



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA e
INTERACCIÓN HUMANO COMPUTADORA



REPORTE DE PRÁCTICA N° 04

NOMBRE COMPLETO: García Soto Jean Carlo

N° de Cuenta: 319226304

GRUPO DE LABORATORIO: 03

GRUPO DE TEORÍA: 04

SEMESTRE 2025-2

FECHA DE ENTREGA LÍMITE: 09/03/2025

CALIFICACIÓN: _____

REPORTE DE PRÁCTICA:

1.- Ejecución de los ejercicios que se dejaron, comentar cada uno y capturas de pantalla de bloques de código generados y de ejecución del programa.

1.- Terminar la Grúa con:

- cuerpo (prisma rectangular)
- base (pirámide cuadrangular)
- 4 llantas (4 cilindros) con teclado se pueden girar las 4 llantas por separado

Se crearon 3 variables auxiliares, que en el ejercicio de la grúa solo se usa el modelaux y modelaux2, y en la del gato robot se usan las 3 variables auxiliares.

```
318 glm::mat4 model(1.0); //Inicializar matriz de Modelo 4x4
319 glm::mat4 modelaux(1.0); //Inicializar matriz de Modelo 4x4 auxiliar para la jerarquía
320 glm::mat4 modelaux2(1.0);
321 glm::mat4 modelaux3(1.0);
```

Para empezar, se corrigió lo de inicializar una nueva matriz con el mat4(1.0), y ahora se respetaron todas las jerarquías haciendo que se comporte como solo una figura y no como dos

```
348 //Cabina (Cubo)
349 model = glm::mat4(1.0);
350 model = glm::translate(model, glm::vec3(2.0f, 4.0f, -4.0f));
351 modelaux = model;
352 modelaux2 = model;
353 model = glm::scale(model, glm::vec3(6.0f, 4.0f, 3.0f));
354 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
355 glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
356 color = glm::vec3(0.5f, 0.5f, 0.5f); //Color gris
357 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
358 meshList[0] -> RenderMesh(); //dibuja cubo y pirámide triangular
359 model = modelaux;
360
361 //AQUÍ SE DIBUJA LA CABINA, LA BASE, LAS 4 LLANTAS
362 //Base de la cabina (Piramide cuadrangular)
363 model = glm::translate(model, glm::vec3(0.0f, -2.0f, 0.0f));
364 modelaux = model;
365 model = glm::scale(model, glm::vec3(8.0f, 1.5f, 4.5f));
366 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
367 glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
368 glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
369 color = glm::vec3(0.5f, 0.0f, 0.0f); //Color Blanco
370 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
371 meshList[4] -> RenderMeshGeometry(); //dibuja piramide cuadrangular
372 model = modelaux;
373
374 //Llanta izquierda delantera (Cilindro)
375 model = glm::translate(model, glm::vec3(-3.8f, -0.7f, 2.65f));
376 model = glm::rotate(model, glm::radians(90.0f), glm::vec3(1.0f, 0.0f, 0.0f));
377 model = glm::scale(model, glm::vec3(1.2f, 0.8f, 1.2f));
378 modelaux = model;
379 model = glm::rotate(model, glm::radians(mainWindow.getarticulacion9()), glm::vec3(0.0f, 1.0f, 0.0f));
380 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
381 glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
382 color = glm::vec3(0.0f, 0.0f, 0.0f); //Color Negro
383 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
384 meshList[2] -> RenderMeshGeometry(); //dibuja piramide cuadrangular
385 model = modelaux;
```

Las llantas están ancladas a la base de la cabina (pirámide cuadrangular), por lo que una vez acomodando solo una llanta, las demás llantas están fáciles de reubicar.

```

387 //Llanta derecha delantera (Cilindro)
388 model = glm::translate(model, glm::vec3(6.6f, 0.0f, 0.0f));
389 modelaux = model;
390 model = glm::rotate(model, glm::radians(mainWindow.getarticulacion10()), glm::vec3(0.0f, 1.0f, 0.0f));
391 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
392 glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
393 color = glm::vec3(0.0f, 0.0f, 0.0f); //Color Negro
394 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
395 meshList[2]->RenderMeshGeometry(); //dibuja piramide cuadrangular
396 model = modelaux;
397
398 //Llanta derecha trasera (Cilindro)
399 model = glm::translate(model, glm::vec3(0.0f, -6.63f, 0.0f));
400 modelaux = model;
401 model = glm::rotate(model, glm::radians(mainWindow.getarticulacion11()), glm::vec3(0.0f, 1.0f, 0.0f));
402 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
403 glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
404 color = glm::vec3(0.0f, 0.0f, 0.0f); //Color Negro
405 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
406 meshList[2]->RenderMeshGeometry(); //dibuja piramide cuadrangular
407 model = modelaux;
408
409 //Llanta izquierda trasera (Cilindro)
410 model = glm::translate(model, glm::vec3(-6.6f, 0.0f, 0.0f));
411 modelaux = model;
412 model = glm::rotate(model, glm::radians(mainWindow.getarticulacion12()), glm::vec3(0.0f, 1.0f, 0.0f));
413 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
414 glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
415 color = glm::vec3(0.0f, 0.0f, 0.0f); //Color Negro
416 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
417 meshList[2]->RenderMeshGeometry(); //dibuja piramide cuadrangular
418 model = glm::rotate(model, glm::radians(-90.0f), glm::vec3(1.0f, 0.0f, 0.0f));
419 model = modelaux;
420
421 //articulación 1 (Esfera)
422 //rotación alrededor de la articulación que une con la cabina
423 model = modelaux2;
424 //model = glm::translate(model, glm::vec3(3.0f, 3.0f, 3.5f));
425 model = glm::rotate(model, glm::radians(mainWindow.getarticulacion1()), glm::vec3(0.0f, 0.0f, 1.0f));
426 modelaux = model; //matriz secundario auxiliar temporal para que no se herede la escala
427 model = glm::scale(model, glm::vec3(0.5f, 0.5f, 0.5f));
428 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
429 glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
430 color = glm::vec3(1.0f, 1.0f, 1.0f); //Color Blanco
431 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
432 sp.render(); //dibuja esfera
433 model = modelaux;
434
435 // SE EMPIEZA EL DIBUJO DEL BRAZO
436 //primer brazo que conecta con la cabina (Cubo)
437 model = glm::rotate(model, glm::radians(135.0f), glm::vec3(0.0f, 0.0f, 1.0f));
438 model = glm::translate(model, glm::vec3(2.5f, 0.0f, 0.0f));
439 modelaux = model;
440 model = glm::scale(model, glm::vec3(5.0f, 1.0f, 1.0f));
441 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
442 glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
443 color = glm::vec3(1.0f, 0.0f, 0.0f); //Color rojo para el primer brazo
444 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
445 meshList[0]->RenderMesh(); //dibuja cubo
446 //meshList[2]->RenderMeshGeometry(); //dibuja las figuras geométricas cilindro y cono
447 model = modelaux;
448
449 //articulación 2 (Esfera)
450 model = glm::translate(model, glm::vec3(2.5f, 0.0f, 0.0f));
451 model = glm::rotate(model, glm::radians(mainWindow.getarticulacion2()), glm::vec3(0.0f, 0.0f, 1.0f));
452 modelaux = model;
453 //dibujar una pequeña esfera
454 model = glm::scale(model, glm::vec3(0.5f, 0.5f, 0.5f));
455 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
456 color = glm::vec3(1.0f, 1.0f, 1.0f); //Color blanco
457 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
458 sp.render(); //dibuja esfera
459 model = modelaux;
460
461 //Segundo brazo (Cubo)
462 model = glm::translate(model, glm::vec3(0.0f, -2.5f, 0.0f));
463 modelaux = model;
464 model = glm::scale(model, glm::vec3(1.0f, 5.0f, 1.0f));
465 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
466 color = glm::vec3(0.0f, 0.5f, 0.7f); //Color verde grisaseo
467 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
468 meshList[0]->RenderMesh(); //dibuja cubo
469 model = modelaux;
470
471 //articulación 3 extremo derecho del segundo brazo (esfera)
472 model = glm::translate(model, glm::vec3(0.0f, -2.5f, 0.0f));
473 model = glm::rotate(model, glm::radians(mainWindow.getarticulacion3()), glm::vec3(0.0f, 0.0f, 1.0f));
474 modelaux = model;
475 //dibujar una pequeña esfera
476 model = glm::scale(model, glm::vec3(0.5f, 0.5f, 0.5f));
477 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
478 color = glm::vec3(1.0f, 1.0f, 1.0f); //Color Blanco
479 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
480 sp.render(); //dibuja esfera
481 model = modelaux;

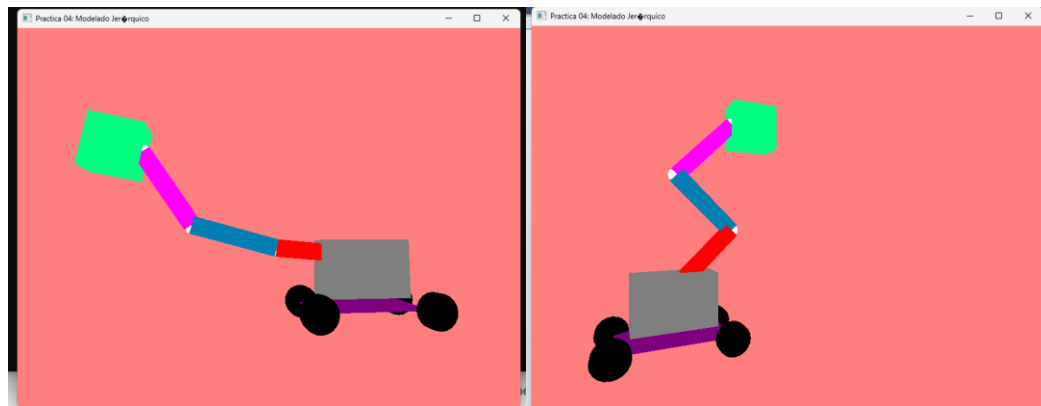
```

```

483 //Brazo 3 (Cubo)
484 model = glm::translate(model, glm::vec3(2.5f, 0.0f, 0.0f));
485 modelaux = model;
486 model = glm::scale(model, glm::vec3(5.0f, 1.0f, 1.0f));
487 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
488 color = glm::vec3(1.0f, 0.0f, 1.0f); //Color magenta
489 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
490 meshList[0]-->RenderMesh(); //dibuja cubo
491 model = modelaux;
492
493 //articulacion 4 (esfera)
494 model = glm::translate(model, glm::vec3(2.5f, 0.0f, 0.0f));
495 model = glm::rotate(model, glm::radians(-135.0f), glm::vec3(0.0f, 0.0f, 1.0f)); //Para girar la cabina en el eje x inicial
496 model = glm::rotate(model, glm::radians(mainWindow.getarticulacion4()), glm::vec3(0.0f, 1.0f, 0.0f));
497 modelaux = model;
498 //dibujar una pequeña esfera
499 model = glm::scale(model, glm::vec3(0.5f, 0.5f, 0.5f));
500 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
501 color = glm::vec3(1.0f, 1.0f, 1.0f); //Color Blanco
502 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
503 sp.render(); //dibuja esfera
504
505 model = modelaux;
506
507 //Canasta (Cubo)
508 model = glm::translate(model, glm::vec3(-1.5f, 0.0f, 0.0f));
509 modelaux = model;
510 model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
511 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
512 color = glm::vec3(0.0f, 1.0f, 0.5f); //Color verde agua
513 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
514 meshList[0]-->RenderMesh(); //dibuja cubo
515

```

Se anexó después el mismo código que se tenía en el ejercicio 4 de la práctica, y esta fue el resultado de la grúa.



2.- Crear un animal robot 3d

- -Instanciando cubos, pirámides, cilindros, conos, esferas:
- -4 patas articuladas en 2 partes (con teclado se puede mover las dos articulaciones de cada pata)
- -cola articulada o 2 orejas articuladas. (con teclado se puede mover la cola o cada oreja independiente)

Se empezó primero con la parte del cuerpo superior del gato, y de ahí se fue haciendo la cabeza, con sus ojos, boca, nariz y orejas, con sus respectivas rotaciones en cada uno. Y siempre tratando de no romper la jerarquía de las figuras.

```
508 // ----- Gato Robot ----- //
509
510 //Cuerpo del gato (Cubo)
511 model = glm::mat4(1.0);
512 model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
513 modelaux = model;
514 modelaux2 = model; //Referencia al cuerpo superior del gato
515 model = glm::scale(model, glm::vec3(6.0f, 4.0f, 3.0f));
516 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
517 glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
518 glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
519 color = glm::vec3(0.5f, 0.5f, 0.5f); //Color gris
520 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
521 meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular
522 model = modelaux;
523
524 //Cuello (Cilindro)
525 model = glm::translate(model, glm::vec3(-3.0f, 0.5f, 0.0f));
526 model = glm::rotate(model, glm::radians(35.0f), glm::vec3(0.0f, 0.0f, 1.0f));
527 modelaux = model;
528 model = glm::scale(model, glm::vec3(0.8f, 4.0f, 0.8f));
529 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
530 color = glm::vec3(0.0f, 0.0f, 0.0f); //Color Negro
531 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
532 meshList[2]->RenderMeshGeometry(); //dibuja cilindro
533 model = modelaux;
534
535 //Cabeza (Esfera)
536 model = glm::translate(model, glm::vec3(0.0f, 4.0f, 0.0f));
537 model = glm::rotate(model, glm::radians(-35.0f), glm::vec3(0.0f, 0.0f, 1.0f));
538 modelaux = model;
539 model = glm::scale(model, glm::vec3(2.5f, 2.5f, 2.5f));
540 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
541 color = glm::vec3(0.5f, 0.5f, 0.5f); //Color gris
542 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
543 sp.render(); //dibuja esfera
544 model = modelaux;
545
546 //Ojo izquierdo
547 model = glm::translate(model, glm::vec3(-1.8f, 0.5f, 1.0f));
548 modelaux = model;
549 model = glm::scale(model, glm::vec3(0.5f, 0.5f, 0.7f));
550 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
551 color = glm::vec3(1.0f, 1.0f, 1.0f); //Color blanco
552 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
553 sp.render(); //dibuja esfera
554 model = modelaux;
```

```

555
556 //Pupila del ojo izquierdo
557 model = glm::translate(model, glm::vec3(-0.5f, 0.0f, 0.2f));
558 modelaux = model;
559 model = glm::scale(model, glm::vec3(0.2f, 0.2f, 0.2f));
560 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
561 color = glm::vec3(0.0f, 0.4f, 0.0f); //Color verde
562 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
563 sp.render(); //dibuja esfera
564 model = modelaux;
565
566 //Ojo derecho
567 model = glm::translate(model, glm::vec3(0.5f, 0.0f, -2.1f));
568 modelaux = model;
569 model = glm::scale(model, glm::vec3(0.5f, 0.5f, 0.7f));
570 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
571 color = glm::vec3(1.0f, 1.0f, 1.0f); //Color blanco
572 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
573 sp.render(); //dibuja esfera
574 model = modelaux;
575
576 //Pupila del ojo derecho
577 model = glm::translate(model, glm::vec3(-0.5f, 0.0f, -0.25f));
578 modelaux = model;
579 model = glm::scale(model, glm::vec3(0.2f, 0.2f, 0.2f));
580 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
581 color = glm::vec3(0.0f, 0.4f, 0.0f); //Color verde
582 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
583 sp.render(); //dibuja esfera
584 model = modelaux;
585
586 //Boca
587 model = glm::translate(model, glm::vec3(1.4f, -1.0f, 1.1f));
588 modelaux = model;
589 model = glm::rotate(model, glm::radians(45.0f), glm::vec3(0.0f, 0.0f, 1.0f));
590 model = glm::scale(model, glm::vec3(1.7f, 0.01f, 1.7f));
591 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
592 color = glm::vec3(0.0f, 0.0f, 0.0f); //Color Negro
593 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
594 model = glm::rotate(model, glm::radians(-45.0f), glm::vec3(0.0f, 0.0f, 1.0f));
595 meshList[2]->RenderMeshGeometry(); //dibuja cilindro
596 model = modelaux;
597
598
599 //Nariz
600 model = glm::translate(model, glm::vec3(-1.1f, 0.1f, 0.0f));
601 modelaux = model;
602 model = glm::rotate(model, glm::radians(35.0f), glm::vec3(0.0f, 0.0f, 1.0f));
603 model = glm::scale(model, glm::vec3(0.0f, 0.3f, 0.0f));
604 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
605 color = glm::vec3(0.0f, 0.0f, 0.0f); //Color negro
606 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
607 meshList[2]->RenderMeshGeometry(); //dibuja cono
608 model = glm::rotate(model, glm::radians(-35.0f), glm::vec3(0.0f, 0.0f, 1.0f));
609 model = modelaux;
610
611 //Oreja izquierda (Cono)
612 model = glm::translate(model, glm::vec3(2.0f, 2.9f, 2.0f));
613 model = glm::rotate(model, glm::radians(25.0f), glm::vec3(1.0f, 0.0f, 0.0f));
614 modelaux = model;
615 model = glm::rotate(model, glm::radians(mainWindow.getarticulacion13()), glm::vec3(0.0f, 1.0f, 0.0f));
616 model = glm::scale(model, glm::vec3(0.1f, 2.0f, 0.3f));
617 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
618 color = glm::vec3(0.3f, 0.3f, 0.3f); //Color gris oscuro
619 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
620 meshList[3]->RenderMeshGeometry(); //dibuja Cono
621 model = modelaux;
622
623 //Oreja derecha (Cono)
624 model = glm::rotate(model, glm::radians(-25.0f), glm::vec3(1.0f, 0.0f, 0.0f));
625 model = glm::translate(model, glm::vec3(0.0f, 0.0f, -3.9f));
626 modelaux = model;
627 model = glm::rotate(model, glm::radians(-25.0f), glm::vec3(1.0f, 0.0f, 0.0f));
628 model = glm::rotate(model, glm::radians(mainWindow.getarticulacion14()), glm::vec3(0.0f, 1.0f, 0.0f));
629 model = glm::scale(model, glm::vec3(0.1f, 2.0f, 0.3f));
630 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
631 color = glm::vec3(0.3f, 0.3f, 0.3f); //Color gris oscuro
632 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
633 meshList[3]->RenderMeshGeometry(); //dibuja Cono
634 model = modelaux;
635
636 //Cuerpo inferior (Cubo)
637 model = glm::rotate(model, glm::radians(25.0f), glm::vec3(1.0f, 0.0f, 0.0f));
638 model = modelaux2;
639 model = glm::translate(model, glm::vec3(5.0f, 0.0f, 0.0f));
640 modelaux = model;
641 modelaux3 = model;
642 //Referencia al cuerpo inferior del gato
643 model = glm::scale(model, glm::vec3(4.0f, 3.0f, 2.0f));
644 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
645 color = glm::vec3(0.0f, 0.0f, 0.0f); //Color negro
646 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
647 meshList[0]->RenderMesh(); //dibuja cubo
648 model = modelaux;

```


Después de la creación de la cabeza con los ojos, orejas, nariz y boca; se usa el `modelaux2` para poder referenciarnos a la parte del cuerpo superior del gato y proceder a crear la parte inferior.

```

634 //Cuerpo inferior (Cubo)
635 model = glm::rotate(model, glm::radians(25.0f), glm::vec3(1.0f, 0.0f, 0.0f));
636 model = modelaux2;
637 model = glm::translate(model, glm::vec3(5.0f, 0.0f, 0.0f));
638 modelaux = model;
639 modelaux3 = model;
640 //Referencia al cuerpo inferior del gato
641 model = glm::scale(model, glm::vec3(4.0f, 3.0f, 2.0f));
642 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
643 color = glm::vec3(0.0f, 0.0f, 0.0f); //Color negro
644 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
645 meshList[0] -> RenderMesh(); //dibuja cubo
646 model = modelaux;

```

Para las patas traseras se utilizó la referencia del cuerpo inferior del gato para así poder pegarles a ella. Se crearon 2 articulaciones por pata, agregando las teclas para poder rotar en los ejes respectivos. Y para las patas del frente se usa el `modelaux3` para poder volver a referenciarnos al cuerpo superior del gato y ubicarlos mejor que tenerlos anclado a alguna pata realizada anterior.

```

648 //----- Pata izquierda trasera -----
649 //Articulación superior (Cilindro)
650 model = glm::translate(model, glm::vec3(0.0f, 0.0f, 2.0f));
651 model = glm::rotate(model, glm::radians(90.0f), glm::vec3(1.0f, 0.0f, 0.0f));
652 model = glm::rotate(model, glm::radians(mainWindow.getarticulacion1()), glm::vec3(0.0f, 1.0f, 0.0f));
653 modelaux = model;
654 model = glm::scale(model, glm::vec3(0.5f, 0.5f, 0.5f));
655 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
656 color = glm::vec3(0.3f, 0.3f, 0.3f); //Color gris oscuro
657 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
658 meshList[2] -> RenderMeshGeometry(); //dibuja cilindro
659 model = modelaux;
660
661 //Parte de la pata superior (Cubo)
662 model = glm::translate(model, glm::vec3(0.0f, -0.64f, 1.0f));
663 modelaux = model;
664 model = glm::scale(model, glm::vec3(1.8f, 0.75f, 3.3f));
665 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
666 color = glm::vec3(0.5f, 0.5f, 0.5f); //Color gris
667 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
668 meshList[0] -> RenderMesh(); //dibuja cubo
669 model = modelaux;
670
671 //Articulación inferior (Cilindro)
672 model = glm::translate(model, glm::vec3(0.0f, 0.0f, 2.3f));
673 model = glm::rotate(model, glm::radians(mainWindow.getarticulacion2()), glm::vec3(0.0f, 1.0f, 0.0f));
674 modelaux = model;
675 model = glm::scale(model, glm::vec3(0.7f, 1.0f, 0.7f));
676 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
677 color = glm::vec3(0.3f, 0.3f, 0.3f); //Color gris oscuro
678 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
679 meshList[2] -> RenderMeshGeometry(); //dibuja cilindro
680 model = modelaux;
681
682 //Parte de la pata inferior (Cubo)
683 model = glm::translate(model, glm::vec3(0.0f, 0.0f, 2.2f));
684 modelaux = model;
685 model = glm::scale(model, glm::vec3(1.8f, 0.75f, 3.0f));
686 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
687 color = glm::vec3(0.5f, 0.5f, 0.5f); //Color gris
688 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
689 meshList[0] -> RenderMesh(); //dibuja cubo
690 model = modelaux;

```

```

691 //Planta del pie inferior (Esfera)
692 model = glm::translate(model, glm::vec3(-0.7f, 0.0f, 1.5f));
693 modelaux = model;
694 model = glm::scale(model, glm::vec3(1.3f, 1.3f, 1.0f));
695 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
696 color = glm::vec3(0.3f, 0.3f, 0.3f); //Color gris oscuro
697 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
698 sp.render(); //dibuja circulo
699 model = glm::rotate(model, glm::radians(-90.0f), glm::vec3(1.0f, 0.0f, 0.0f)); //Dejamos los ejes en sus orientaciones originales
700 model = modelaux;
701 //----- Fin de la pata izquierda trasera -----
702
703

```

Ya, por último, se crea la cola, el cual va a tener cuatro articulaciones, y cuatro trozos de la cola, por cada articulación se le agregó una tecla para así poder rotarlas.

```

875 //----- Inicio de la cola -----
876 //Primera articulación (Esfera)
877 model = modelaux3;
878 model = glm::translate(model, glm::vec3(2.1f, 0.8f, 0.0f));
879 model = glm::rotate(model, glm::radians(mainWindow.getarticulacion9()), glm::vec3(0.0f, 0.0f, 1.0f));
880 modelaux = model;
881 model = glm::scale(model, glm::vec3(0.4f, 0.4f, 0.4f));
882 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
883 color = glm::vec3(0.3f, 0.3f, 0.3f); //Color gris oscuro
884 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
885 sp.render(); //dibuja circulo
886 model = modelaux;
887
888 //Primera trozo de la cola
889 model = glm::translate(model, glm::vec3(1.3f, 0.5f, 0.0f));
890 model = glm::rotate(model, glm::radians(110.0f), glm::vec3(0.0f, 0.0f, 1.0f));
891 modelaux = model;
892 model = glm::scale(model, glm::vec3(0.4f, 2.5f, 0.4f));
893 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
894 color = glm::vec3(0.0f, 0.0f, 0.0f); //Color negro
895 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
896 meshList[2]->RenderMeshGeometry(); //dibuja cilindro
897 model = modelaux;
898
899 //Segunda articulación (Esfera)
900 model = glm::translate(model, glm::vec3(0.0f, -1.35f, 0.0f));
901 model = glm::rotate(model, glm::radians(mainWindow.getarticulacion10()), glm::vec3(0.0f, 0.0f, 1.0f));
902 modelaux = model;
903 model = glm::scale(model, glm::vec3(0.3f, 0.3f, 0.3f));
904 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
905 color = glm::vec3(0.3f, 0.3f, 0.3f); //Color gris oscuro
906 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
907 sp.render(); //dibuja circulo
908 model = modelaux;
909
910 //Segundo trozo de la cola
911 model = glm::translate(model, glm::vec3(0.73f, -0.9f, 0.0f));
912 model = glm::rotate(model, glm::radians(40.0f), glm::vec3(0.0f, 0.0f, 1.0f));
913 modelaux = model;
914 model = glm::scale(model, glm::vec3(0.3f, 2.0f, 0.3f));
915 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
916 color = glm::vec3(0.0f, 0.0f, 0.0f); //Color negro
917 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
918 meshList[2]->RenderMeshGeometry(); //dibuja cilindro
919 model = modelaux;
920
921

```

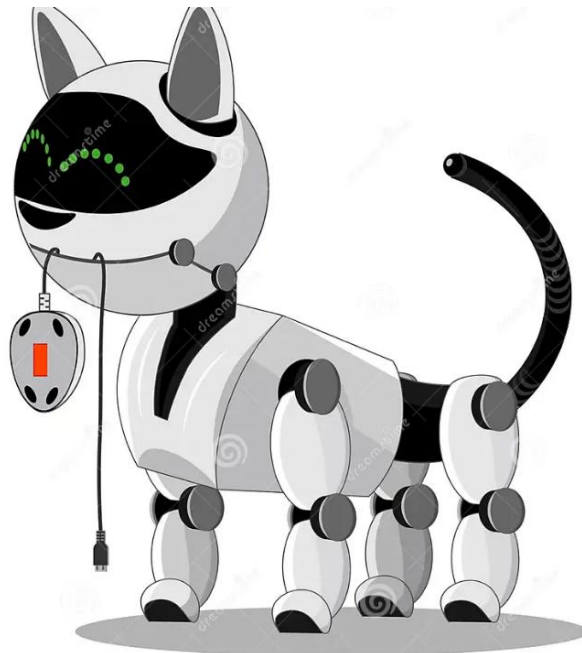


```

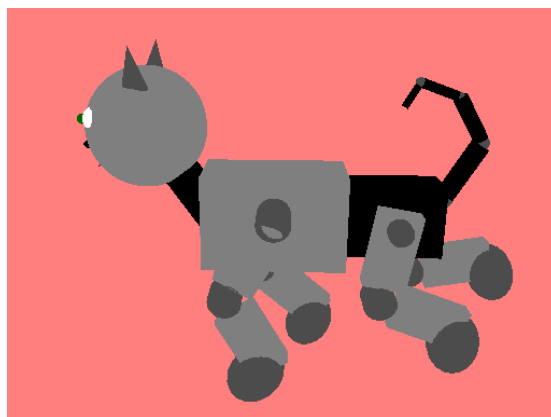
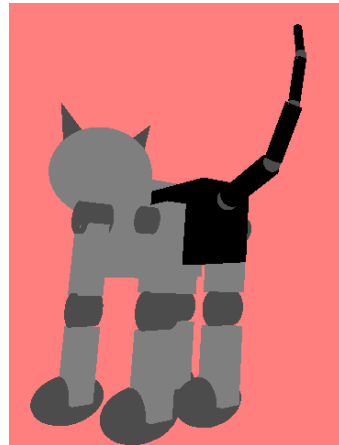
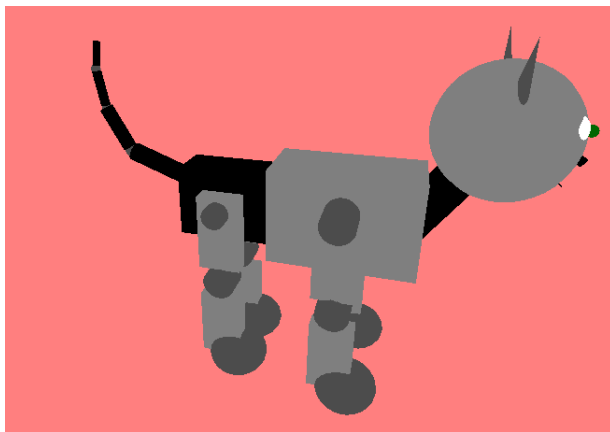
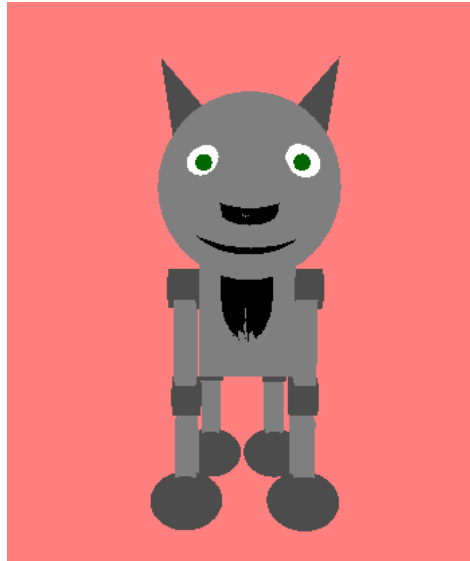
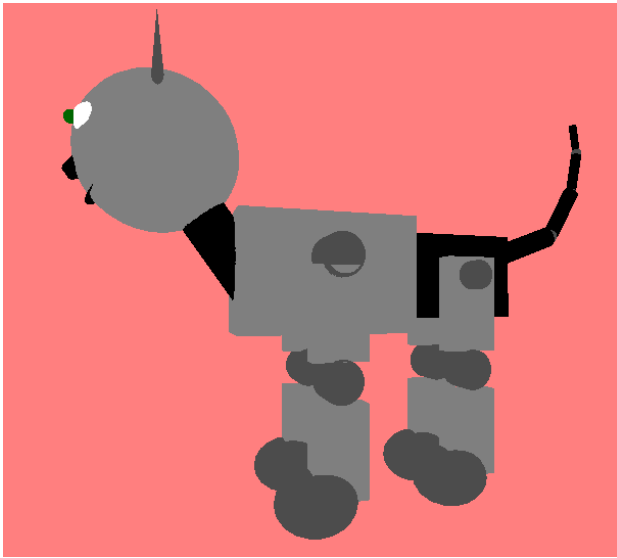
921 //Tercera articulación (Esfera)
922 model = glm::translate(model, glm::vec3(0.0f, -1.0f, 0.0f));
923 model = glm::rotate(model, glm::radians(mainWindow.getarticulacion11()), glm::vec3(0.0f, 0.0f, 1.0f));
924 modelaux = model;
925 model = glm::scale(model, glm::vec3(0.2f, 0.2f, 0.2f));
926 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
927 color = glm::vec3(0.3f, 0.3f, 0.3f); //Color gris oscuro
928 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
929 sp.render(); //dibuja círculo
930 model = modelaux;
931
932 //Tercer trozo de la cola
933 model = glm::translate(model, glm::vec3(0.30f, -0.8f, 0.0f));
934 model = glm::rotate(model, glm::radians(20.0f), glm::vec3(0.0f, 0.0f, 1.0f));
935 modelaux = model;
936 model = glm::scale(model, glm::vec3(0.2f, 1.5f, 0.2f));
937 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
938 color = glm::vec3(0.0f, 0.0f, 0.0f); //Color negro
939 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
940 meshList[2]->RenderMeshGeometry(); //dibuja cilindro
941 model = modelaux;
942
943 //Cuarta articulación (Esfera)
944 model = glm::translate(model, glm::vec3(0.0f, -0.85f, 0.0f));
945 model = glm::rotate(model, glm::radians(mainWindow.getarticulacion12()), glm::vec3(0.0f, 0.0f, 1.0f));
946 modelaux = model;
947 model = glm::scale(model, glm::vec3(0.2f, 0.2f, 0.2f));
948 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
949 color = glm::vec3(0.3f, 0.3f, 0.3f); //Color gris oscuro
950 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
951 sp.render(); //dibuja círculo
952 model = modelaux;
953
954 //Cuarto trozo de la cola
955 model = glm::translate(model, glm::vec3(0.23f, -0.6f, 0.0f));
956 model = glm::rotate(model, glm::radians(20.0f), glm::vec3(0.0f, 0.0f, 1.0f));
957 modelaux = model;
958 model = glm::scale(model, glm::vec3(0.15f, 1.0f, 0.15f));
959 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
960 color = glm::vec3(0.0f, 0.0f, 0.0f); //Color negro
961 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
962 meshList[2]->RenderMeshGeometry(); //dibuja cilindro
963 model = glm::rotate(model, glm::radians(-170.0f), glm::vec3(0.0f, 0.0f, 1.0f)); //Rotamos los ejes a sus rotaciones originales
964 model = modelaux;
965
966 //----- Fin de la cola del gato -----

```

Y este fue el modelo ejemplo al cual se trató de renderizar



Y este es el resultado final, con sus diferentes perspectivas:



2.- Liste los problemas que tuvo a la hora de hacer estos ejercicios y si los resolvió explicar cómo fue, en caso de error adjuntar captura de pantalla

Tuve problemas, o más bien dificultad a la hora de hacer al gato, porque el diseño principal de la cara (ojos, nariz y boca) es algo que aun no hemos visto como hacerlo, por lo que tratar de adaptarlo y que se vea lo más posible parecido al modelo principal fue lo difícil

3.- Conclusión:

a. Los ejercicios del reporte: Complejidad, Explicación.

Considero que los ejercicios de la práctica estuvieron al nivel, aunque si fue algo extenso la elaboración de la elaboración en especial la del gato, debido a que tienen bastantes cosas para mostrar.

b. Comentarios generales: Faltó explicar a detalle, ir más lento en alguna explicación, otros comentarios y sugerencias para mejorar desarrollo de la práctica

Considero que la explicación estuvo bien, solo que al inicio no había dejado muy en claro que no podíamos reiniciar la matriz, por lo que al completar la grúa me tocó corregir parte del código, pero de ahí en fuera siento que estuvo bien explicado.

c. Conclusión

Gracias a esta práctica, pude comprender mejor el concepto de jerarquización, y poder aplicarlos, el cual con dos objetos se tuvieron que emplear el cual fue una grúa y un gato robot, es por ello que fue de mucha utilidad, dado que también pude poner en practica sobre la rotación de los ejes y que con esto ganara más experiencia al andar girando figuras en sus respectivos ejes.

1. Bibliografía en formato APA

- Gato robot. (2018, 20 mayo). Shutterstock. Recuperado 9 de marzo de 2025, de <https://www.shutterstock.com/es/image-vector/cat-hightech-robot-computer-mouse-his-1095602219>