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 Nº 04

NOMBRE COMPLETO: Vargas López Miguel Adán

Nº de Cuenta: 421079522

GRUPO DE LABORATORIO: 3

GRUPO DE TEORÍA: 4

SEMESTRE 2025-1

FECHA DE ENTREGA LÍMITE: 03/09/2024

CALIFICACIÓN: _____

Ejercicio: Terminar de Construir la grúa con:
-Cuerpo de la grúa(prisma rectangular).
-brazo de 3 partes, 4 articulaciones, 1 canasta
Para construir la grúa hay que renderizar primero el cuerpo de la misma, representando la cabina de la misma a la que se le da una figura de prisma rectangular y de un color rojo.

```
//cabina
//model = glm::translate(model, glm::vec3(-1.0f, 2.0f, 0.0f));
//model = glm::rotate(model, glm::radians(135.0f), glm::vec3(0.0f, 0.0f, 1.0f));
modelaux = model;
model = glm::scale(model, glm::vec3(6.0f, 4.0f, 3.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
//la línea de proyección solo se manda una vez a menos que en tiempo de ejecución
//se programe cambio entre proyección ortogonal y perspectiva
glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
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, pirámide triangular, pirámide base cuadrangular
```

También en cada intersección de las primitivas dibujamos una esfera para guíar al cuerpo de la grúa, además de establecer su giro con la articulación que le corresponde, en el caso de la primer esfera que conecta el cuerpo con el primer brazo será la primer articulación que se activara con teclado y así sucesivamente en las siguientes articulaciones

```
//esfera a conectar con articulacion 1 y cabina
//para descartar la escala que no quiero heredar se carga la información de la matrix auxiliar
model = modelaux;
//model = glm::translate(model, glm::vec3(-2.5f, 0.0f, 0.0f));
model = glm::rotate(model, glm::radians(mainWindow.getarticulacion1()), 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(0.0f, 0.0f, 1.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
sp.render();
model = modelaux;
```

Para completar la figura dibujamos el tercer brazo así como su esfera que funciona como articulación y además se compacta con la canasta que se renderiza al final de color rojo.

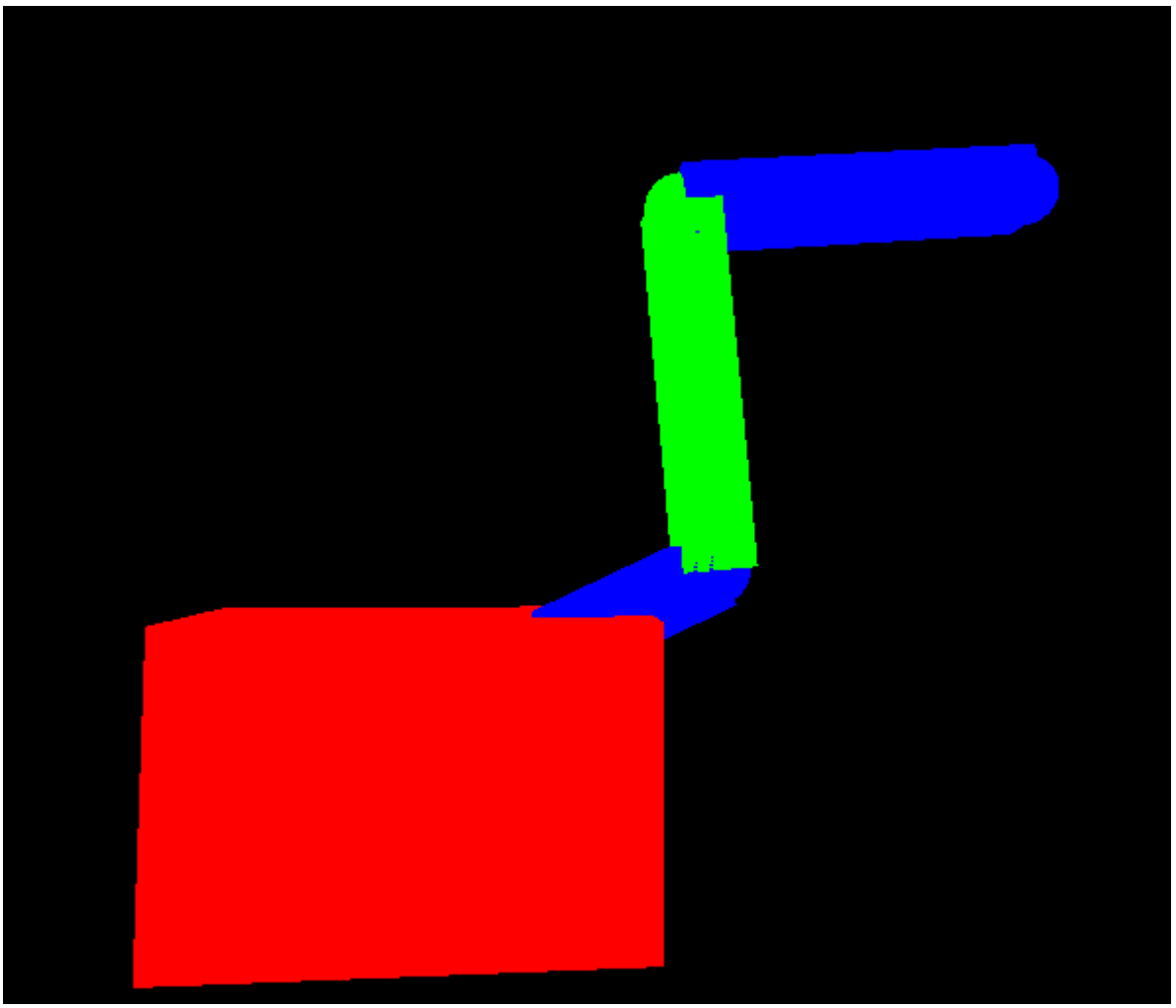
```
//tercer brazo
model = glm::translate(model, glm::vec3(2.5f, 0.0f, 0.0f));
//model = glm::rotate(model, glm::radians(180.0f), glm::vec3(0.0f, 0.0f, 1.0f));
modelaux = model;
model = glm::scale(model, glm::vec3(5.0f, 1.0f, 1.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
color = glm::vec3(0.0f, 0.0f, 1.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular

model = modelaux;
//articulación 3 extremo derecho del tercer brazo
model = glm::translate(model, glm::vec3(2.5f, 0.0f, 0.0f));
model = glm::rotate(model, glm::radians(mainWindow.getarticulacion4()), 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));
sp.render();
model = modelaux;
```

La cabina se instancia como un prisma rectangular en la conexión de la articulación 4 siendo la última parte de la grúa en ser dibujada.

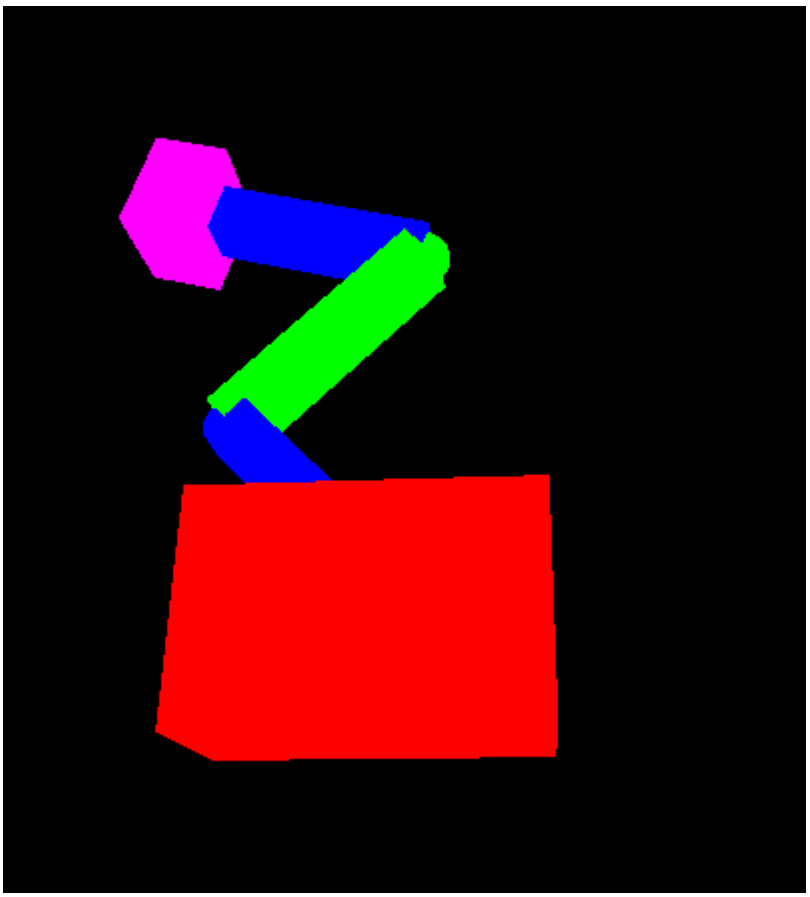
```
//canasta
//modelaux = model;
model = glm::scale(model, glm::vec3(1.5f, 2.0f, 2.0f));
nt main()
  while (!mainWindow.getShouldClose())
                                        1, GL_FALSE, glm::value_ptr(model));
                                    1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
                                    0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[0]->RenderMesh(); //dibuja cubo, pirámide triangular, pirámide base cuadrangular

//esfera a conectar con articulacion 1 y cabina
//para descartar la escala que no quiero heredar se carga la información de la matrix auxiliar
model = modelaux;
```

Complicaciones:

Mayormente fueron por instanciar cada objeto y conectarlos de manera correcta, ya que en ocasiones se instanciaban separados, en grados no correspondía o hacía una mal traslación, para ello me base en el diagrama hecho en clase y en las anotaciones de la imagen de la grúa.



Resultado final:



Conclusiones

Será importante no solo tener conciencia de que el orden a la hora de instanciar objeto es relevante, sino también tener un control y buen manejo de que se hará con estos objetos si es que formarán una figura mayor y por tanto deberán tener conexiones así como una jerarquía, ya que esta jerarquía impacta en la visualización y comportamiento de nuestros objetos, pero ello puede darnos resultados importantes para compactar figuras en una mayor que a su vez puede tener mejores comportamientos o dar una mejor experiencia al usuario como el hecho de mover un brazo de una grúa y este a su vez mueve su canasta.