_PI)/1000; //frecuencia = 1/1000 ciclos por segundo

*pm_rot_cuerpo = rotate(a_lineal+(b_lineal*t_sec), vec3{0.0, 1.0, 0.0});
```

Práctica 4 - Materiales y texturas

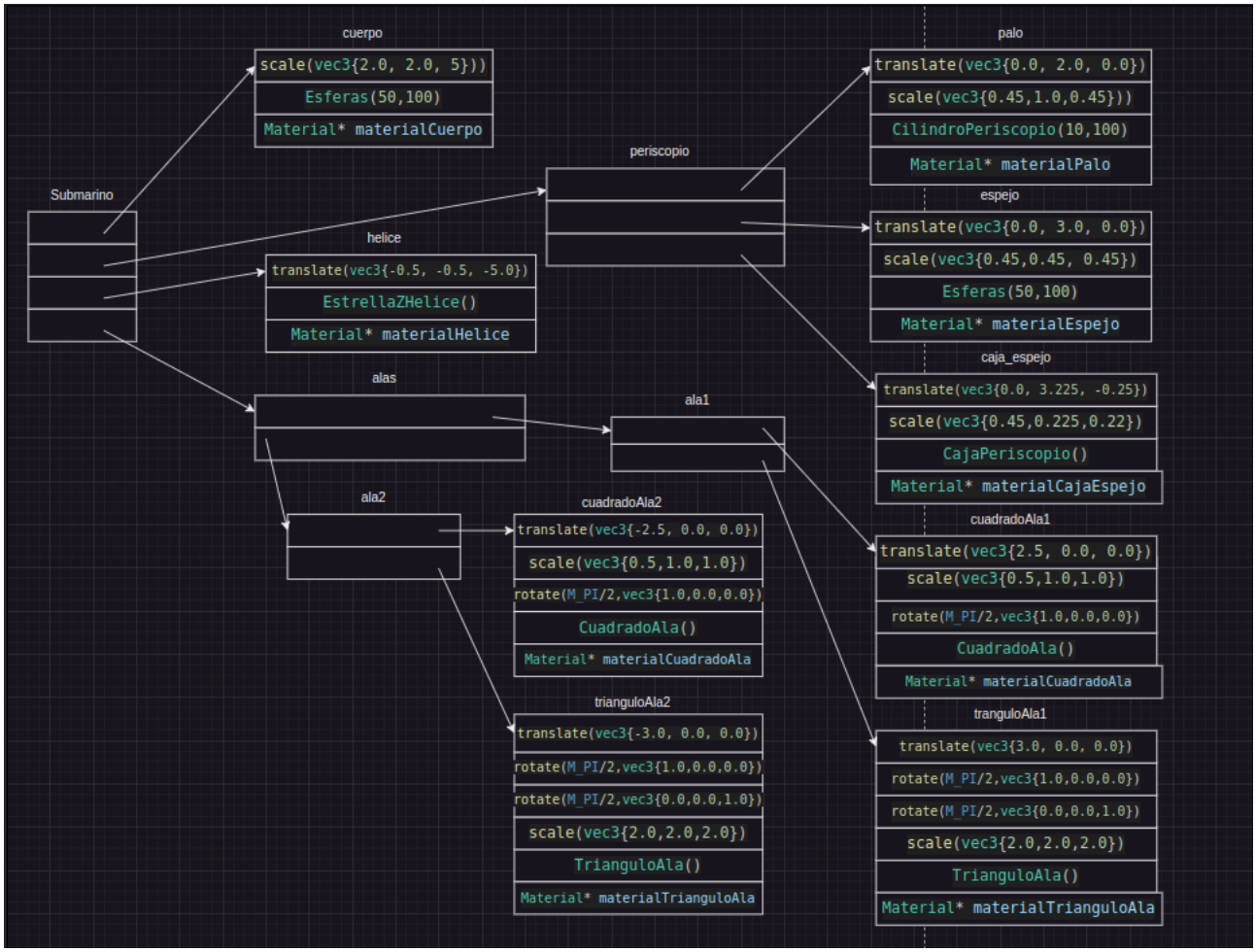
1) Una (o varias) captura de pantalla del modelo, con la iluminación activada






FCB

2)Una nueva versión del grafo de escena, similar al de la práctica 3, pero que incluya las entradas de tipo material(La información de los materiales se encuentran en la sección siguiente)



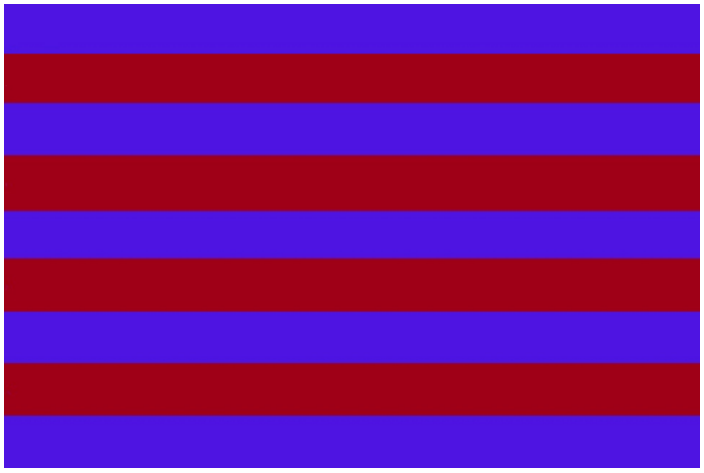
3) Una lista con información de todos y cada uno de los materiales usados en el grafo de escena

- materialCuerpo: Este material lo usamos en el nodo cuerpo. No queremos que tenga mucho brillo para que se camufle mas facilmente y por tanto creamos un materiale eminentemente difuso, para que refleje la luz en todas las direcciones por igual.

```
TexturaXY* texturaCuerpo = new TexturaXY("fcbarcelona_.jpg");
Material* materialCuerpo = new Material(texturaCuerpo, 0.4, 0.9, 0.4, 30.0);
```

Este material usa la textura automatica en los ejes XY, fcbarcelona_.jpg, por lo que los dos 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};
```



- materialHelice: Este material lo usamos con el nodo helice. En este caso por el tipo de textura aunque le subamos la componente pseudo-especular no afecta mucho, el objetivo es que se aprecie bien la textura de madera.

```
TexturaXY* texturaHelice = new TexturaXY("text-madera.jpg");
Material* materialHelice = new Material(texturaHelice, 0.6, 0.7, 0.6, 100.0);
```

Este material usa la textura automática en los ejes XY, text-madera.jpg, por lo que los 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};
```

- materialPalo: Este material lo usamos en el nodo palo. Al igual que el cuerpo es un material eminentemente difuso. Este material no tiene asociada textura.

```
Material* materialPalo = new Material( 0.2, 0.9, 0.1, 15.0);
palo->agregar(materialPalo);
```

- MaterialEspejo: Este material lo usamos en el nodo espejo. En ese caso queremos que tenga mas brillo, luego tenemos un material pseudo-especular con un alto exponente de brillo. Este material no tiene asociada textura.

```
Material* materialEspejo = new Material(0.5, 0.3, 0.8, 100.0);
espejo->agregar(materialEspejo);
```

- MaterialCajaEspejo: Este material lo usamos en el nodo caja_espejo. Como en el caso de la helice, nuestro objetivo en este caso es que se aprecie bien la textura.

```
Textura* texturaCajaEspejo = new Textura("flick.jpeg");
Material* materialCajaEspejo = new Material(texturaCajaEspejo, 0.6, 0.8, 0.7, 50.0);
caja_espejo->agregar(materialCajaEspejo);
```

Este material usa la textura flick.jpeg. No tiene generación automática de texturas.

- MaterialCuadradoAla: Este material lo usamos en los nodos cuadradoAla1 y cuadradoAla2. Tenemos el mismo objetivo que en el caso del cuerpo y del palo. Este material no tiene textura.

```
Material* materialCuadradoAla = new Material(0.3, 0.9, 0.3, 15);
```

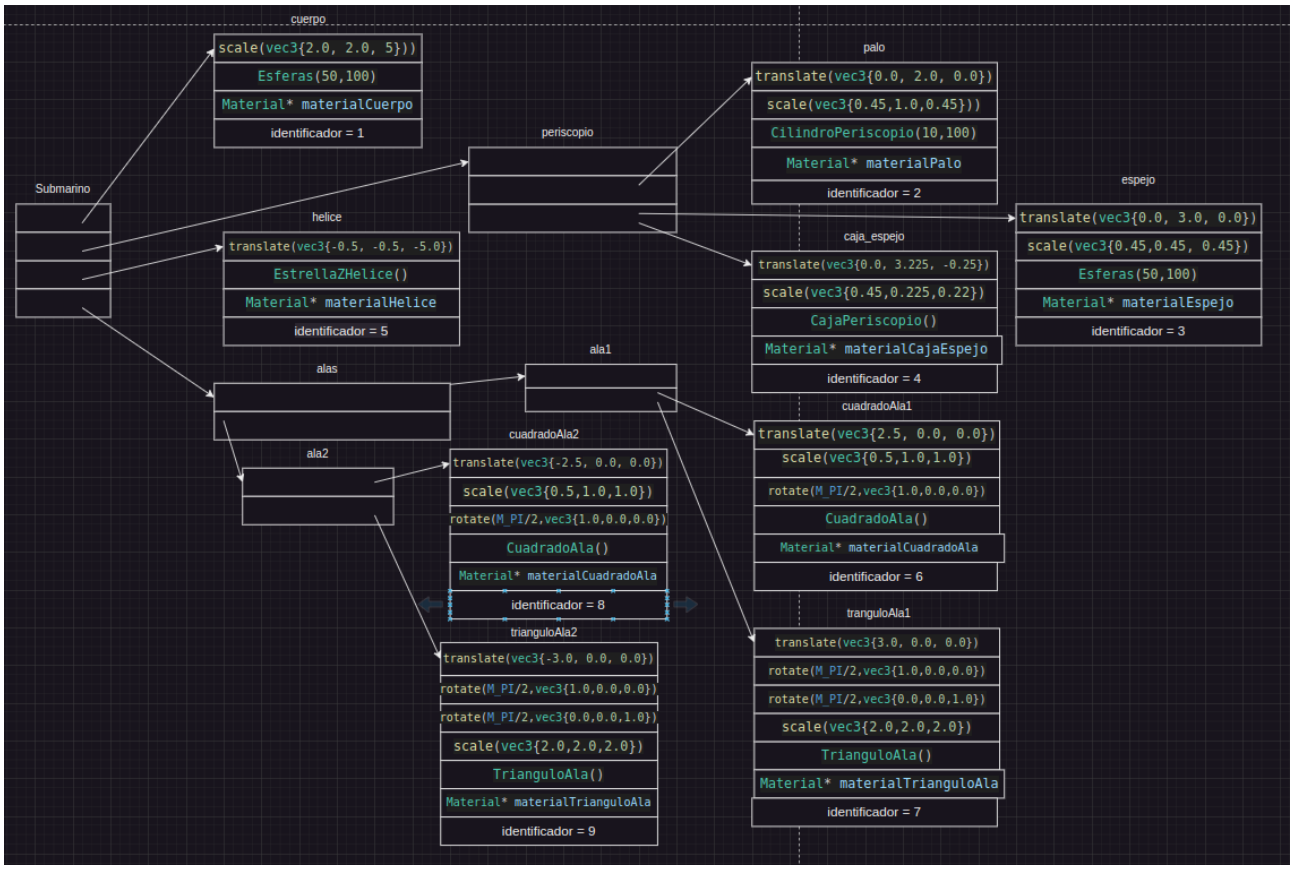
- MaterialTrianguloAla. Este material lo usamos en los nodos trianguloAla1 y trianguloAla2. En este caso el objetivo es que se aprecie bien la textura para ello aumentamos el coeficiente de ambiente y los otros dos coeficientes los dejamos parecidos.

```
Textura* texturaTrianguloAla = new Textura("escudo2.jpeg");
Material* materialTrianguloAla = new Material(texturaTrianguloAla, 0.8, 0.5, 0.7, 100.0);
```

Este material usa la textura escudo2.jpeg. No tiene generación automática de texturas.

Práctica 5 - Identificadores de selección

1) El grafo de escena tipo PHIGS incluyendo las transformaciones (práctica 3), las entradas de tipo material (práctica 4) y además, para cada nodo u objeto, se mostrará el identificador de selección usado para el nodo (un valor entero).



2) Un lista con información de todos y cada uno de los identificadores de selección que se han añadido al grafo

Para crear los identificadores de selección del grafo, hemos creado una variable unsigned indice_param, inicialmente con el valor 1, y cada vez que añado un identificador, aumento en uno el valor de indice_param. Todos se añaden en el archivo modelo-jer.cpp:

→ cuerpo: identificador = 1

```
284     cuerpo->ponerNombre("cuerpo del sumarino");
285     cuerpo->ponerColor({1.0,1.0,0.0}); //Amarillo
286     cuerpo->ponerIdentificador(indice_param);
287     indice_param++;
```

→ palo: identificador = 2

```
302     palo->ponerNombre("palo del periscopio");
303     palo->ponerColor({0.0,0.0,1.0}); //azul
304     palo->ponerIdentificador(indice_param);
305     indice_param++;
```

→ espejo: identificador = 3

```
317     espejo->ponerNombre("espejo del periscopio");
318     espejo->ponerColor({1.0,0.0,0.0}); //rojo
319     espejo->ponerIdentificador(indice_param);
320     indice_param++;
```

→ caja_espejo: identifcador = 4

```
333     caja_espejo->ponerNombre("caja del periscopio");
334     caja_espejo->ponerColor({0.0,0.0,0.0}); //negro
335     caja_espejo->ponerIdentificador(indice_param);
336     indice_param++;
```

→ helice: identificador = 5

```
357     helice->ponerNombre("helice");
358     helice->ponerColor({1.0,0.8,0.0}); //amarillo
359     helice->ponerIdentificador(indice_param);
360     indice_param++;
```

→ cuadradoAla1: identificador = 6

```
383     cuadradoAla1->ponerNombre("cuadrado del primer ala");
384     cuadradoAla1->ponerColor({1.0,0.0,0.0}); //rojo
385     cuadradoAla1->ponerIdentificador(indice_param);
386     indice_param++;
```

→ trianguloAla1: identificador = 7

```
398     trianguloAla1->ponerNombre("triangulo del primer ala");
399     trianguloAla1->ponerIdentificador(indice_param);
400     indice_param++;
401
```

→ cuadradoAla2: identificador = 8

```
420     cuadradoAla2->ponerNombre("cuadrado del segundo ala");
421     cuadradoAla2->ponerColor({1.0,0.0,0.0}); //rojo
422     cuadradoAla2->ponerIdentificador(indice_param);
423     indice_param++;
```

→ trianguloAla2: identificador = 9

```
435     trianguloAla2->ponerNombre("triangulo del segundo ala");
436     trianguloAla2->ponerIdentificador(indice_param);
437     indice_param++;
438
```