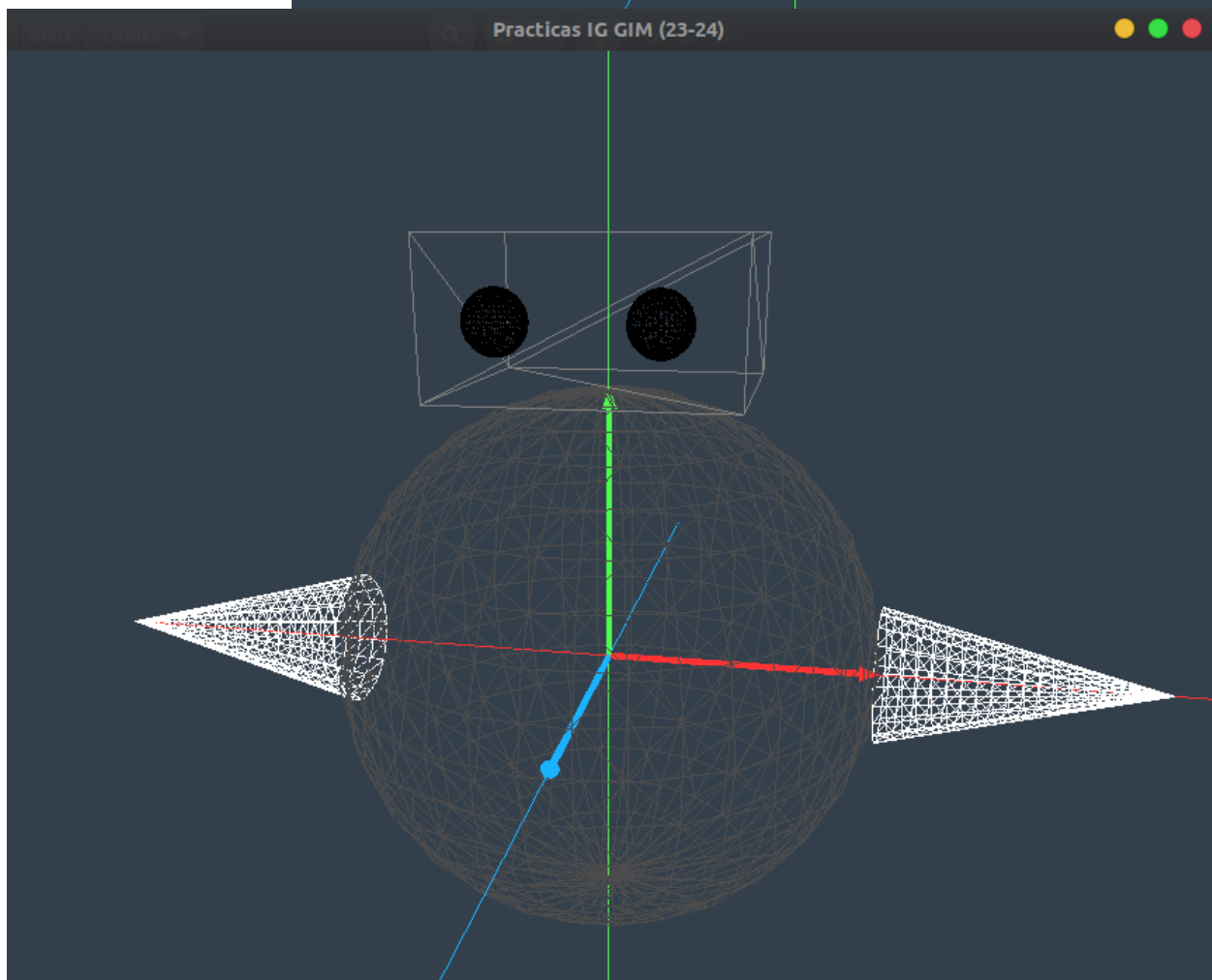
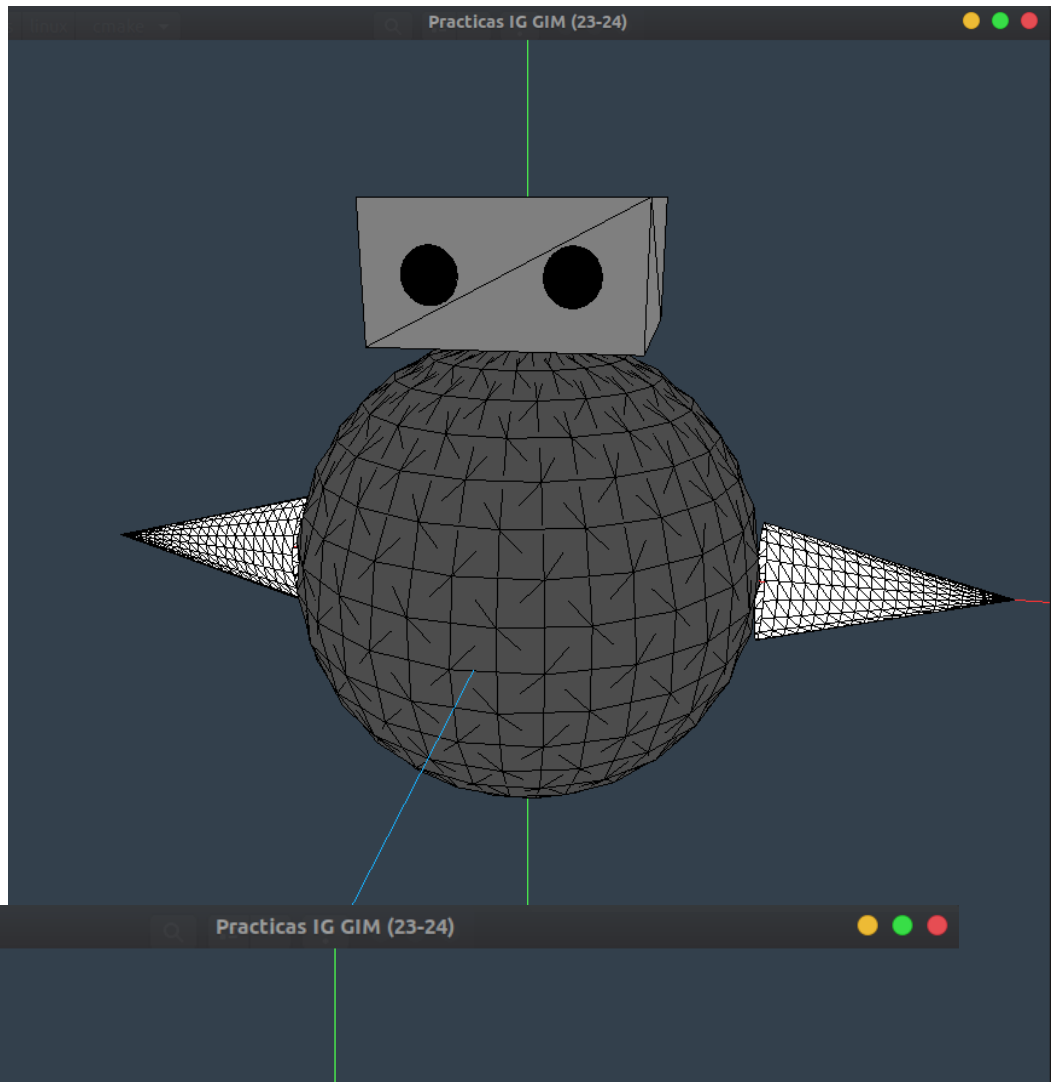
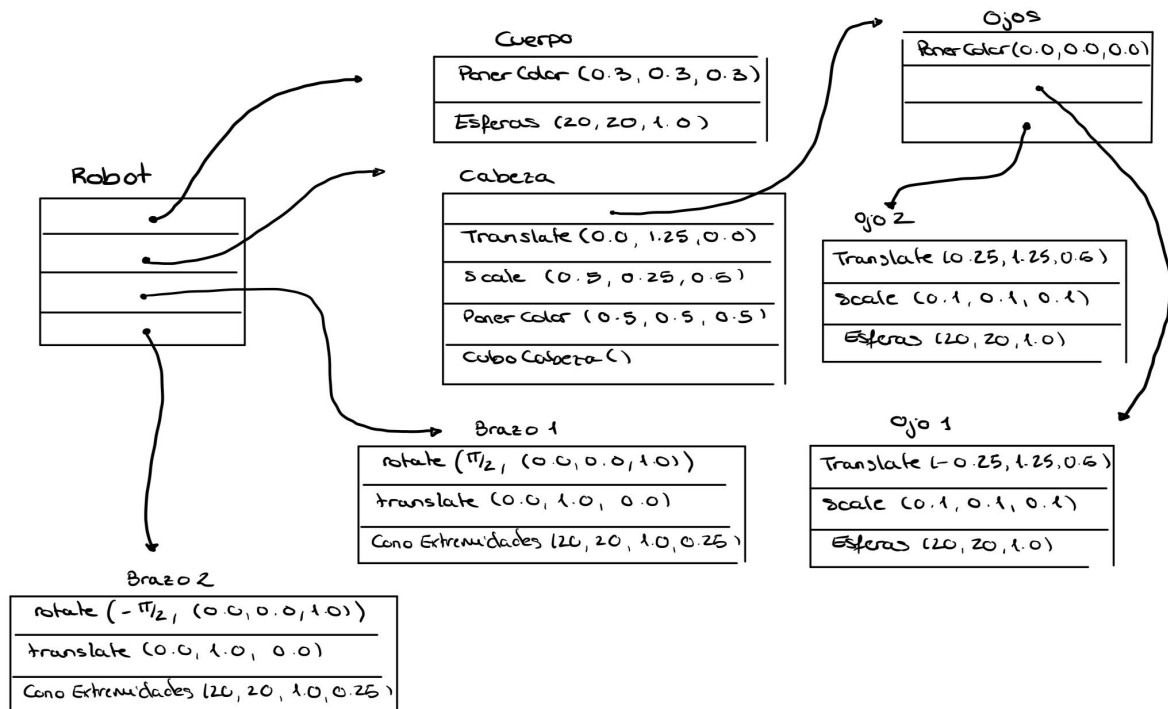


INFORMÁTICA GRÁFICA

Daniel Alconchel Vázquez
Matemáticas e Informática
Curso 23-24



Grafo de escena PHIGS:



Lista de los nodos del grafo:

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

- 1) **Ojos** -> Se tratan de dos esferas trasladadas desde el origen de coordenadas hasta el punto central de donde estará la cabeza y escaladas. Están en color negro $RGB(0.0, 0.0, 0.0)$. Instancias de Esferas.
- 2) **Cabeza** -> Se trata de un cubo desplazado desde el origen hasta la posición que nos interesa y escalado para que aparezca achatado. Tiene color gris $RGB(0.5, 0.5, 0.5)$ y cuenta con el parámetro de libertad **pm_rot_cabeza*. Instancia de CuboCabeza.
- 3) **Extremidad1** -> Se trata de un cono trasladado desde el origen y rotado. Cuenta con un parámetro de libertad **pm_tras_brazo1*. Tiene el color por defecto, que es el blanco. Instancia de ConoExtremidades.
- 4) **Extremidad2** -> Se trata de un cono trasladado desde el origen y rotado. Cuenta con un parámetro de libertad **pm_rot_brazo2*. Tiene el color por defecto, que es el blanco. Instancia de ConoExtremidades.
- 5) **Cuerpo** -> Se trata de otra esfera, instancia de Esferas, de color gris $RGB(0.3, 0.3, 0.3)$ y cuenta con dos parámetros de libertad **pm_rot_cuerpo* y **pm_tras_cuerpo*.
- 6) **Robot** -> Agrupa los nodos anteriores.

Lista de los parámetros de libertad:

- 1) **pm_rot_cabeza* -> Permite rotar la cabeza 360 grados.
`rotate(float(M_PI/10000), vec3{0.0, 1.0, 0.0});`
- 2) **pm_tras_brazo1* -> Permite lanzar el brazo y que vuelva a travel de una traslación senoidal. `translate(vec3(0.5*sin(2*M_PI*t_sec)-0.5, 0.0, 0.0));`
- 3) **pm_rot_brazo2* -> Permite que el brazo rote 360° "simulando el movimiento que se hace al nadar a crol"
`translate(vec3(0.0, 0.5*cos(2*M_PI*t_sec), 0.5*sin(2*M_PI*t_sec)));`

- 4) *pm_rot_cuerpo y *pm_tras_cuerpo -> Permiten avanzar el cuerpo en linea recta y volver, haciendo girar la bola principal como si estuviese rodando por el suelo.

```
rotate(float(0.5*cos(2*M_PI*t_sec)),vec3{1.0,0.0,0.0})translate(v  
ec3(0.0,0.0,0.5*sin(2*M_PI*t_sec)));  
;
```