



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA e  
INTERACCIÓN HUMANO COMPUTADORA



## **EJERCICIOS DE CLASE N° 04**

**NOMBRE COMPLETO:** Hernández Vázquez Daniela

**N° de Cuenta:** 318092867

**GRUPO DE LABORATORIO:** 01

**GRUPO DE TEORÍA:** 02

**SEMESTRE 2024-2**

**FECHA DE ENTREGA LÍMITE:** 5 de marzo de 2024

**CALIFICACIÓN:** \_\_\_\_\_

## EJERCICIOS DE SESIÓN:

Para el ejercicio se solicita terminar de construir la grúa con: 1 Cabina (prisma rectangular), brazo de 3 partes, 4 articulaciones, 1 canasta. Utilizamos como base el archivo *practica4\_conjerarquia*, y fue agregada en primer lugar la articulación unida a la cabina.

```
modelaux = model;

model = glm::translate(model, glm::vec3(0.0f, 3.5f, -4.0f));
model = glm::scale(model, glm::vec3(10.0f, 5.0f, 5.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
color = glm::vec3(0.0f, 1.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[0]->RenderMesh(); //dibuja cubo, pirámide triangular, pirámide base cuadrangular
//para descartar la escala que no quiero heredar se carga la información de la matrix auxiliar

model = modelaux;

// SE EMPIEZA EL DIBUJO DEL BRAZO
//articulación 1 - Esfera
model = glm::translate(model, glm::vec3(0.0f, 6.0f, -4.0f));
model = glm::rotate(model, glm::radians(mainWindow.getarticulacion1()), glm::vec3(0.0f, 0.0f, 1.0f)); //Gira sobre el propio eje

modelaux = model; // cambia de matriz

model = glm::scale(model, glm::vec3(0.5f, 0.5f, 0.5f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
color = glm::vec3(1.0f, 1.0f, 1.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
sp.render();

model = modelaux; // cambia de matriz
```

Figura 1. Cabina y primera articulación

En segundo lugar, para colocar el segundo brazo calculamos respecto a la imagen de referencia del modelo el ángulo al que debemos posicionarlo, como el ángulo es el mismo trabajado de  $135^\circ$  no es necesario moverlo y trabajamos en X.

```
model = modelaux;

//articulación 2
model = glm::translate(model, glm::vec3(2.5f, 0.0f, 0.0f));
model = glm::rotate(model, glm::radians(mainWindow.getarticulacion2()), glm::vec3(0.0f, 0.0f, 1.0f));

modelaux = model;

//dibujar una pequeña esfera
model = glm::scale(model, glm::vec3(0.5f, 0.5f, 0.5f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
color = glm::vec3(1.0f, 0.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
sp.render();

model = modelaux;

//segundo brazo
model = glm::translate(model, glm::vec3(0.0f, -2.5f, 0.0f));

modelaux = model;

model = glm::scale(model, glm::vec3(1.0f, 5.0f, 1.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
color = glm::vec3(0.0f, 1.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular

model = modelaux;
```

Figura 2. Creación de brazos y articulaciones

Como en el archivo ya se encuentran instanciados el brazo 1 y la articulación 2 y brazo 2 y la articulación 3 instanciamos solamente el brazo 3 siguiendo los mismos pasos.

```

//tercer brazo
//model = glm::rotate(model, glm::radians(135.0f), glm::vec3(0.0f, 0.0f, 1.0f)); //
model = glm::translate(model, glm::vec3(2.5f, 0.0f, 0.0f));

modelaux = model;

model = glm::scale(model, glm::vec3(5.0f, 1.0f, 1.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
color = glm::vec3(1.0f, 1.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular

model = modelaux;

```

*Figura 3. Brazo 3*

Finalmente, instanciamos la última articulación y con ella la cabina cambiando el eje de rotación X en lugar de Z y rotándolo 45°

```

//articulación 4 unida a canasta
model = glm::translate(model, glm::vec3(2.5f, 0.0f, 0.0f));
model = glm::rotate(model, glm::radians(45.0f), glm::vec3(0.0f, 0.0f, 1.0f)); //
model = glm::rotate(model, glm::radians(mainWindow.getarticulacion4()), glm::vec3(1.0f, 0.0f, 0.0f));

modelaux = model;

model = glm::scale(model, glm::vec3(0.5f, 0.5f, 0.5f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
color = glm::vec3(0.0f, 1.0f, 1.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
sp.render();

model = modelaux;

//Canasta
//model = glm::rotate(model, glm::radians(45.0f), glm::vec3(0.0f, 0.0f, 1.0f)); //
model = glm::translate(model, glm::vec3(1.0f, 0.0f, 0.0f));

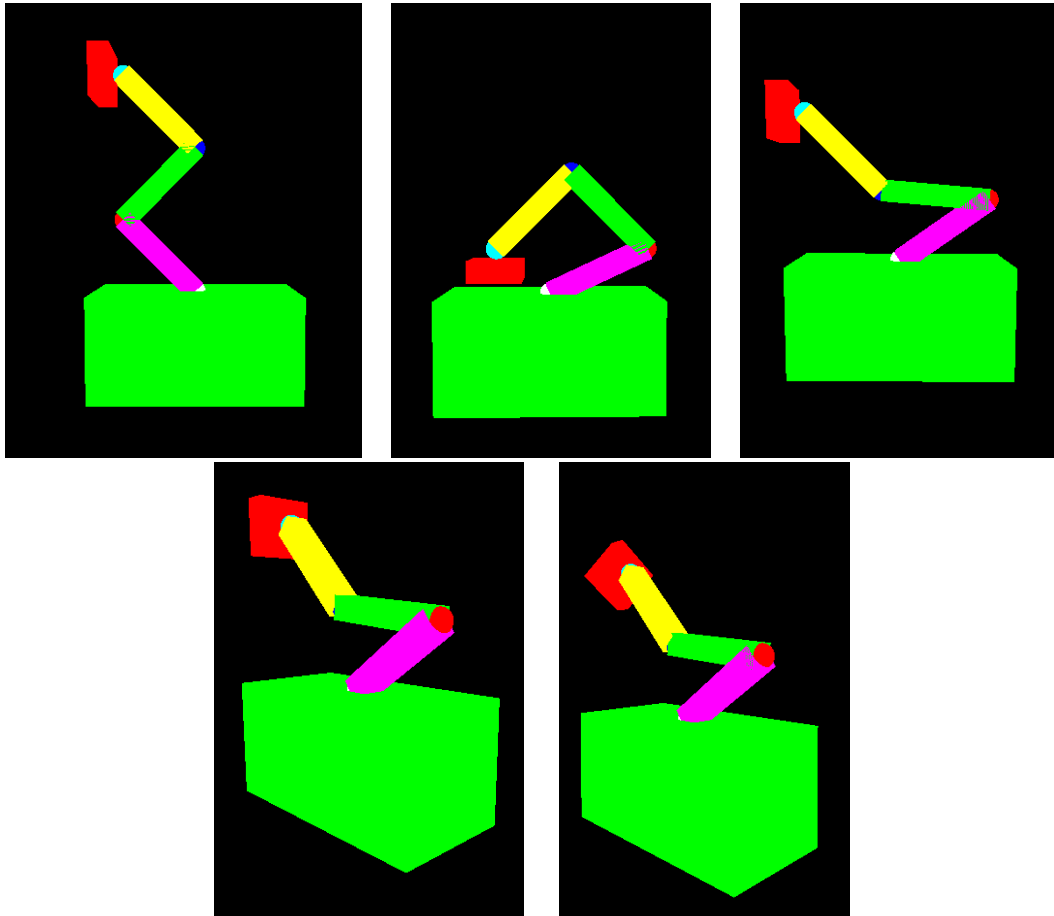
modelaux = model;

model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
model = glm::scale(model, glm::vec3(1.0f, 2.5f, 2.5f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
color = glm::vec3(1.0f, 0.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular

```

*Figura 4. Figura de referencia*

Todo esto dentro de la sección main, en este caso se movilizan los brazos por mediodo las articulaciones con las teclas 'F', 'G', 'H' y 'J'



*Figura 5. Resultado de ejecución*

### **Problemas:**

En este ejercicio solo la mayor complejidad fue hacer que los atributos se heredaran del brazo que estaban utilizando. Los únicos elementos que se mueven dentro de este ejercicio son los brazos, una vez comprendí como funcionaban las matrices de forma lineal fue más fácil comenzar a instanciar las figuras siguientes, debíamos de tomar en cuenta que todos los planos se quedaban con las rotaciones anteriores heredadas así que eso llegó a ser algo difícil de visualizar.

### **Conclusión:**

Creo que los ejercicios están aumentando poco a poco su dificultad y ahora se debe de tener un conocimiento más profundo acerca del funcionamiento de los elementos, lo más tardado de la actividad es realizar los ajustes a la geometría. Me ayudó el tener el video de cómo realizar las configuraciones y más aún tener un archivo que ya tenía los elementos integrados ya que podía comparar que era exactamente lo que se estaba cambiando.