



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO COMPUTACIÓN GRÁFICA e  
INTERACCIÓN HUMANOCOMPUTADORA



## **Reporte de Práctica 4: Modelado Jerárquico**

**NOMBRE COMPLETO:** Gonzalez Villalba Bryan Jesus

**Nº de Cuenta:** 421530869

**GRUPO DE LABORATORIO:** 11

**GRUPO DE TEORÍA:** 4

**SEMESTRE 2025-1**

**FECHA DE ENTREGA LÍMITE:** 11 de septiembre del 2024

**CALIFICACIÓN:** \_\_\_\_\_

## Introducción:

Para esta práctica realizaremos el reporte, en donde crearemos una figura que ocupe 2 articulaciones por pata, ocupando dos patas y una articulación por oreja, en la cual en mi caso es un robot.

### Desarrollo:

**Imagen 1.0, 1.1 Y 1.2** Primero dibujamos las partes como dibujamos los brazos de la grúa, es un ejemplo del dibujado de los demás brazos.

```

361
362 //para reiniciar la matriz de modelo con valor de la matriz identidad
363 model = glm::mat4(1.0);
364 //rotación alrededor de la articulación que une con la cabina
365 // Articulacion K
366 model = glm::translate(model, glm::vec3(-15.0f, 0.0f, -3.5f));
367 model = glm::rotate(model, glm::radians(mainWindow.getarticulacion5()), glm::vec3(0.0f, 0.0f, 1.0f));
368 // Dibujo de esfera
369 modelaux = model;
370 model = glm::scale(model, glm::vec3(0.5f, 0.5f, 0.5f));
371 color = glm::vec3(1.0f, 1.0f, 1.0f);
372 glUniform3fv(uniformColor, 1, glm::value_ptr(color));
373 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
374 sp.render();
375 modelaux = model;
376
377 //otras transformaciones para el objeto
378 model = glm::translate(model, glm::vec3(-1.0f, 2.0f, 0.0f));
379 model = glm::rotate(model, glm::radians(135.0f), glm::vec3(0.0f, 0.0f, 1.0f));
380 modelaux = model;
381 model = glm::scale(model, glm::vec3(5.0f, 1.0f, 1.0f));
382
383 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
384 //la línea de proyección solo se manda una vez a menos que en tiempo de ejecución
385 //se programe cambio entre proyección ortogonal y perspectiva
386 glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));

```

## Imagen 1.0

```

387 glm::uniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
388 color = glm::vec3(1.0f, 0.0f, 1.0f);
389 glm::uniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
390 meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular
391 //meshList[3]->RenderMeshGeometry(); //dibuja las figuras geométricas cilindro, cono, pirámide base cuadrangular
392 //sp.render(); //dibuja esfera
393
394
395 //para reiniciar la matriz de modelo con valor de la matriz identidad
396 model = modelaux;
397
398 // Articulacion L
399 model = glm::translate(model, glm::vec3(2.5f, 0.0f, 0.0f));
400 model = glm::rotate(model, glm::radians(mainWindow.getarticulacion6()), glm::vec3(0.0f, 0.0f, 1.0f));
401 // Dibujo de esfera
402 modelaux = model;
403 model = glm::scale(model, glm::vec3(0.5f, 0.5f, 0.5f));
404 color = glm::vec3(1.0f, 1.0f, 1.0f);
405 glm::uniform3fv(uniformColor, 1, glm::value_ptr(color));
406 glm::uniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
407 sp.render();
408 model = modelaux;
409
410 // Segundo brazo
411 model = glm::translate(model, glm::vec3(0.0f, -2.5f, 0.0f));
412 //otras transformaciones para el objeto
413 modelaux = model;
414 model = glm::scale(model, glm::vec3(1.0f, 5.0f, 1.0f));
415
416 glm::uniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
417 //La línea de proyección solo se manda una vez a menos que en tiempo de ejecución

```

### Imagen 1.1

```

409 // Segundo brazo
410 model = glm::translate(model, glm::vec3(0.0f, -2.5f, 0.0f));
411 //otras transformaciones para el objeto
412 modelaux = model;
413 model = glm::scale(model, glm::vec3(1.0f, 5.0f, 1.0f));
414
415 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
416 //La línea de proyección solo se manda una vez a menos que en tiempo de ejecución
417 //se programe cambio entre proyección ortogonal y perspectiva
418 glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
419 glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
420 color = glm::vec3(0.0f, 1.0f, 0.0f);
421 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
422 meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular
423
424
425

```

Imagen 1.2

Imagen 1.3 Dibujamos el cuerpo y la cabeza del robot fuera de la jerarquía.

```

687
688 modelaux = modelauxdos;
689 model = modelaux;
690 //Cuerpo
691 model = glm::translate(model, glm::vec3(-13.0f, 1.0f, -4.0f));
692 modelaux = model;
693 model = glm::scale(model, glm::vec3(3.0f, 4.0f, 3.0f));
694 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
695 color = glm::vec3(0.5f, 0.0f, 1.0f);
696 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
697 meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular
698 model = modelaux;
699
700 modelaux = modelauxdos;
701 model = modelaux;
702 //Cabeza
703 model = glm::translate(model, glm::vec3(-13.0f, 4.0f, -4.0f));
704 modelaux = model;
705 model = glm::scale(model, glm::vec3(1.0f, 1.0f, 1.0f));
706 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
707 color = glm::vec3(0.0f, 1.0f, 1.0f);
708 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
709 sp.render();
710 model = modelaux;
711
712
713

```

Imagen 1.4 Matrices que fueron utilizadas para dibujar al robot y a la grúa.

```

327 glm::mat4 model(1.0); //Inicializar matriz de Modelo 4x4
328 glm::mat4 modelaux(1.0); //Inicializar matriz de Modelo 4x4
329 glm::mat4 modelauxpl(1.0); //Inicializar nueva matriz de modelo 4x4 para no heredar, pl de pata 1
330 glm::mat4 modelauxdos(1.0); //Inicializar nueva matriz de modelo 4x4 para no heredar
331
332 glm::mat4 modelgrua(1.0); //Inicializar matriz de Modelo 4x4
333 glm::mat4 modelauxgrua(1.0); //Inicializar matriz de Modelo 4x4
334 glm::mat4 modelauxgrua2(1.0); //Inicializar nueva matriz de modelo 4x4 para no heredar
335
336 glm::vec3 color = glm::vec3(0.0f, 0.0f, 0.0f); //inicializar Color para enviar a variable Uniform;

```

Imagen 1.4

Imagen 1.5, 1.6 y 1.7 Declaré más variables de articulación y ahí vienen sus teclas.

```

162     if (key == GLFW_KEY_Q)
163     {
164         theWindow->articulacion7 += 10.0;
165     }
166     if (key == GLFW_KEY_E)
167     {
168         theWindow->articulacion8 += 10.0;
169     }
170     if (key == GLFW_KEY_U)
171     {
172         theWindow->articulacion9 += 10.0;
173     }
174     if (key == GLFW_KEY_I)
175     {
176         theWindow->articulacion10 += 10.0;
177     }
178     if (key == GLFW_KEY_V)
179     {
180         theWindow->articulacion11 += 10.0;
181     }
182     if (key == GLFW_KEY_B)
183     {
184         theWindow->articulacion12 += 10.0;
185     }
186     if (key == GLFW_KEY_N)
187     {
188         theWindow->articulacion13 += 10.0;
189     }
190     if (key == GLFW_KEY_M)
191     {
192         theWindow->articulacion14 += 10.0;
193     }

```

Imagen 1.5

```

23     GLfloat getarticulacion1() { return articulacion1; }
24     GLfloat getarticulacion2() { return articulacion2; }
25     GLfloat getarticulacion3() { return articulacion3; }
26     GLfloat getarticulacion4() { return articulacion4; }
27     GLfloat getarticulacion5() { return articulacion5; }
28     GLfloat getarticulacion6() { return articulacion6; }
29     GLfloat getarticulacion7() { return articulacion7; }
30     GLfloat getarticulacion8() { return articulacion8; }
31     GLfloat getarticulacion9() { return articulacion9; }
32     GLfloat getarticulacion10() { return articulacion10; }
33     GLfloat getarticulacion11() { return articulacion11; }
34     GLfloat getarticulacion12() { return articulacion12; }
35     GLfloat getarticulacion13() { return articulacion13; }
36     GLfloat getarticulacion14() { return articulacion14; }

```

Imagen 1.6

```

42     GLfloat rotax, rotay, rotaz, articulacion1, articulacion2, articulacion3, articulacion4, articulacion5, articulacion6,
43     articulacion7, articulacion8, articulacion9, articulacion10, articulacion11, articulacion12, articulacion13, articulacion14;
44     bool keys[1024];

```

Imagen 1.7

Imagen 1.8, 1.9 y 2.0 Resultado final.

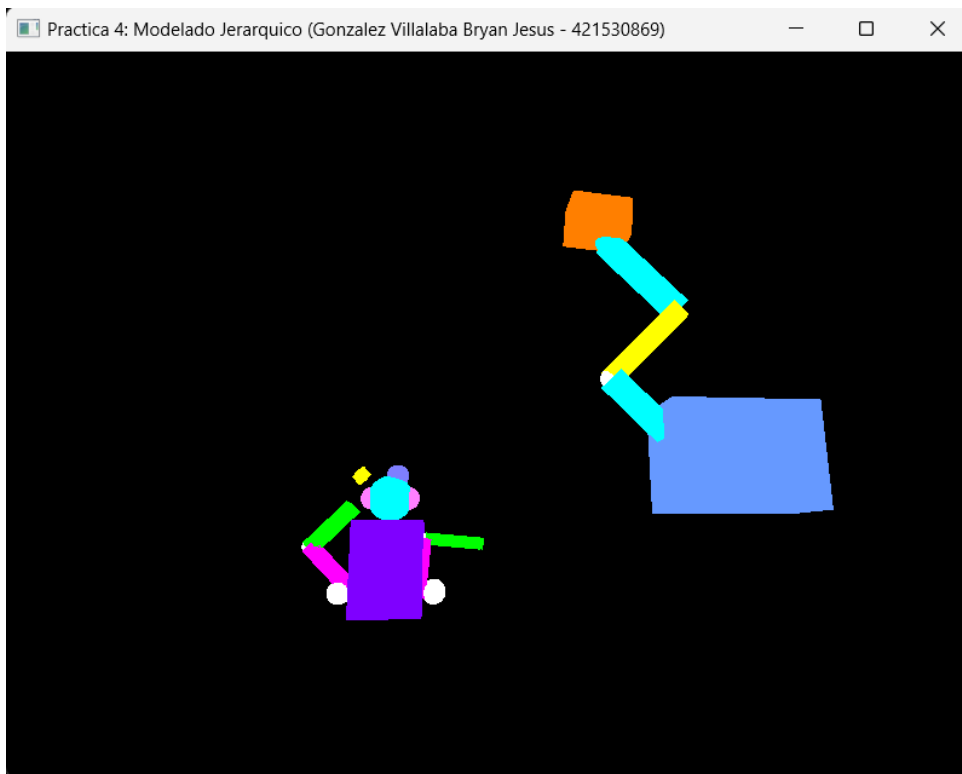


Imagen 1.8

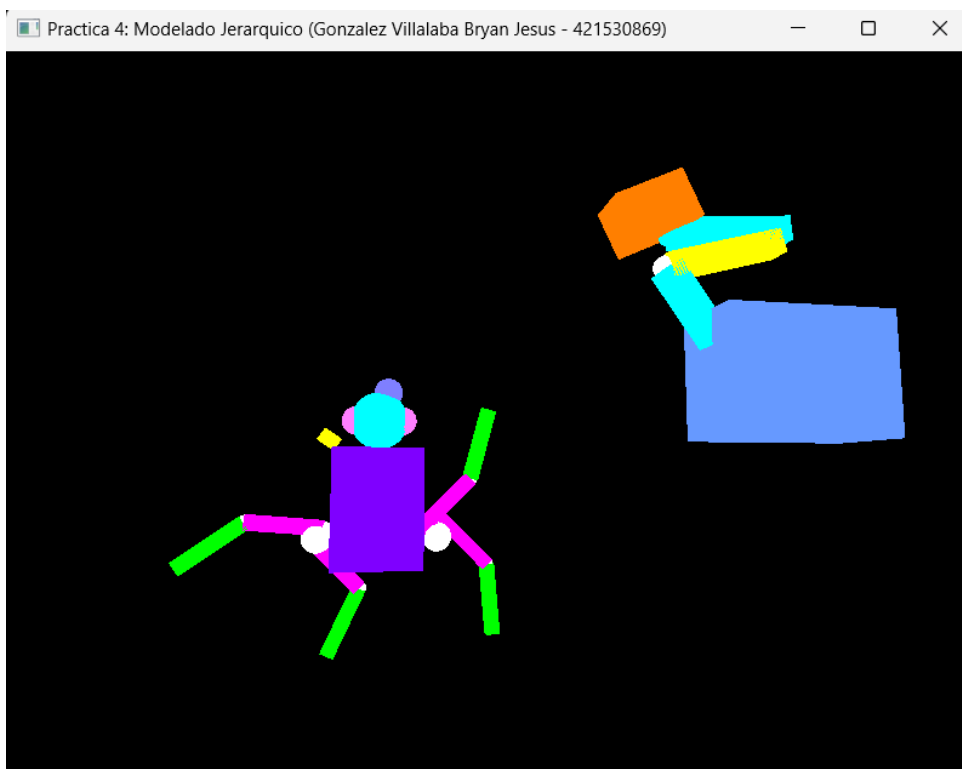


Imagen 1.9

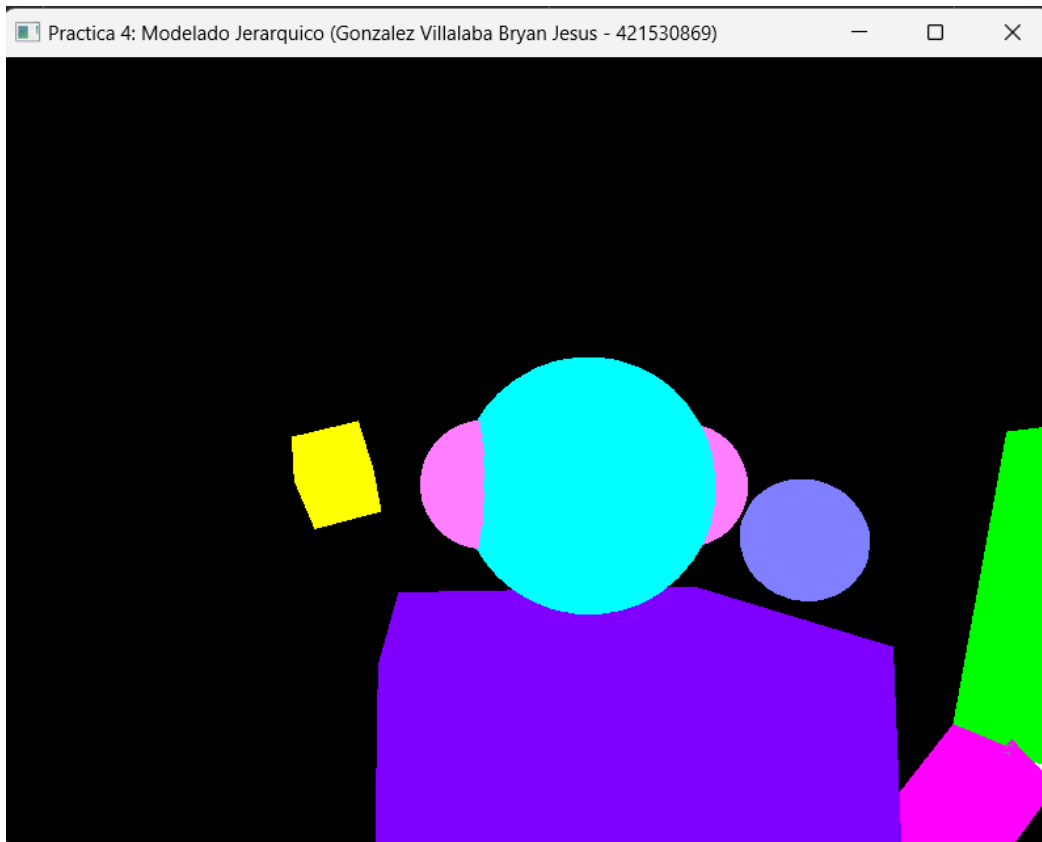


Imagen 2.0

### Conclusión:

#### 421530869:

La práctica fue difícil, porque al principio no entiendo como ocupar bien las matrices, pero después le agarre la onda y pude acomodar mis figuras, y dibujar más o menos bien la grúa, por tiempo y por qué se me junto otro trabajo de teoría, no acomode bien las patas de mi robot, las patas del lado izquierdo están invertidas, además quise dibujar un cilindro pero no pude, me dibujaba una pieza hueca por abajo, así que decidí utilizar un cuadrado, y para las orejas decidí ponerle articulaciones sobresalientes de la cabeza y hacer rotar una esfera y un cuadrado, como si estuviera en órbita alrededor del robot.